Exercises - Chapter 12

Carl Fredriksson, c@msp.se

Exercise 12.1

Just as the return can be written recursively in terms of the first reward and itself one-step later (3.9), so can the λ-return. Derive the analogous recursive relationship from (12.2) and (12.1).

My answer:

$$
\begin{aligned}
G_t^\lambda &\doteq (1-\lambda)\sum_{n=1}^{\infty}\lambda^{n-1}G_{t:t+n} \\
&= (1-\lambda)\Big(G_{t:t+1} + \lambda G_{t:t+2} + \lambda^2 G_{t:t+3} + \dots\Big) \\
&= (1-\lambda)\Big(\big[R_{t+1} + \gamma\hat{v}(S_{t+1},\mathbf{w}_t)\big] + \lambda\big[R_{t+1} + \gamma R_{t+2} + \gamma^2\hat{v}(S_{t+2},\mathbf{w}_{t+1})\big] + \lambda^2\big[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3\hat{v}(S_{t+3},\mathbf{w}_{t+2})\big] + \dots\Big) \\
&= (1-\lambda)R_{t+1}\Big(\sum_{n=0}^{\infty}\gamma^n\Big) + (1-\lambda)\gamma\hat{v}(S_{t+1},\mathbf{w}_t) + (1-\lambda)\Big(\lambda\big[\gamma R_{t+2} + \gamma^2\hat{v}(S_{t+2},\mathbf{w}_{t+1})\big] + \lambda^2\big[\gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3\hat{v}(S_{t+3},\mathbf{w}_{t+2})\big] + \dots\Big) \\
&= \frac{(1-\lambda)}{(1-\lambda)}R_{t+1} + (1-\lambda)\gamma\hat{v}(S_{t+1},\mathbf{w}_t) + (1-\lambda)\lambda\gamma\Big(\big[R_{t+2} + \gamma\hat{v}(S_{t+2},\mathbf{w}_{t+1})\big] + \lambda\big[R_{t+2} + \gamma R_{t+3} + \gamma^2\hat{v}(S_{t+3},\mathbf{w}_{t+2})\big] + \dots\Big) \\
&= R_{t+1} + (1-\lambda)\gamma\hat{v}(S_{t+1},\mathbf{w}_t) + \lambda\gamma(1-\lambda)\sum_{n=1}^{\infty}\lambda^{n-1}G_{t+1:t+1+n} \\
&= R_{t+1} + (1-\lambda)\gamma\hat{v}(S_{t+1},\mathbf{w}_t) + \lambda\gamma G_{t+1}^\lambda
\end{aligned}
$$

Exercise 12.2

The parameter λ characterizes how fast the exponential weighting in Figure 12.2 falls off, and thus how far into the future the λ-return algorithm looks in determining its update. But a rate factor such as is sometimes an awkward way of characterizing the speed of the decay. For some purposes it is better to specify a time constant, or half-life. What is the equation relating λ and the half-life, τ_λ, the time by which the weighting sequence will have fallen to half of its initial value?

My answer:

$$
\begin{aligned}
(1-\lambda)\lambda^\tau &= (1-\lambda)\frac{1}{2} \\
\lambda^\tau &= \frac{1}{2} \\
\tau &= \log_\lambda(\frac{1}{2})
\end{aligned}
$$

$$
\tau_\lambda = t + \tau + 1 = t + \log_\lambda(\frac{1}{2}) + 1
$$

Adding 1 since the n-step return $G_{t:t+n}$ is weighted by λ^{n-1}.

Exercise 12.3

Some insight into how TD(λ) can closely approximate the off-line λ-return algorithm can be gained by seeing that the latter's error term (in brackets in (12.4)) can be written as the sum of TD errors (12.6) for a single fixed $\mathbf{w}$. Show this, following the pattern of (6.6), and using the recursive relationship for the λ-return you obtained in Exercise 12.1.

My answer:

$$
\begin{aligned}
G_t^\lambda - \hat{v}(S_t,\mathbf{w}) &= R_{t+1} + (1-\lambda)\gamma\hat{v}(S_{t+1},\mathbf{w}) + \lambda\gamma G_{t+1}^\lambda - \hat{v}(S_t,\mathbf{w}) \\
&= R_{t+1} + \gamma\hat{v}(S_{t+1},\mathbf{w}) - \hat{v}(S_t,\mathbf{w}) + \lambda\gamma G_{t+1}^\lambda - \lambda\gamma\hat{v}(S_{t+1},\mathbf{w}) \\
&= \delta_t + \lambda\gamma\big[G_{t+1}^\lambda - \hat{v}(S_{t+1},\mathbf{w})\big] \\
&= \delta_t + \lambda\gamma\delta_{t+1} + (\lambda\gamma)^2\big[G_{t+2}^\lambda - \hat{v}(S_{t+2},\mathbf{w})\big] \\
&= \delta_t + \lambda\gamma\delta_{t+1} + (\lambda\gamma)^2\delta_{t+2} + \dots + (\lambda\gamma)^{T-t-2}\delta_{T-2} + (\lambda\gamma)^{T-t-1}\big[G_{T-1}^\lambda - \hat{v}(S_{T-1},\mathbf{w})\big] \\
&= \delta_t + \lambda\gamma\delta_{t+1} + (\lambda\gamma)^2\delta_{t+2} + \dots + (\lambda\gamma)^{T-t-2}\delta_{T-2} + (\lambda\gamma)^{T-t-1}\big[R_T - \hat{v}(S_{T-1},\mathbf{w})\big] \\
&= \sum_{k=t}^{T-1}(\lambda\gamma)^{k-t}\delta_k
\end{aligned}
$$

Exercise 12.4

Use your result from the preceding exercise to show that, if the weight updates over an episode were computed on each step but not actually used to change the weights ($\mathbf{w}$ remained fixed), then the sum of TD(λ)'s weight updates would be the same as the sum of the off-line λ-return algorithm's updates.

My answer:

The sum of TD(λ)'s weight updates would be

$$
\begin{aligned}
\sum_{t=0}^{T-1} \alpha\delta_t z_t &= \alpha\Bigg(\delta_0\nabla\hat{v}(S_0,\mathbf{w}) + \delta_1\big[\gamma\lambda\nabla\hat{v}(S_0,\mathbf{w}) + \nabla\hat{v}(S_1,\mathbf{w})\big] + \delta_2\big[(\gamma\lambda)^2\nabla\hat{v}(S_0,\mathbf{w}) + \gamma\lambda\nabla\hat{v}(S_1,\mathbf{w}) + \nabla\hat{v}(S_2,\mathbf{w})\big] + \dots\Bigg) \\
&= \alpha\Bigg(\nabla\hat{v}(S_0,\mathbf{w})\big[\delta_0 + \gamma\lambda\delta_1 + (\gamma\lambda)^2\delta_2 + \dots + (\gamma\lambda)^T - 1\delta_{T-1}\big] + \\
&\qquad \nabla\hat{v}(S_1,\mathbf{w})\big[\delta_1 + \gamma\lambda\delta_2 + (\gamma\lambda)^2\delta_3 + \dots + (\gamma\lambda)^{T-2}\delta_{T-1}\big] + \dots + \\
&\qquad \nabla\hat{v}(S_{T-1},\mathbf{w})\big[\delta_{T-1}\big]\Bigg) \\
&= \alpha\Bigg(\nabla\hat{v}(S_0,\mathbf{w})\sum_{k=0}^{T-1}(\gamma\lambda)^k\delta_k + \nabla\hat{v}(S_1,\mathbf{w})\sum_{k=1}^{T-1}(\gamma\lambda)^{k-1}\delta_k + \dots + \nabla\hat{v}(S_{T-1},\mathbf{w})\sum_{k=T-1}^{T-1}(\gamma\lambda)^{k-(T-1)}\delta_k\Bigg) \\
&= \alpha\Bigg(\big[G_0^\lambda - \hat{v}(S_0,\mathbf{w})\big]\nabla\hat{v}(S_0,\mathbf{w}) + \big[G_1^\lambda - \hat{v}(S_1,\mathbf{w})\big]\nabla\hat{v}(S_1,\mathbf{w}) + \dots + \big[G_{T-1}^\lambda - \hat{v}(S_{T-1},\mathbf{w})\big]\nabla\hat{v}(S_{T-1},\mathbf{w})\Bigg) \\
&= \alpha\sum_{t=0}^{T-1}\big[G_t^\lambda - \hat{v}(S_t,\mathbf{w})\big]\nabla\hat{v}(S_t,\mathbf{w})
\end{aligned}
$$

which is the same as the sum of the offline λ-return algorithm's updates (12.4).

Exercise 12.5

Several times in this book (often in exercises) we have established that returns can be written as sums of TD errors if the value function is held constant. Why is (12.10) another instance of this? Prove (12.10).

My answer:

Let's start by writing the k-step λ-return recursively ($h = t + k$)

$$
\begin{aligned}
G_{t:t+k}^\lambda &\doteq (1-\lambda)\sum_{n=1}^{k-1}\lambda^{n-1}G_{t:t+n} + \lambda^{k-1}G_{t:t+k} \\
&= (1-\lambda)\Bigg(\big[R_{t+1} + \gamma\hat{v}(S_{t+1},\mathbf{w}_t)\big] + \lambda\big[R_{t+1} + \gamma R_{t+2} + \gamma^2\hat{v}(S_{t+2},\mathbf{w}_{t+1})\big] + \dots + \\
&\qquad\qquad \lambda^{k-2}\big[R_{t+1} + \gamma R_{t+2} + \dots + \gamma^{k-1}\hat{v}(S_{t+k-1},\mathbf{w}_{t+k-2})\big]\Bigg) + \lambda^{k-1}\big[R_{t+1} + \gamma R_{t+2} + \dots + \gamma^k\hat{v}(S_{t+k},\mathbf{w}_{t+k-1})\big] \\
&= (1-\lambda)\gamma\hat{v}(S_{t+1},\mathbf{w}_t) + R_{t+1}\Bigg(\lambda^{k-1} + (1-\lambda)\sum_{n=0}^{k-2}\lambda^n\Bigg) + \\
&\quad (1-\lambda)\Bigg(\gamma\lambda\big[R_{t+2} + \gamma\hat{v}(S_{t+2},\mathbf{w}_{t+1})\big] + \dots + \gamma\lambda^{k-2}\big[R_{t+2} + \dots + \gamma^{k-2}\hat{v}(S_{t+k-1},\mathbf{w}_{t+k-2})\big]\Bigg) + \\
&\quad \gamma\lambda^{k-1}\big[R_{t+2} + \dots + \gamma^{k-1}\hat{v}(S_{t+k},\mathbf{w}_{t+k-1})\big] \\
&= (1-\lambda)\gamma\hat{v}(S_{t+1},\mathbf{w}_t) + R_{t+1}\Bigg(\lambda^{k-1}(1-\lambda)\sum_{n=0}^{\infty}\lambda^n + (1-\lambda)\sum_{n=0}^{k-2}\lambda^n\Bigg) + \\
&\quad \gamma\lambda\Bigg((1-\lambda)\Bigg(\big[R_{t+2} + \gamma\hat{v}(S_{t+2},\mathbf{w}_{t+1})\big] + \dots + \lambda^{k-3}\big[R_{t+2} + \dots + \gamma^{k-2}\hat{v}(S_{t+k-1},\mathbf{w}_{t+k-2})\big]\Bigg) + \\
&\quad \lambda^{k-2}\big[R_{t+2} + \dots + \gamma^{k-1}\hat{v}(S_{t+k},\mathbf{w}_{k+t-1})\big]\Bigg) \\
&= (1-\lambda)\gamma\hat{v}(S_{t+1},\mathbf{w}_t) + R_{t+1}\Bigg((1-\lambda)\sum_{n=k-1}^{\infty}\lambda^n + (1-\lambda)\sum_{n=0}^{k-2}\lambda^n\Bigg) + \\
&\quad \gamma\lambda\Bigg((1-\lambda)\sum_{n=1}^{k-2}\lambda^{n-1}G_{t+1:t+1+n} + \lambda^{k-2}G_{t+1:t+k}\Bigg) \\
&= (1-\lambda)\gamma\hat{v}(S_{t+1},\mathbf{w}_t) + R_{t+1}\Bigg((1-\lambda)\sum_{n=0}^{\infty}\lambda^n\Bigg) + \gamma\lambda G_{t+1:t+k}^\lambda \\
&= R_{t+1} + (1-\lambda)\gamma\hat{v}(S_{t+1},\mathbf{w}_t) + \gamma\lambda G_{t+1:t+k}^\lambda
\end{aligned}
$$

We can now use the recursive relationship to prove (12.10)

$$
\begin{aligned}
G_{t:t+k}^\lambda &= R_{t+1} + (1-\lambda)\gamma\hat{v}(S_{t+1},\mathbf{w}_t) + \gamma\lambda G_{t+1:t+k}^\lambda \\
&= R_{t+1} + \gamma\hat{v}(S_{t+1},\mathbf{w}_t) - \lambda\gamma\hat{v}(S_{t+1},\mathbf{w}_t) + \gamma\lambda G_{t+1:t+k}^\lambda + \hat{v}(S_t,\mathbf{w}_{t-1}) - \hat{v}(S_t,\mathbf{w}_{t-1}) \\
&= \hat{v}(S_t,\mathbf{w}_{t-1}) + \delta_t' + \gamma\lambda\big[G_{t+1:t+k}^\lambda - \hat{v}(S_{t+1},\mathbf{w}_t)\big] \\
&= \hat{v}(S_t,\mathbf{w}_{t-1}) + \delta_t' + \gamma\lambda\big[R_{t+2} + (1-\lambda)\gamma\hat{v}(S_{t+2},\mathbf{w}_{t+1}) + \gamma\lambda G_{t+2:t+k}^\lambda - \hat{v}(S_{t+1},\mathbf{w}_t)\big] \\
&= \hat{v}(S_t,\mathbf{w}_{t-1}) + \delta_t' + \gamma\lambda\delta_{t+1}' + (\gamma\lambda)^2\big[G_{t+2:t+k}^\lambda - \hat{v}(S_{t+2},\mathbf{w}_{t+1})\big] \\
&= \hat{v}(S_t,\mathbf{w}_{t-1}) + \delta_t' + \gamma\lambda\delta_{t+1}' + (\gamma\lambda)^2\delta_{t+2}' + \dots + (\gamma\lambda)^{k-1}\delta_{t+k-1}' + (\gamma\lambda)^k\big[G_{t+k:t+k}^\lambda - \hat{v}(S_{t+k},\mathbf{w}_{t+k-1})\big] \\
&= \hat{v}(S_t,\mathbf{w}_{t-1}) + \delta_t' + \gamma\lambda\delta_{t+1}' + (\gamma\lambda)^2\delta_{t+2}' + \dots + (\gamma\lambda)^{k-1}\delta_{t+k-1}' + (\gamma\lambda)^k\big[\hat{v}(S_{t+k},\mathbf{w}_{t+k-1}) - \hat{v}(S_{t+k},\mathbf{w}_{t+k-1})\big] \\
&= \hat{v}(S_t,\mathbf{w}_{t-1}) + \delta_t' + \gamma\lambda\delta_{t+1}' + (\gamma\lambda)^2\delta_{t+2}' + \dots + (\gamma\lambda)^{k-1}\delta_{t+k-1}' \\
&= \hat{v}(S_t,\mathbf{w}_{t-1}) + \sum_{i=t}^{t+k-1}(\gamma\lambda)^{i-t}\delta_i'
\end{aligned}
$$

Exercise 12.6

Modify the pseudocode for Sarsa(λ) to use dutch traces (12.11) without the other distinctive features of a true online algorithm. Assume linear function approximation and binary features.

My answer:

Change this part

```
- Loop for $i$ in $\mathcal{F}(S, A)$:
    - $\delta \leftarrow \delta - w_i$
    - $z_i \leftarrow z_i + 1$
    - or $z_i \leftarrow 1$
```

to

```
- $s \leftarrow 0$
- Loop for $i$ in $\mathcal{F}(S, A)$:
    - $s \leftarrow s + z_i$
- Loop for $i$ in $\mathcal{F}(S, A)$:
    - $\delta \leftarrow \delta - w_i$
    - $z_i \leftarrow z_i + (1 - \alpha s)$
```

Exercise 12.7

Generalize the three recursive equations above to their truncated versions, defining $G_{t:h}^{\lambda s}$ and $G_{t:h}^{\lambda a}$.

My answer:

$$G_{t:h}^{\lambda s} \doteq R_{t+1} + \gamma_{t+1}\Big((1-\lambda_{t+1})\hat{v}(S_{t+1}, \mathbf{w}_t) + \lambda_{t+1} G_{t+1:h}^{\lambda s}\Big)$$

$$G_{t:h}^{\lambda a} \doteq R_{t+1} + \gamma_{t+1}\Big((1-\lambda_{t+1})\hat{q}(S_{t+1}, A_{t+1}, \mathbf{w}_t) + \lambda_{t+1} G_{t+1:h}^{\lambda a}\Big)$$

$$G_{t:h}^{\lambda a} \doteq R_{t+1} + \gamma_{t+1}\Big((1-\lambda_{t+1})\bar{V}_t(S_{t+1}) + \lambda_{t+1} G_{t+1:h}^{\lambda a}\Big)$$

Exercise 12.8

Prove that (12.24) becomes exact if the value function does not change. To save writing, consider the case of $t = 0$, and use the notation $V_k = \hat{v}(S_k, \mathbf{w})$.

My answer:

$$\begin{aligned}
G_0^{\lambda s} &\doteq \rho_0\Big(R_1 + \gamma_1\big[(1-\lambda_1)V_1 + \lambda_1 G_1^{\lambda s}\big]\Big) + (1-\rho_0)V_0 \\
&= \rho_0 R_1 + \rho_0\gamma_1 V_1 - \rho_0\gamma_1\lambda_1 V_1 + \rho_0\gamma_1\lambda_1 G_1^{\lambda s} + V_0 - \rho_0 V_0 \\
&= V_0 + \rho_0(R_1 + \gamma_1 V_1 - V_0) - \rho_0\gamma_1\lambda_1 V_1 + \rho_0\gamma_1\lambda_1 G_1^{\lambda s} \\
&= V_0 + \rho_0\delta_0^s + \rho_0\gamma_1\lambda_1(G_1^{\lambda s} - V_1) \\
&= V_0 + \rho_0\delta_0^s + \rho_0\gamma_1\lambda_1\big(V_1 + \rho_1\delta_1^s + \rho_1\gamma_2\lambda_2(G_2^{\lambda s} - V_2) - V_1\big) \\
&= V_0 + \rho_0\delta_0^s + \rho_0\rho_1\gamma_1\lambda_1\delta_1^s + \rho_0\rho_1\gamma_1\lambda_1\gamma_2\lambda_2(G_2^{\lambda s} - V_2) \\
&= V_0 + \rho_0\delta_0^s + \rho_0\rho_1\gamma_1\lambda_1\delta_1^s + \rho_0\rho_1\rho_2\gamma_1\lambda_1\gamma_2\lambda_2\delta_2^s + \dots \\
&= V_0 + \rho_0\sum_{k=0}^{\infty}\delta_k^s\prod_{i=1}^{k}\gamma_i\lambda_i\rho_i
\end{aligned}$$

Exercise 12.9

The truncated version of the general off-policy return is denoted $G_{t:h}^{\lambda s}$. Guess the correct equation, based on (12.24).

My answer:

$$G_{t:h}^{\lambda s} \approx \hat{v}(S_t, \mathbf{w}) + \rho_t\sum_{k=t}^{h-1}\delta_k^s\prod_{i=t+1}^{k}\gamma_i\lambda_i\rho_i$$

Exercise 12.10

Prove that (12.27) becomes exact if the value function does not change. To save writing, consider the case of $t = 0$, and use the notation $Q_k = \hat{q}(S_k, A_k, \mathbf{w})$. Hint: Start by writing out δ_0^a and $G_0^{\lambda a}$, then $G_0^{\lambda a} - Q_0$.

My answer:

$$\delta_0^a = R_1 + \gamma_1\bar{V}_0(S_1) - Q_0$$

$$\begin{aligned}
G_0^{\lambda a} &= R_1 + \gamma_1 \Big(\bar{V}_0(S_1) + \lambda_1 \rho_1 \big[G_1^{\lambda a} - Q_1 \big] \Big) \\
&= R_1 + \gamma_1 \bar{V}_0(S_1) + \gamma_1 \lambda_1 \rho_1 \big[G_1^{\lambda a} - Q_1 \big] + Q_0 - Q_0 \\
&= Q_0 + \delta_0^a + \gamma_1 \lambda_1 \rho_1 \big[G_1^{\lambda a} - Q_1 \big] \\
&= Q_0 + \delta_0^a + \gamma_1 \lambda_1 \rho_1 \Big(R_2 + \gamma_2 \bar{V}_1(S_2) + \gamma_2 \lambda_2 \rho_2 \big[G_2^{\lambda a} - Q_2 \big] - Q_1 \Big) \\
&= Q_0 + \delta_0^a + \gamma_1 \lambda_1 \rho_1 \delta_1^a + \gamma_1 \lambda_1 \rho_1 \gamma_2 \lambda_2 \rho_2 \big[G_2^{\lambda a} - Q_2 \big] \\
&= Q_0 + \delta_0^a + \gamma_1 \lambda_1 \rho_1 \delta_1^a + \gamma_1 \lambda_1 \rho_1 \gamma_2 \lambda_2 \rho_2 \delta_2^a + \dots \\
&= Q_0 + \sum_{k=0}^{\infty} \delta_k^a \prod_{i=1}^{k} \gamma_i \lambda_i \rho_i
\end{aligned}$$

Exercise 12.11

The truncated version of the general off-policy return is denoted $G_{t:h}^{\lambda a}$. Guess the correct equation for it, based on (12.27).

My answer:

$$G_{t:h}^{\lambda a} \approx \hat{q}(S_t, A_t, \mathbf{w}_t) + \sum_{k=t}^{h-1} \delta_k^a \prod_{i=t+1}^{k} \gamma_i \lambda_i \rho_i$$

Exercise 12.12

Show in detail the steps outlined above for deriving (12.29) from (12.27). Start with the update (12.15), substitute $G_t^{\lambda a}$ from (12.26) for G_t^{λ}, then follow similar steps as led to (12.25).

My answer:

$$\begin{aligned}
w_{t+1} &\doteq w_t + \alpha \big[G_t^{\lambda a} - \hat{q}(S_t, A_t, \mathbf{w}_t) \big] \nabla \hat{q}(S_t, A_t, \mathbf{w}_t) \\
&\approx w_t + \alpha \big[\sum_{k=t}^{\infty} \delta_k^a \prod_{i=t+1}^{k} \gamma_i \lambda_i \rho_i \big] \nabla \hat{q}(S_t, A_t, \mathbf{w}_t)
\end{aligned}$$

The sum of the forward update over time is

$$\begin{aligned}
\sum_{t=0}^{\infty} (w_{t+1} - w_t) &\approx \sum_{t=0}^{\infty} \sum_{k=t}^{\infty} \alpha \delta_k^a \nabla \hat{q}(S_t, A_t, \mathbf{w}_t) \prod_{i=t+1}^{k} \gamma_i \lambda_i \rho_i \\
&= \sum_{k=0}^{\infty} \sum_{t=0}^{k} \alpha \delta_k^a \nabla \hat{q}(S_t, A_t, \mathbf{w}_t) \prod_{i=t+1}^{k} \gamma_i \lambda_i \rho_i \\
&= \sum_{k=0}^{\infty} \alpha \delta_k^a \sum_{t=0}^{k} \nabla \hat{q}(S_t, A_t, \mathbf{w}_t) \prod_{i=t+1}^{k} \gamma_i \lambda_i \rho_i
\end{aligned}$$

Now we show that if the entire expression from the second sum on was the trace at time k, we could update it from its value at time $k - 1$ by:

$$\begin{aligned}
\mathbf{z}_k &= \sum_{t=0}^{k} \nabla \hat{q}(S_t, A_t, \mathbf{w}_t) \prod_{i=t+1}^{k} \gamma_i \lambda_i \rho_i \\
&= \sum_{t=0}^{k-1} \nabla \hat{q}(S_t, A_t, \mathbf{w}_t) \prod_{i=t+1}^{k} \gamma_i \lambda_i \rho_i + \nabla \hat{q}(S_k, A_k, \mathbf{w}_k) \\
&= \gamma_k \lambda_k \rho_k \sum_{t=0}^{k-1} \nabla \hat{q}(S_t, A_t, \mathbf{w}_t) \prod_{i=t+1}^{k-1} \gamma_i \lambda_i \rho_i + \nabla \hat{q}(S_k, A_k, \mathbf{w}_k) \\
&= \gamma_k \lambda_k \rho_k \mathbf{z}_{k-1} + \nabla \hat{q}(S_k, A_k, \mathbf{w}_k)
\end{aligned}$$

which, changing the index from k to t, is the general accumulating trace update for action values:

$$\mathbf{z}_t = \gamma_t \lambda_t \rho_t \mathbf{z}_{t-1} + \nabla \hat{q}(S_t, A_t, \mathbf{w}_t)$$

Exercise 12.13

What are the dutch-trace and replacing-trace versions of off-policy eligibility traces for state-value and action-value methods?

My answer:

I guessed using the on-policy versions of all traces and the off-policy accumulating traces.

Dutch-trace, state-values:

$$\begin{aligned}
\mathbf{z}_{-1} &\doteq 0 \\
\mathbf{z}_t &= \rho_t \Big(\gamma_t \lambda_t \mathbf{z}_{t-1} + (1 - \alpha \gamma_t \lambda_t \mathbf{z}_{t-1}^{\top} \mathbf{x}_t) \mathbf{x}_t \Big)
\end{aligned}$$

Dutch-trace, action-values:

$$\begin{aligned}
\mathbf{z}_{-1} &\doteq 0 \\
\mathbf{z}_t &= \gamma_t \lambda_t \rho_t \mathbf{z}_{t-1} + (1 - \alpha \gamma_t \lambda_t \mathbf{z}_{t-1}^{\top} \mathbf{x}_t) \mathbf{x}_t
\end{aligned}$$

Replacing-trace, state-values:

$$\begin{cases}\rho_t & \text{if } x_{i,t}=1\\ \gamma_t\lambda_t\rho_t z_{i,t-1} & \text{otherwise.}\end{cases}$$

Replacing-trace, action-values:

$$\begin{cases}1 & \text{if } x_{i,t}=1\\ \gamma_t\lambda_t\rho_t z_{i,t-1} & \text{otherwise.}\end{cases}$$

Exercise 12.14

How might Double Expected Sarsa be extended to eligibility traces?

My answer:

I'm far from confident about this one. I think you have two different weight vectors, let's call them $\mathbf{w}^1, \mathbf{w}^2$, and two different eligibility traces, one for each weight vector. I think you do the update $\mathbf{w}_{t+1} = \mathbf{w}_t + \alpha\delta_t\mathbf{z}_t$ for both weight vectors on every timestep, but flip a coin for which eligibility trace gets updated each step. $\mathbf{w}^1$ is used in the expectation $\bar{V}_t(S_t)$ when computing δ_t for the $\mathbf{w}^2$ and vice versa. ϵ-greedy action selection would use $Q(S_t, A_t, \mathbf{w}^1) + Q(S_t, A_t, \mathbf{w}^2)$.