Variational Deep Q-Networks in Edward

Harri Bell-Thomas

R244: Open Source Project Presentation

19/11/2019

Q-Learning

Q-Learning is *model-free* reinforcement learning.

$\mathcal{Q}$ is the action-value function defining the reward used for reinforcement — this is learned.

Conceptually,

$$\mathcal{Q}^{\pi}(s,\, a) = \mathbb{E}_{a_t \sim \pi(\cdot|s_t)}\left[\sum_{t=0}^{\infty} r_t \gamma^t \,|\, s_0 = s, a_0 = a\right]$$

Q-Learning: Bellman Error

The *value* of $\mathcal{Q}^\pi$ at a certain point in time, t, in terms of the payoff from an initial choice, a_t, and the *value* of the remaining decision problem that results after that choice.

$$J(\pi) = \mathbb{E}_\alpha \left[\left(\mathcal{Q}^\pi(s_t, a_t) - \max_a \mathbb{E}[r_t + \gamma \mathcal{Q}^\pi(s_{t+1}, a)] \right)^2 \right]$$

$$\alpha \to s_0 \sim \rho,\ a_t \sim \pi(\cdot|s_t)$$

Deep Q-Networks

Briefly

Approximate the action-value function $\mathcal{Q}^{\pi}(s, a)$ with a neural network $\mathcal{Q}_{\theta}(s, a)$. The (greedy) policy represented by this is π_{θ}.

Discretise the expectation using K sample trajectories, each with period T. Use this to approximate $J(\theta)$.

$$\tilde{J}(\theta) = \frac{1}{K}\frac{1}{T}\sum_{i=1}^{K}\sum_{t=0}^{T-1}(\mathcal{Q}_{\theta}^{(i)}(s_t^{(i)}, a_t^{(i)})) - \max_{a}\left[r_t + \gamma\mathcal{Q}_{\theta}^{(i)}(s_{t+1}^{(i)}, a)\right])^2$$

Variational Inference

Main Concepts:

1. Try to solve an optimisation problem over a class of tractable distributions, q, parameterised by ϕ, in order to find the one most similar to p.
2. $\phi \leftarrow \min_{\phi} \mathbb{KL}(q_\phi(\theta) \parallel p(\theta|D))$
3. Approximate this using gradient descent.

Variational Deep Q-Networks

Idea: For efficient exploration we need $q_\phi(\theta)$ to be dispersed — near even coverage of the parameter space. Encourage this by adding an entropy bonus to the objective.

$$\mathbb{E}_{\theta \sim q_\phi(\theta)} \left[\left(\mathcal{Q}_\theta(s_j, a_j) - \max_{a'} \mathbb{E}[r_j + \gamma \mathcal{Q}_\theta(s'_j, a')] \right)^2 \right] - \lambda \mathbb{H}(q_\phi(\theta))$$

Assigning systematic randomness to $\mathcal{Q}$ enables efficient exploration of the policy space. Further, encouraging high entropy over parameter distribution prevent premature convergence.

tl;dr Higher chance of finding maximal rewards in a faster time than standard DQNs.

Algorithm

Algorithm 1 Variational DQN

INPUT: improper uniform prior $p(\theta)$; target parameter update period τ ; learning rate α; generative model variance σ^2
INITIALIZE: parameters ϕ, ϕ^-; replay buffer $R \leftarrow \{\}$; step counter $counter \leftarrow 0$
for $e = 1, 2, 3...E$ **do**
  while episode not terminated **do**
    $counter \leftarrow counter + 1$
    Sample $\theta \sim q_\theta(\phi)$
    In state s_t, choose $a_t = \arg\max_a Q_\theta(s_t, a)$, get transition s_{t+1} and reward r_t
    Save experience tuple $\{s_t, a_t, r_t, s_{t+1}\}$ to buffer R
    Sample N parameters $\theta_j^- \sim q_\theta(\phi^-)$ and sample N tuples $D = \{s_j, a_j, t_j, s'_j\}$ from R
    Compute target $d_j = r_j + \max_{a'} Q_{\theta_j^-}(s'_j, a')$ for jth tuple in D
    Take gradient $\Delta\phi$ of the KL divergence in (8)
    $\phi \leftarrow \phi - \alpha\Delta\phi$
    if $counter$ mod $\tau = 0$ **then**
      Update target parameter $\phi^- \leftarrow \phi$
    end if
  end while
end for

Figure: VDQN Pseudocode.

Aim / Goals

Workplan

Questions?