1. Cara Kerja Algoritma Q-Learning dan SARSA Berdasarkan Kode

Q-Learning

Algoritma Q-Learning adalah algoritma **off-policy** yang bertujuan untuk memaksimalkan total reward dengan menemukan kebijakan optimal.

- **Inisialisasi Q-Table:** Q-table diinisialisasi dengan nilai nol untuk setiap pasangan state-action. Dalam game ini, ada 10 state (sesuai panjang papan) dan 2 aksi ('kiri' dan 'kanan'), jadi Q-table berbentuk (10, 2).
- **Epsilon-Greedy Policy:** Pada setiap langkah dalam sebuah episode, agen memilih aksi berdasarkan kebijakan epsilon-greedy. Ini berarti agen memilih aksi acak dengan probabilitas epsilon (untuk eksplorasi), atau memilih aksi dengan nilai Q tertinggi (greedy) dengan probabilitas 1-epsilon (untuk eksploitasi).
- **Update Q-Value:** Setelah agen melakukan aksi dan mendapatkan reward, nilai Q diperbarui menggunakan rumus:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \times \left(r + \gamma \times \max_{a'} Q(s',a') - Q(s,a)\right)$$

- **Penurunan Epsilon:** Epsilon dikurangi secara bertahap (epsilon decay) di setiap episode agar agen semakin sering mengeksploitasi kebijakan optimal.

SARSA (State-Action-Reward-State-Action)

Algoritma SARSA adalah algoritma **on-policy** yang memperbarui nilai Q berdasarkan aksi yang dipilih oleh agen berdasarkan kebijakan yang sedang digunakan.

- **Inisialisasi Q-Table:** Sama seperti pada Q-Learning, Q-table diinisialisasi dengan nilai nol untuk setiap pasangan state-action.
- **Epsilon-Greedy Policy:** Aksi pertama dipilih berdasarkan kebijakan epsilon-greedy, mirip dengan Q-Learning. Agen dapat memilih aksi acak (eksplorasi) atau aksi dengan nilai Q tertinggi (eksploitasi).
- **Update Q-Value:** Setelah agen mengambil aksi pertama dan bergerak ke state berikutnya, agen mendapatkan reward dan memilih aksi berikutnya dari state baru ini. Q-value kemudian diperbarui menggunakan rumus:

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

2. Analisis Hasil

Hasil dari kedua algoritma, Q-Learning dan SARSA, menunjukkan kinerja yang sangat mirip dalam konteks permainan sederhana ini. Selama proses pelatihan, rata-rata reward yang diperoleh oleh Q-Learning adalah 1017.87, sedangkan SARSA mencapai rata-rata reward sebesar 1017.92. Perbedaan ini sangat kecil, yang mungkin menunjukkan bahwa kedua algoritma mampu belajar dengan efisien dan menemukan strategi yang hampir sama optimalnya. Reward tertinggi yang dicapai selama pelatihan oleh Q-Learning adalah 1076.00, sementara SARSA sedikit lebih tinggi, yaitu 1080.00. Untuk reward terendah, Q-Learning mencapai -498.00, sedikit lebih rendah dibandingkan dengan SARSA yang mencapai -492.00. Rata-rata reward setelah evaluasi untuk kedua algoritma adalah identik, yaitu 1028.00, menandakan bahwa setelah fase pelatihan selesai, keduanya mungkin mampu mempertahankan kinerja optimal yang setara ketika diterapkan kembali dalam lingkungan permainan yang sama.

Meskipun hasilnya sangat mirip, ada beberapa perbedaan halus yang bisa diamati dari nilai-nilai dalam Q-Table yang dihasilkan oleh kedua algoritma. Pada Q-Table yang dihasilkan oleh Q-Learning, terlihat bahwa nilai Q lebih tinggi di beberapa state, terutama saat mendekati titik apel (di posisi 9). Ini sesuai dengan karakteristik Q-Learning yang merupakan algoritma off-policy, yang cenderung lebih agresif dan optimis. Q-Learning memperbarui nilai Q berdasarkan nilai maksimum yang bisa dicapai di state berikutnya tanpa memperhitungkan apakah aksi tersebut benar-benar akan diambil atau tidak. Oleh karena itu, Q-Learning lebih banyak mencoba berbagai opsi untuk mencapai reward maksimum, yang mungkin menyebabkan nilai Q di beberapa state menjadi lebih tinggi.

Di sisi lain, Q-Table yang dihasilkan oleh SARSA menunjukkan nilai yang sedikit lebih rendah di beberapa state dibandingkan dengan Q-Learning. Ini mencerminkan pendekatan SARSA yang lebih konservatif dan realistis sebagai algoritma on-policy. SARSA memperbarui nilai Q berdasarkan aksi yang sebenarnya diambil oleh agen, sesuai dengan kebijakan yang sedang diterapkan. Hal ini membuat SARSA lebih berhati-hati dalam mengeksplorasi jalur-jalur yang ada dan lebih mempertimbangkan risiko dari jalur yang benar-benar ditempuh oleh agen. Akibatnya, nilai Q yang dihasilkan lebih stabil dan mungkin sedikit lebih rendah, tetapi juga lebih mencerminkan kenyataan dari keputusan-keputusan yang diambil oleh agen.

Perbedaan hasil ini, meskipun halus, dapat dijelaskan oleh kondisi permainan yang relatif sederhana dan deterministik. Dalam permainan ini, agen hanya dihadapkan pada pilihan dua aksi, yaitu 'kiri' atau 'kanan', di setiap state, dan tidak ada banyak ketidakpastian atau variabilitas dalam lingkungan. Oleh karena itu, perbedaan pendekatan antara Q-Learning yang lebih optimis dan SARSA yang lebih konservatif mungkin tidak terlalu terlihat. Jika permainan ini berada dalam lingkungan yang lebih kompleks atau tidak pasti, perbedaan antara pendekatan off-policy dan on-policy ini mungkin bisa menyebabkan variasi hasil yang lebih signifikan. Namun, dalam konteks permainan sederhana ini, kedua algoritma berhasil belajar untuk melakukan strategi

optimal yang sama. Hal ini ditunjukkan oleh jalur yang diambil oleh pemain yang identik dan nilai reward yang sangat serupa, menegaskan bahwa baik Q-Learning maupun SARSA sama-sama efektif dalam lingkungan permainan ini.