Patrick Pegus
Mini Project 3
December 10, 2015
CMPSCI-689
Prof. Sridhar Mahadevan

1. (a)

$$\begin{aligned} w_{FP} &= argmin_w \left\| \Pi_\Phi T^\pi(\hat{V}) - \hat{V} \right\| \\ &= argmin_w \left\| \Pi_\Phi \left(R^\pi + \gamma P^\pi \Phi w \right) - \Phi w \right\| \end{aligned}$$

The norm is minimized if

$$\begin{aligned} \Phi w &= \Pi_\Phi \left(R^\pi + \gamma P^\pi \Phi w \right) \\ \Phi w &= \Phi \left(\Phi^T \Phi \right)^{-1} \Phi^T \left(R^\pi + \gamma P^\pi \Phi w \right) \\ w &= \left(\Phi^T \Phi \right)^{-1} \Phi^T \left(R^\pi + \gamma P^\pi \Phi w \right) \\ w &= \left(\Phi^T \Phi \right)^{-1} \Phi^T R^\pi + \gamma \left(\Phi^T \Phi \right)^{-1} \Phi^T P^\pi \Phi w \\ w - \gamma \left(\Phi^T \Phi \right)^{-1} \Phi^T P^\pi \Phi w &= \left(\Phi^T \Phi \right)^{-1} \Phi^T R^\pi \\ \left(I - \gamma \left(\Phi^T \Phi \right)^{-1} \Phi^T P^\pi \Phi \right) w &= \left(\Phi^T \Phi \right)^{-1} \Phi^T R^\pi \\ w &= \left(I - \gamma \left(\Phi^T \Phi \right)^{-1} \Phi^T P^\pi \Phi \right)^{-1} \left(\Phi^T \Phi \right)^{-1} \Phi^T R^\pi \\ w &= \left(\left(\Phi^T \Phi \right) \left(I - \gamma \left(\Phi^T \Phi \right)^{-1} \Phi^T P^\pi \Phi \right) \right)^{-1} \Phi^T R^\pi \\ w &= \left(\Phi^T \Phi - \gamma \Phi^T P^\pi \Phi \right)^{-1} \Phi^T R^\pi \\ w &= \left(\Phi^T \left(\Phi - \gamma P^\pi \Phi \right) \right)^{-1} \Phi^T R^\pi \\ w &= \left(\Phi^T \left(I - \gamma P^\pi \right) \Phi \right)^{-1} \Phi^T R^\pi \end{aligned}$$

(b)

$$\begin{aligned} w_{LS} &= argmin_w \left\| T^\pi(\hat{V}) - \hat{V} \right\| \\ &= argmin_w \left\| R^\pi + \gamma P^\pi \Phi w - \Phi w \right\| \end{aligned}$$

The norm is minimized if

$$\Phi w - \gamma P^\pi \Phi w = \left(I - \gamma P^\pi \right) \Phi w \approx R^\pi$$

This is equivalent to a weighted least squares problem $Ax \approx b$, whose solution is $x = \left(A^T C A \right)^{-1} A^T C b$, where C is a diagonal matrix with entries giving the non-uniform weights measuring the "length" in the space [1]. Let $A = \left(I - \gamma P^\pi \right) \Phi w$, $b = R^\pi$, and $C = I$, if the basis functions uniform importance. Therefore,

$$\begin{aligned} w &= \left(\left(\left(I - \gamma P^\pi \right) \Phi \right)^T I \left(\left(I - \gamma P^\pi \right) \Phi \right) \right)^{-1} \left(\left(I - \gamma P^\pi \right) \Phi \right)^T I R^\pi \\ &= \left(\Phi^T \left(I - \gamma P^\pi \right)^T \left(I - \gamma P^\pi \right) \Phi \right)^{-1} \Phi^T \left(I - \gamma P^\pi \right)^T R^\pi \end{aligned}$$

2. (a)

$$\begin{aligned}
\Delta_{\theta_i} L_i(\theta_i) &= \Delta_{\theta_i} E_{s,a\sim p(\cdot)}\left[(y_i - Q(s,a;\theta_i))^2\right] \\
&= -2E_{s,a\sim p(\cdot)}\left[(y_i - Q(s,a;\theta_i))\,\Delta_{\theta_i} Q(s,a;\theta_i)\right] \\
&= -2E_{s,a\sim p(\cdot)}\left[\left(E_{s'\sim\varepsilon}\left[r + \gamma \max_{a'} Q(s',a';\theta_{i-1})\middle| s,a\right] - Q(s,a;\theta_i)\right)\Delta_{\theta_i} Q(s,a;\theta_i)\right] \\
&= -2E_{s,a\sim p(\cdot)}\left[\left(\sum_{s'}\left[\left(r + \gamma \max_{a'} Q(s',a';\theta_{i-1})\right)p(s')\right] - Q(s,a;\theta_i)\right)\Delta_{\theta_i} Q(s,a;\theta_i)\right] \\
&= -2E_{s,a\sim p(\cdot)}\sum_{s'}\left[\left(\left(r + \gamma \max_{a'} Q(s',a';\theta_{i-1})\right)p(s') - Q(s,a;\theta_i)\right)\Delta_{\theta_i} Q(s,a;\theta_i)\right] \\
&= -2E_{s,a\sim p(\cdot);s'\sim\varepsilon}\left[\left(r + \gamma \max_{a'} Q(s',a';\theta_{i-1}) - Q(s,a;\theta_i)\right)\Delta_{\theta_i} Q(s,a;\theta_i)\right]
\end{aligned}$$

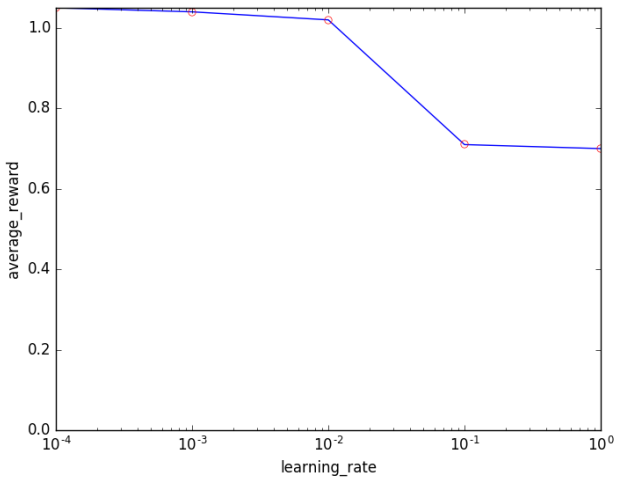
We can disregard the -2, since a constant multiplier will not affect when the gradient is 0.

(b) TD-gammon approximated a value function for states instead of an action-value function approximation used in Deep Mind. Also, TD-gammon learnt this state value function on-policy, meaning that it follows an exploration policy. On the other hand, Deep Mind learnt about the greedy strategy of choosing the action with the greatest current value.

(c) Deep Mind uses experience replay to smooth the behavior distribution over many previously seen sequences. If experience replay was not used, then feedback loops could occur, meaning the behavior distribution could shift dramatically based on the most recent actions. These feedback loops may cause parameters choices that get stuck on poor local minima or diverge.

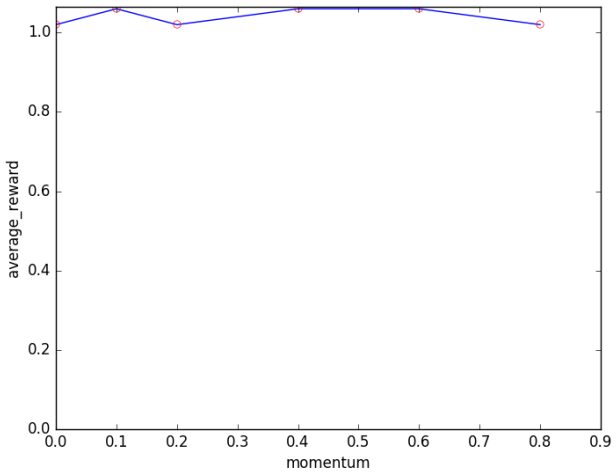
(d) The two left graphs show the average reward per episode vs. training epochs. For both Breakout and Seaquest, it is difficult to tell whether Deep Mind is creating more valuable policies with increased training time. In the two right graphs, they more directly measure this by showing the average action value vs. training epochs. Here we can see that the learning algorithm makes more consistent progress on Seaquest.

Figure 1: Average reward vs. learning rate.



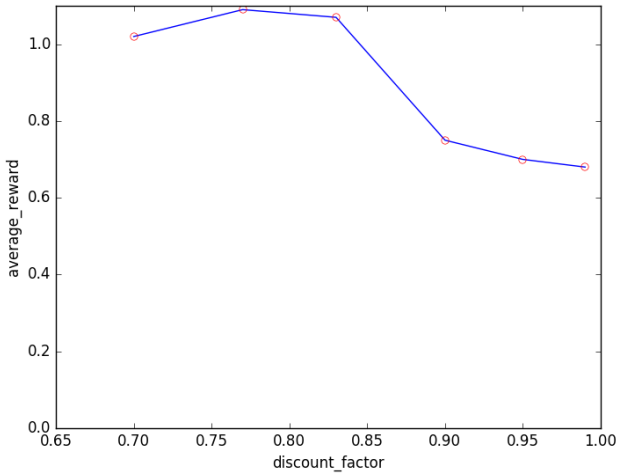
3. (a) Generally, in Figure 1 we see average reward increase as the learning rate or amount we move the parameters in the direction of the gradient decrease. However, as the learning rate decreased, the training time or the time it took the average reward to stabilize was substantially greater. No divergence in average reward occurred at the higher learning rates.

Figure 2: Average reward vs. momentum.



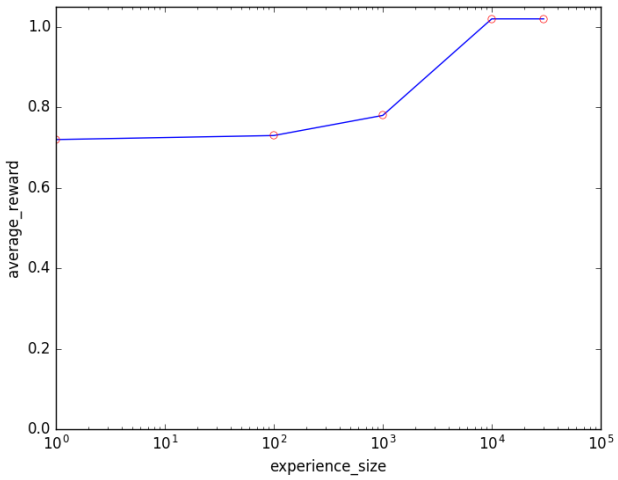
(b) As Figure 2 shows, changing the momentum had no visible effect on the average reward. This makes sense as increased momentum will at best force stochastic gradient descent to reach optimal values more quickly.

Figure 3: Average reward vs. discount factor.



(c) Figure 3 shows that average reward generally decreases as the discount factor increases. Therefore, the greedy strategy of maximizing the reward of short-term actions results in a more valuable policy. This agrees with the environment shown in the graphic. For instance, there don't seem to be instances in which the agent must eat poison in order to get many apples or alternatively avoid an apple in order to avoid eating a lot of poison.

Figure 4: Average reward vs. experience size.



(d) As Figure 4 shows, increased replay memory size improved average reward, especially as the size reached 10000. With this implementation, there was no divergence with low memory size alluded to in [2].

References

[1] S. Mahadevan. *Learning Representation and Control in Markov Decision Processes*. Now Publishers Inc, 2009.

[2] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.