۸۱۰۱۰۱۵۲۰

استاد: دکتر فدایی و دکتر یعقوب زاده

هوش مصنوعی

تمرین سری ۶

(۱)

S	(۱,۱)	(۱,۲)	(۱,۳)	(۲,۱)	(۲,۲)	(۲,۳)
V_0	۰	۰	-۵	۰	۰	+۵
V_1	۰	۰	-۵	۰	۳,۶	+۵
V_2	۰	۲,۱۴۲	-۵	۲,۵۹۲	۳,۹۲۴	+۵

* مقادیر جدول با صورت سؤال متفاوت بود (جای ۵+ و ۵- ها)، صورت سؤال صحیح در نظر گرفته شد

* مقدار با پاداش یک سری از خانه‌ها مقتضی صفر بود که در برخی جاها نیاز به محاسبه آن دقت نمی‌شود

$$V_{i+1}(S) = \max_a \left(\sum_{s'} T(s,a,s')\left(R(s,a,s') + \gamma V_i(s')\right)\right)$$

$$V_1(1,2) = \max_a \left[\sum_{s'} T((1,2),a,s')\left(0 + \gamma V_0(s')\right)\right]$$

$$= \max_a \left[\sum_{s'} 0.9 \times \begin{matrix} T((1,2),L,s') \times 0 \\ T((1,2),R,(1,3)) \times V_0(1,3) \\ T((1,2),U,(1,3)) \times V_0(1,3) \\ T((1,2),D,(1,3)) \times V_0(1,3) \end{matrix} \right] = \max \begin{bmatrix} 0 \\ 0.9 \times 0.8 \times (-5) \\ 0.9 \times 0.1 \times (-5) \\ 0.9 \times 0.1 \times (-5) \end{bmatrix} = 0$$

$$V_1((2,2)) = \max_a \left(\sum_{s'} T((2,2),a,s')\left(0+\gamma V_0(s')\right)\right)$$

$$= \max \left(\sum_{s'} 0.9 \times \begin{matrix} T((2,2),L,s') \times 0 \\ T((2,2),R,(2,3)) \times V_0(2,3) \\ T((2,2),U,(2,3)) \times V_0(2,3) \\ T((2,2),D,(2,3)) \times V_0(2,3) \end{matrix} \right) = \max \begin{bmatrix} 0 \\ 0.9 \times 0.8 \times 5 \\ 0.9 \times 0.1 \times 5 \\ 0.9 \times 0.1 \times 5 \end{bmatrix} = 3.6$$

بعلت صفر بودن پاداش خانه‌های مجاور بقیه خانه‌ها، تغییری نمی‌کنند (صفر می‌مانند)

$$V_2((1,2)) = \max_a \left(\sum_{s'} T((1,2),a,s')\left(0+\gamma V_1(s')\right)\right) = \max \left(0.9 \times \begin{cases} T((1,2),L,(2,2))V_1(2,2) + T((1,2),L,(1,2))V_1(1,2) \\ T((1,2),R,(1,3))V_1(1,3) + T((1,2),R,(2,2))V_1(2,2) + T((1,2),R,(1,2))V_1(1,2) \\ T((1,2),D,(1,2))V_1(1,2) + T((1,2),D,(1,3))V_1(1,3) \\ T((1,2),U,(2,2))V_1(2,2) + T((1,2),U,(1,3))V_1(1,3) \end{cases}\right)$$

$$= \max \begin{bmatrix} 0.9 \times [0.1 \times 3.6 + 0.1 \times 0] \\ 0.9 \times [0.8 \times (-5) + 0.1 \times 3.6 + 0.1 \times 0] \\ 0.9 \times [0.8 \times 0 + 0.1 \times (-5)] \\ 0.9 \times [0.8 \times 3.6 + 0.1 \times (-5)] \end{bmatrix} = \max \begin{bmatrix} 0.324 \\ -3.276 \\ -0.45 \\ +2.142 \end{bmatrix} = 2.142$$

$$V_2((2,2)) = \max \left(\sum_{s'} T((2,2),a,s')\left(0+\gamma V_1(s')\right)\right) = \max \left(0.9 \times \begin{cases} T((2,2),L,(2,2)) \times V_1(2,2) + T((1,2),L,(1,2))V_1(1,2) \\ T((2,2),R,(2,3))V_1(2,3) + T((2,2),R,(2,2))V_1(2,2) + T((2,2),R,(1,2)) \times V_1(1,2) \\ T((2,2),D,(1,2))V_1(1,2) + T((2,2),D,(2,3))V_1(2,3) \\ T((2,2),U,(2,2))V_1(2,2) + T((2,2),U,(2,3))V_1(2,3) \end{cases}\right)$$

$$= \max \begin{bmatrix} 0.9 \times [0.1 \times 3.6 + 0.1 \times 0] \\ 0.9 \times [0.8 \times 5 + 0.1 \times 3.6 + 0.1 \times 0] \\ 0.9 \times [0.8 \times 0 + 0.1 \times 5] \\ 0.9 \times [0.8 \times 3.6 + 0.1 \times 5] \end{bmatrix} = \max \begin{bmatrix} 0.324 \\ 3.924 \\ 0.45 \\ 3.042 \end{bmatrix} = 3.924$$

$$V_2((2,1)) = \max \left(\sum_{s'} T((2,1),a,s')\left(0+\gamma V_1(s')\right)\right) = \max \left(0.9 \times \begin{cases} T((2,1),U,(2,2))V_1(2,2) \\ T((2,1),D,(2,2))V_1(2,2) \\ T((2,1),L,s') \times 0 \\ T((2,1),R,(2,2))V_1(2,2) \end{cases}\right)$$

$$= \max \begin{bmatrix} 0.9 \times 0.1 \times 3.6 \\ 0.9 \times 0.1 \times 3.6 \\ 0 \\ 0.9 \times 0.8 \times 3.6 \end{bmatrix} = \max \begin{bmatrix} 0.324 \\ 0.324 \\ 0 \\ 2.592 \end{bmatrix} = 2.592$$

۲) Policy

S	(1,1)	(1,2)	(1,3)	(2,1)	(2,2)	(2,3)
$\pi^*(S)$	U	U	-	R	R	-

۳) $V(1,1) = \frac{-5-5+5}{3} = 1,97$

$V(2,1) = \frac{5+5}{2} = 5$

۴] $V_{\pi}(s) \leftarrow V_{\pi}(s) + \alpha(R(s,\pi(s),s') + \gamma V_{\pi}(s') - V_{\pi}(s))$ جدید

$\hookrightarrow V_{\pi}(1,1) = V_{\pi}(1,1) + \alpha[R((1,1),R,(1,2)) + \gamma V_{\pi}(1,2) - V_{\pi}(1,1)]$

$= 0 + 0.1 \times [0 + 0.9 \times 0 - 0] = 0$

$V_{\pi}(1,2) \leftarrow V_{\pi}(1,2) + \alpha[R((1,2),R,(1,3)) + \gamma V_{\pi}(1,3) - V_{\pi}(1,2)]$ جدید

$= 0 + 0.1 \times [0 + 0.9 \times (-5) - 0] = -0.45$

$V_{\pi}(1,1) = V_{\pi}(1,1) + \alpha[R((1,1),R,(1,2)) + \gamma V_{\pi}(1,2) - V_{\pi}(1,1)]$

$= 0 + 0.1 \times [0 + 0.9 \times (-0.45) - 0] = -0.04$

$V_{\pi}(1,2) = V_{\pi}(1,2) + \alpha[R((1,2),R,(2,2)) + \gamma V_{\pi}(2,2) - V_{\pi}(1,2)]$

$= -0.45 + 0.1 \times [0 + 0.9 \times 0 + 0.45] = -0.405$

$V_{\pi}(2,2) = V_{\pi}(2,2) + \alpha[R((2,2),R,(2,3)) + \gamma V_{\pi}(2,3) - V_{\pi}(2,2)]$

$= 0 + 0.1 \times [0 + 0.9 \times 5 + 0] = 0.45$

S	(1,1)	(1,2)	(1,3)	(2,1)	(2,2)	(2,3)
V_0	0	0	-5	0	0	5
V_1	0	-0.45	-5	0	0	5
V_2	-0.04	-0.405	-5	0	0.45	5

الگوریتم DQN (Deep Q-Network) یکی از الگوریتم‌های یادگیری تقویتی عمیق است که توسط DeepMind توسعه یافته است. این الگوریتم در سال 2015 معرفی شد و توانست به موفقیت‌های قابل توجهی در بازی‌های آتاری دست یابد .

درDQN ، از شبکه‌های عصبی عمیق (Deep Neural Networks) برای تقریب تابع Q استفاده می‌شود. تابع Q به ارزیابی ارزش یک اقدام (Action) در یک وضعیت خاص (State) می‌پردازد. هدف DQN این است که یک سیاست (Policy) را بیاموزد که به حداکثر رساندن مجموع پاداش‌ها (Rewards) در طول زمان کمک کند.

اجزای اصلی:DQN

1) تجربه تکراری(Experience Replay) : در این روش، تجربه‌های عامل (Agent) در محیط ذخیره می‌شود و به صورت تصادفی از این حافظه برای به‌روزرسانی شبکه عصبی استفاده می‌شود. این کار به کاهش وابستگی زمانی و افزایش کارایی آموزش کمک می‌کند.

2) شبکه هدف DQN :(Target Network) از دو شبکه عصبی استفاده می‌کند - شبکه اصلی (Main Network)و شبکه هدف. شبکه هدف هر چند وقت یکبار به‌روزرسانی می‌شود تا مقادیر Q پایدارتر باشند.

کاربردهای:DQN

بازی‌های رایانه‌ای DQN :برای اولین بار در بازی‌های آتاری معرفی شد و توانست عملکردی هم‌سطح یا حتی بهتر از انسان داشته باشد. این نشان داد که الگوریتم می‌تواند سیاست‌های پیچیده‌ای را یاد بگیرد.

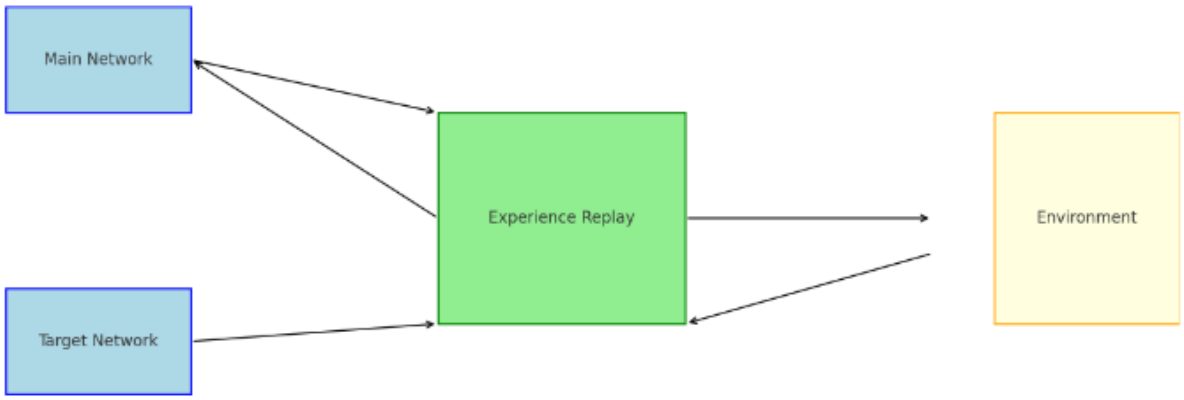
رباتیک DQN :در کنترل و بهینه‌سازی حرکت ربات‌ها مورد استفاده قرار می‌گیرد. مثلاً، در آموزش ربات‌ها برای انجام وظایف پیچیده مانند راه رفتن، پرش و تعامل با اشیاء.

سیستم‌های توصیه‌گر: این الگوریتم می‌تواند در بهینه‌سازی پیشنهادات به کاربران در سیستم‌های توصیه‌گر (Recommender Systems)مورد استفاده قرار گیرد.

مدیریت منابع DQN :در مدیریت منابع شبکه‌های کامپیوتری و تخصیص بهینه منابع برای بهبود عملکرد سیستم‌ها نیز کاربرد دارد.

نتیجه‌گیری:

الگوریتم DQN به عنوان یکی از الگوریتم‌های قدرتمند یادگیری تقویتی عمیق، نقش مهمی در پیشرفت‌های اخیر این حوزه ایفا کرده است. توانایی این الگوریتم در یادگیری سیاست‌های پیچیده از داده‌های خام و اعمال آن‌ها در محیط‌های متنوع، نشان‌دهنده پتانسیل بالای آن برای حل مسائل پیچیده و واقعی است.



این تصویر اجزا و مراحل اصلی الگوریتم DQN را نشان می‌دهد:

1. **شبکه عصبی اصلی (Main Network):**
   - در سمت چپ بالای تصویر قرار دارد و وظیفه تقریب تابع Q را بر عهده دارد.
2. **شبکه عصبی هدف (Target Network):**
   - در سمت چپ پایین تصویر قرار دارد و برای محاسبه مقادیر هدف Q استفاده می‌شود.
3. **تجربه تکراری (Experience Replay):**
   - در مرکز تصویر قرار دارد و تجربیات عامل در این حافظه ذخیره می‌شود.
   - این تجربیات به صورت تصادفی برای به‌روزرسانی شبکه‌های عصبی استفاده می‌شوند.
4. **محیط (Environment):**
   - در سمت راست تصویر قرار دارد و عامل با آن تعامل می‌کند.

مراحل تعامل اجزا:

- **انتخاب اقدام و تعامل با محیط:**
  - عامل یک اقدام را بر اساس وضعیت فعلی و سیاست ϵحریصانه انتخاب می‌کند.
  - اقدام در محیط اعمال می‌شود و پاداش و وضعیت جدید مشاهده می‌شود.
  - این تجربه (s, a, r, s')در حافظه تکراری ذخیره می‌شود.(پیکان از "Environment" به "Experience Replay")

- **به‌روزرسانی شبکه‌های عصبی:**
  - تجربیات به صورت تصادفی از حافظه تکراری نمونه‌گیری می‌شوند.
  - شبکه عصبی اصلی (Main Network) با استفاده از این نمونه‌ها به‌روزرسانی می‌شود (پیکان از "Experience Replay"به"Main Network")
  - شبکه هدف (Target Network) هر چند وقت یک‌بار به‌روزرسانی می‌شود تا مقادیر Q پایدارتر باشند (پیکان از "Main Network" به"Target Network")

این تصویر به خوبی نشان می‌دهد که چگونه الگوریتم DQN با استفاده از شبکه‌های عصبی و حافظه تجربه تکراری به یادگیری سیاست‌های بهینه می‌پردازد .