Rizqika Mulia Pratama
13522126

Reinforcement Learning

1. Cara kerja algoritma

- Q-Learning

  Algoritma Q-learning adalah salah satu metode reinforcement learning yang digunakan untuk menemukan policy optimal dalam suatu environment. Policy ini mengarahkan agent untuk mengambil action terbaik berdasarkan state saat ini, dengan tujuan memaksimalkan reward jangka panjang.

  1. Inisialisasi Q-table

     Q-learning menggunakan q-table untuk menyimpan q-value untuk setiap pasangan state dan action. Tabel ini diinisialisasi dengan nilai nol atau nilai random pada awalnya. Q-value merepresentasikan estimasi reward yang bisa diperoleh jika agent memilih action tertentu pada state tertentu.

  2. Memilih action dengan epsilon-greedy

     Pada setiap step, agent harus memutuskan apakah akan mengeksplorasi (memilih aksi secara acak) atau mengeksplorasi (memilih aksi terbaik berdasarkan q-value saat ini). Ini dikenal sebagai policy epsilon-greedy:

     - Eksplorasi: Dengan probabilitas epsilon, agent akan memilih aksi secara acak.
     - Eksploitasi: Dengan probabilitas 1 - epsilon, agent akan memilih aksi dengan q-value tertinggi pada state saat ini.

  3. Update Q-value

     Setelah agent melakukan aksi dan berpindah ke state berikutnya, q-value di-update menggunakan rumus berikut:

     $$Q(s,a) \leftarrow Q(s,a) \ + \ \alpha(r \ + \ \gamma \cdot a' maxQ(s', a') \ - \ Q(s,a))$$

     - Q(s,a) adalah q-value saat ini untuk state s dan action a.
     - α adalah learning rate (seberapa cepat agent belajar).
     - r adalah reward yang diperoleh setelah melakukan action a.
     - γ adalah discount factor (seberapa jauh reward masa depan dipertimbangkan).
     - $maxQ(s', a')$ adalah estimasi q-value tertinggi untuk state berikutnya s’

  4. Episode dan akhir game

     Proses ini diulangi untuk setiap episode, di mana agent terus belajar hingga mencapai kondisi terminal, seperti menang, kalah, atau mencapai jumlah episode maksimal. Setelah cukup banyak episode,

q-table akan terisi dengan nilai-nilai optimal, dan agent dapat memilih aksi terbaik untuk setiap state.

- SARSA

  SARSA (State-Action-Reward-State-Action) adalah algoritma reinforcement learning yang mirip dengan q-learning, tetapi memiliki perbedaan pada cara mengupdate q-value.

  5. Inisialisasi Q-table

     Sama seperti q-learning, SARSA menggunakan q-table untuk menyimpan q-value bagi setiap pasangan state dan action. Q-table diinisialisasi dengan nol atau nilai random.

  6. Memilih action dengan epsilon-greedy

     Seperti pada q-learning, SARSA juga menggunakan kebijakan epsilon-greedy untuk memilih action:
     - Eksplorasi: Memilih aksi secara acak dengan probabilitas epsilon.
     - Eksploitasi: Memilih aksi terbaik berdasarkan q-value dengan probabilitas 1 - epsilon.

  7. Update Q-value

     Perbedaan utama SARSA dengan q-learning adalah pada cara mengupdate q-value. SARSA memperbarui q-value berdasarkan aksi yang sebenarnya dipilih oleh agent di state berikutnya, bukan aksi terbaik seperti pada q-learning:

     $$Q(s,a) \leftarrow Q(s,a) \ + \ \alpha(r \ + \ \gamma \cdot Q(s',a') \ - \ Q(s,a))$$

     - $Q(s',a')$ adalah q-value untuk state dan action berikutnya yang sebenarnya dipilih, bukan nilai maksimal dari q-value seperti pada Q-learning.

  8. Episode dan akhir game

     Proses ini diulangi hingga agent mencapai state terminal atau episode terakhir. SARSA juga bekerja dengan mengisi q-table melalui iterasi, dan pada akhirnya, agent dapat memilih action terbaik berdasarkan q-value yang dipelajari.

2. Perbandingan Q-learning dan SARSA

Test	Algoritma	Final Score	Total Episodes
1	Q-Learning	-204	6
	SARSA	-214	16

Rizqika Mulia Pratama
13522126

2	Q-Learning	573	136
	SARSA	-206	8
3	Q-Learning	553	156
	SARSA	569	140
4	Q-Learning	-222	24
	SARSA	562	149
5	Q-Learning	-216	18
	SARSA	-216	18
6	Q-Learning	536	179
	SARSA	562	149
7	Q-Learning	567	142
	SARSA	582	149

Hasil perbandingan pada environment yang sama antara algoritma Q-learning dan SARSA menunjukan variasi kinerja yang menarik. Dalam beberapa test, Q-learning berhasil mencapai kemenangan dengan lebih cepat dan mencetak skor tinggi, seperti terlihat pada test 2 dan 7, dimana Q-learning unggul secara signifikan. Namun, Q-learning juga mengalami kekalahan lebih sering dibandingkan SARSA, yang mencerminkan sifat algoritma ini yang lebih agresif dalam eksplorasi. Di sisi lain, SARSA menunjukkan performa yang lebih stabil dan konsisten, seperti yang terlihat pada test 3, 4, dan 6, dimana SARSA mampu mencapai kemenangan dengan skor positif secara lebih teratur. Konsistensi ini disebabkan pendekatan SARSA yang lebih konservatif dalam memilih aksi, karena algoritma ini belajar dari aksi yang benar-benar diambil (on-policy). Meskipun demikian, ada beberapa situasi dimana SARSA kurang efektif, seperti pada test 2, dimana Q-learning justru berhasil mencapai hasil yang jauh lebih baik. Perbedaan utama antara kedua algoritma ini terletak pada sifat off-policy dari Q-learning yang memungkinakn eksplorasi yang lebih luas, namun dengan risiko hasil yang kurang stabil, sedangkan SARSA lebih berfokus pada kestabilan dan hasil yang lebih aman, tetapi kadang memerlukan lebih banyak waktu untuk mencapai hasil optimal.