Comparison between conventional TD algorithm and TD with gradient descent

1. Intuition

In general, since the usage of bootstrapping, conventional TD algorithms like Sarsa and Q-learning are not actually used the stochastic gradient decent to minimize the Objective function for VFA (value functional approximation). We could consider like that, in our algorithms, our weights are updated immediately after each step and each step is dependent on the weights. They influence each other mutually. (ie. It is not the real "stochastic", we could compare with MC method, the weights will not be updated until end of the episodes).

However, the true SGD is likely to diverge under off-policy learning. As the behavior policy may choose the action which will not be chosen by target policy. During our experiments, sometimes, we see the divergence of Q-weights under Q-learning and Sarsa. Intuitively, we introduce the TDC to extend VFA for off-policy learning and on-policy Sarsa

(In general, we can't guarantee the convergence of "not stochastic" gradient decent method)

2. Explanation for TDC

For conventional q-learning, we use the Value Error (denoted as VE) as the Objective function.

$$\overline{VE} \doteq \sum_{s \in S} \mu(s)[v_\pi(s) - \widehat{v}(s,w)]^2$$

But the TDC q-learning bases on a different Objective function PMSBE (projected minimize square bellman error)

Bellman equation is that

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)\left[r + \gamma v_\pi(s')\right], \qquad \text{for all } s \in \mathcal{S}.$$

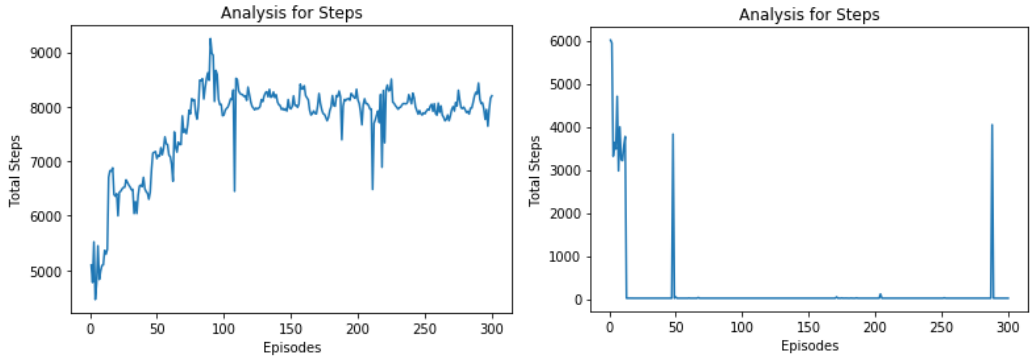
But it only holds for true value function (not approximated), therefore, we could use the left and right sides to build the bellman error

$$\begin{aligned}\bar{\delta}_{\mathbf{w}}(s) &\doteq \left(\sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)\left[r + \gamma v_{\mathbf{w}}(s')\right]\right) - v_{\mathbf{w}}(s) \\ &= \mathbb{E}_\pi\left[R_{t+1} + \gamma v_{\mathbf{w}}(S_{t+1}) - v_{\mathbf{w}}(S_t) \;\middle|\; S_t = s, A_t \sim \pi\right],\end{aligned}$$

The PBE is the bellman error after projecting to parameters hyperplane.

In order to compare the performance of 2 algorithms, given the 2 plots below, we could see that conventional q-learning diverges and q-learning with gradient correction converges. (left is conventional q-learning and right is TDC q-learning)

In particular, the convergent speed is fast, it converges just after around 50 episodes.



We also use TDC to on-policy learning Sarsa. Given the 2 plots below, we could see the divergence of sarsa and convergence of tdc Sarsa