1. Introduction

Reinforcement Learning (RL) has been popular due to its performance in recent complex task solving, namely playing Atari games and Go [1], [2]. RL has also made inroads into the subject of deep learning, intuitively called Deep reinforcement learning. This breakthrough allowed RL to accomplish many previously impossible problems, including learning to play video games from pixels and progressing robotic development to the point where they could drive themselves using camera inputs [3]. It opened many variations of top-grade RL methods, such as Deep Q-Network [4], Asynchronous Advantage Actor- Critic [5], Trust region policy optimization [6], and Soft Actor- Critic [7].

The actor-critic technique is a type of Reinforcement Learning (RL) strategy that combines the Actor-only and Criticonly approaches [8]. Value function approximation using the Bellman equation is used by critic-only approaches, whereas actor-only methods use parameterized policy [9]. Both of them are mixed in such a way that one advantage cancels out each of the disadvantages of the other. Actor-only methods can have large variances. Meanwhile, Critic-only techniques do not provide solid guarantees of near-optimality of the final policy [9]. As illustrated in Fig. 1, the Critic tries to estimate how excellent the current state is with the value function and then employs some method of updating the Actor's policy parameters. As stated in [8], critic-only approaches that employ temporal difference (TD) learning show smaller variance in estimations of anticipated returns [10], [11], [12]. By using TD learning critic, we can minimize the Actor's disadvantages.

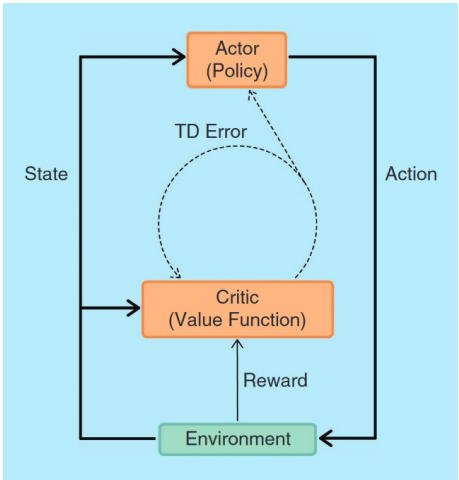

Figure 1 The actor-critic setup. The actor receive the current state and predict an action. The critic take the state and predict the state value. This value is then used to calculate TD Error that will be used to update both the actor and critic parameters.

Temporal Difference (TD) learning, as proposed by Sutton [12], is a method that is driven by the error or difference between two successive temporal predictions. According to [12], the earliest best-known TD method accomplishment was in the Checker playing program in 1959 [15]. TD learning itself can be considered as the center of RL according to [21], where SARSA and Q-learning [16] are also considered as temporal difference learning.

The estimated TD error from the Critic is used to calculate the Actor-loss, which will then be propagated back to the Actor for its policy update, thus called the Actor's learning. One of the properties that can be used to restrict the amount of loss propagated back is to multiply the actor-loss calculation by a constant, usually called the learning rate in the machine learning field. Using a constant with a value of smaller than one can also be seen as discounting the learning aspect of an actor for each learning step.

Some background about the conducted experiment, this experiment is part of a series of others that we will carry out in the future. The goal of this experiment is to see if restricting an actor's learning, specifically in the form of TD error [17], affects the agent's success rate. If it has any impact, we can take it as a hyperparameter to control how much an agent learns. The epsilon here is not to be confused with epsilon in ϵ-greedy policy [17] where the epsilon value is used as a uniform probability to decide whether the agent follows its parameterized policy or takes some random actions. In contrast to the discount factor, which is usually denoted with γ and used to discount future rewards, epsilon (ϵ) suppresses not only the next state value but the TD error itself. It is only applied to the actor learning process. We will explain more about this in Section II. We will discuss the experiment in more detail in Section III.