

Abstract

I present a comparison of computational efficiency between training an agent for playing Pong in the Atari environment in a sequential manner and in a parallelized manner. The agent is trained using a Deep Reinforcement Learning Advantage Actor Critic (A2C). The parallel version of the algorithm Asynchronous A2C (A3C) is analyzed using multiple CPU cores on a single machine instead of special hardware like GPU(Graphical Processing Units) and TPU(Tensor Processing Units).

Keywords

Advantage Actor Critic, Asynchronous

Intorduction

Reinforcement Learning algorithms are a class of algorithms that have proven to be quite promising in the field of decision making. That makes games a suitable area for their application. Combine them with Deep Learning which is known to produce rich representations of input data (visual data in this case) and we get the ability to play games at a human or even better-than-human level (1,2). The game chosen here is Pong which is an atari game and is provided as ann environment in OpenAI gym (3).

Reinforcement Learning

The setup consists of an agent A that interacts with the environment E . The agent interprets the environment by moving from one state s_t at time step t to another state s_{t+1} at time step $t + 1$. In each state it receives an observation in the form of an n -dimensional tensor which might contain partial or all information about the environment in that state. It chooses an action from an action space $A = (a_1, a_2, \dots, a_m)$ based on some computation and as a result lands in another state and receives a signal called the reward R_t quantifying how good or bad the action was. The agent keeps accumulating these reward still it reaches the goal or the episode terminates. An episode runs for a predefined t_{max} number of time steps.

The action at each time step is taken by the agent in accordance with a policy π that is either learned as in Policy Gradient methods or predefined as in Value Function Based methods.

A policy is a function that given a state s , gives a probability distributution over A .

The value of a state s under a policy π is the expected return when starting in s and following π thereafter. Similarly the value of taking action a in state s under policy π is the expected return starting from s , taking the action a and thereafter following policy π .

Advantage Actor Critic

The agent here consists of 2 neural networks. One of them, the Actor, spits out a probability distribution over a set of actions A at each time step and learns the optimal policy, based on a loss function, more about which is described later. The other neural network, the Critic calculates the action-value at each time step telling how good or bad the action taken is and learns to give better estimates of the value, that is learns to be a better critic.

Loss Function

The loss function makes use of the value V_π predicted by the critic. G is used to denote the total return

$$G = R_0 + \gamma R_1 + \gamma^2 R_2 + \dots$$

γ is the discount factor used to place decaying importance, in this case, on later time steps so that earlier steps are reinforced. In Pong, if the opponent misses, it was probably because the shooter maneuvered the ball in a good way on receiving. Similarly earlier actions had a greater role to play in helping the agent becoming a better player.