Asynchronous Advantage Actor Critic

$V_{k+1}^*(s) = \max_a \sum_{s'} P(s'|s,a)[r + \gamma V_k^*(s')]$

$Q_{i+1}(s) \leftarrow \sum_{s'} P(s'|s,a,) \left[r + \gamma \cdot \max_{a'} Q_i(s',a')\right]$

Policy Evaluation

$V_{i+1}^{\pi_k}(s) = \sum_{s'} P(s'|s,\pi_k(s))\left[r + \gamma V_i^{\pi_k}(s')\right]$

Policy Improvement

$\pi_{k+1}(s) = \underset{a}{\operatorname{argmax}} \sum_{s'} P(s'|s,a)[r + \gamma V^{\pi_k}(s')]$

Q – Learning:

$Q_{i+1}(s,a) \leftarrow Q_i(s,a) + \alpha\left(r + \gamma \cdot \max_{a'} Q_i(s',a') - Q_i(s,a)\right)$

DQN Update:

$\theta_{t+1} = \theta_t + \alpha(y_t - Q(s_t,a_t;\theta))\nabla_{\theta_t} Q(s_t,a_t;\theta_t)$

DQN Q Update:

$Q(s_t,a_t;\theta_{t+1}) \approx Q(s_t,a_t;\theta) + \alpha(y_t - Q(s_t,a_t;\theta_t)) \left|\left|\nabla_{\theta_t} Q(s_t,a_t;\theta_t)\right|\right|_2^2$

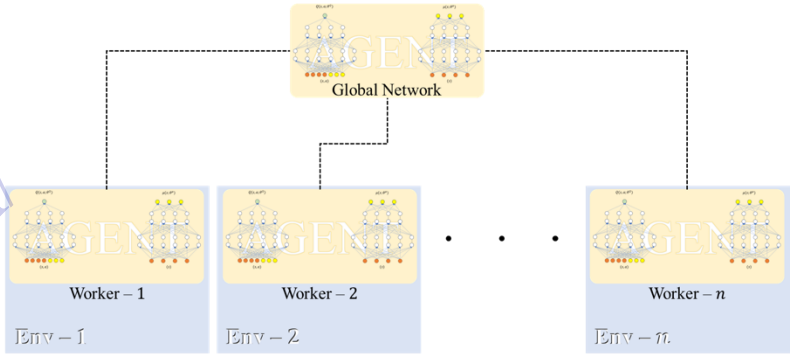
n – step return:

$G_{t:t+n} = R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^n \hat{v}(s_{t+n})$

$\nabla J(\theta) = \mathbb{E}_\pi\left[G_t \frac{\nabla \pi(a|s,\theta)}{\pi(a|s,\theta)}\right]$

$\nabla J(\theta) \propto \sum_s \mu(s) \sum_a (q_\pi(s,a) - b(s))\nabla \pi(a|s,\theta)$

- Actor – Critic with Baseline
  - $\nabla J(\theta) \propto \sum_s \mu(s) \sum_a (q_\pi(s,a) - b(s))\nabla \pi(a|s,\theta)$
  - Helps to minimize variance.
  - No effect on performance gradient $\nabla J(\theta)$
    - $\sum_a b(s)\nabla \pi(a|s,\theta) = b(s)\nabla \sum_a \pi(a|s,\theta) = b(s)\nabla 1 = 0$
- Modified Policy Gradient Theorem:
  - $\theta_{t+1} = \theta_t + \alpha(G_t - b(s_t))\nabla \ln \pi(a_t|s_t,\theta_t)$
  - $\theta_{t+1} = \theta_t + \alpha \hat{A}_t \nabla \ln \pi(a_t|s_t,\theta_t)$
- Baseline should vary with state.
  - In some states all actions have high values. → high baseline
  - In other states all actions may have lower values. → low baseline
  - ∴ a natural choice of baseline is a state value function. → $\hat{V}(s_t,w)$
- Asynchronous updates
  - Multiple actors explores different part of the environment.
  - Online updates in parallel are less likely to be correlated in time. ∴ replay memory isn't required
  - Stabilized learning



Algorithm 1: A3C Algorithm (for each actor-learner thread)

// Assume global shared parameter vectors θ *&* w *& global shared counter* $T = 0$
// Assume thread – specific parameter vectors θ' *&* w'
Initialize thread specific counter $t \leftarrow 1$
Repeat
  Reset gradients: $d\theta \leftarrow 0$ & $dw \leftarrow 0$
  Synchronize thread – specific parameters $\theta' = \theta$ & $w' = \theta_v$
  $t_{start} = t$; Get state s_t
  Repeat
    Perform a_t according to policy $\pi(a_t|s_t;\theta')$
    Receive reward r_t & new state s_{t+1}
    Increment t & T
  Until s_t is terminal **or** $t - t_{start} == t_{max}$
  $G = \begin{cases} 0, & | \ for\ terminal\ s_t \\ V(s_t,w'), for\ non\ -terminal\ s_t & | \ Bootstrap\ from\ last\ state \end{cases}$
  For $i \in \{t-1, \ldots, t_{start}\}$ **do**
    $G \leftarrow r_i + \gamma G$
    Accumulate gradients w.r.t θ' : $d\theta \leftarrow d\theta + \nabla_{\theta'} \log \pi(a_t|s_t;\theta')(G - V(s_i;w'))$
    Accumulate gradients w.r.t w': $dw \leftarrow dw + \partial\left(G - V(s_i;w)\right)^2/\partial w$
  end For
  Perform asynchronous update of θ using $d\theta$ and of w using w'
Until $T > T_{max}$

https://www.profkumar.co.in/

Mnih, Volodymyr, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. "Asynchronous methods for deep reinforcement learning." In International conference on machine learning, pp. 1928-1937. PMLR, 2016.