Reinforcement Learning

Razi Rachman Widyadhana - 13523004

1. Cara kerja Q-Learning

Q-Learning adalah algoritma *off-policy* reinforcement learning yang mempelajari kebijakan optimal dengan memperbarui nilai Q berdasarkan aksi terbaik yang mungkin diambil.

Jika dijabarkan, berikut tahapan algoritma Q-Learning:

1. Inisialisasi nilai Q.

   Langkah pertama adalah menginisialisasi matriks Q dengan nilai awal (biasanya nol atau kecil) untuk semua pasangan state-action (s, a).

   Tujuannya untuk menyediakan dasar untuk pembelajaran nilai Q.

2. Pilih aksi menggunakan kebijakan eksplorasi.

   Pilih aksi a di state s menggunakan kebijakan seperti ϵ-greedy, yang menyeimbangkan eksplorasi (memilih aksi acak) dan eksploitasi (memilih aksi dengan Q tertinggi).

   Tujuannya untuk memungkinkan agen menjelajahi lingkungan sambil memanfaatkan pengetahuan yang ada.

3. Lakukan aksi dan amati reward serta state berikutnya.

   Eksekusi aksi a, terima reward r, dan pindah ke state berikutnya s'. Proses ini dilakukan sesuai dinamika lingkungan.

   Tujuannya untuk mengumpulkan pengalaman untuk pembaruan Q.

4. Perbarui nilai Q.

   Perbarui nilai Q menggunakan rumus:

   $$Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma \max_{a'} Q(s', a') - Q(s, a)]$$

   di mana α adalah learning rate, γ adalah faktor diskon, dan $\max_{a'} Q(s', a')$ adalah estimasi nilai optimal state berikutnya.

   Tujuannya untuk meningkatkan estimasi Q berdasarkan aksi terbaik hipotetis.

5. Ulangi hingga konvergensi.

   Ulangi langkah 2-4 untuk semua state hingga nilai Q konvergen atau mencapai jumlah iterasi maksimum, menghasilkan kebijakan optimal.

   Tujuannya untuk mencapai solusi stabil untuk pengambilan keputusan.

2. Cara kerja SARSA

SARSA (State-Action-Reward-State-Action) adalah algoritma *on-policy* reinforcement learning yang memperbarui nilai Q berdasarkan aksi yang benar-benar diambil oleh agen.

Jika dijabarkan, berikut tahapan algoritma SARSA:

1. Inisialisasi nilai Q.

   Langkah pertama adalah menginisialisasi matriks Q dengan nilai awal (biasanya nol atau kecil) untuk semua pasangan state-action (s, a).

   Tujuannya untuk menyediakan dasar untuk pembelajaran nilai Q.

2. Pilih aksi awal menggunakan kebijakan.

   Pilih aksi a di state s menggunakan kebijakan saat ini, seperti ϵ-greedy, yang menyeimbangkan eksplorasi dan eksploitasi.

   Tujuannya untuk menentukan langkah awal berdasarkan kebijakan aktif.

3. Lakukan aksi dan amati reward serta state berikutnya.

   Eksekusi aksi a, terima reward r, pindah ke state berikutnya s', dan pilih aksi berikutnya a' sesuai kebijakan.

   Tujuannya untuk mengumpulkan pengalaman berurutan untuk pembaruan.

4. Perbarui nilai Q.

   Perbarui nilai Q menggunakan rumus:

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

   di mana α adalah learning rate, γ adalah faktor diskon, dan $Q(s', a')$ adalah nilai Q untuk aksi berikutnya yang dipilih.

   Tujuannya untuk menyesuaikan Q berdasarkan pengalaman aktual agen.

5. Ulangi hingga konvergensi.

   Ulangi langkah 2-4 untuk semua state-action pair hingga nilai Q konvergen atau mencapai iterasi maksimum, menghasilkan kebijakan yang konsisten dengan perilaku agen.

   Tujuannya untuk mencapai solusi stabil berdasarkan kebijakan on-policy.

3. Perbedaan fundamental off-policy dan on-policy

Perbedaan fundamental antara *off-policy* dan *on-policy* terletak pada cara pembelajaran kebijakan. *Off-policy*, seperti Q-Learning, belajar dari aksi optimal hipotetis (bukan aksi yang diambil), memungkinkan agen mengevaluasi kebijakan terbaik secara independen dari kebijakan eksplorasi sehingga lebih fleksibel dan cenderung mengambil risiko lebih besar. Sebaliknya, *on-policy*, seperti SARSA, belajar dari aksi yang benar-benar dieksekusi sesuai kebijakan saat ini, membuatnya lebih

sesuai dengan perilaku aktual agen dan cenderung lebih konservatif, terutama dalam lingkungan berisiko.

4. Perbandingan

Dalam konteks Wumpus World, perbandingan antara Q-Learning dan SARSA menunjukkan karakteristik berikut berdasarkan hasil:

1. Kecepatan Konvergensi: Q-Learning menunjukkan konvergensi lebih cepat dengan menang pada episode 32, sedangkan SARSA memerlukan satu episode lebih lama (episode 33). Ini mencerminkan sifat *off-policy* Q-Learning yang mengeksploitasi aksi optimal lebih agresif, mempercepat pembelajaran dibandingkan SARSA yang terikat pada kebijakan saat ini.

2. Final policy: Kedua algoritma menghasilkan Final policy yang identik, baik saat belum membawa gold maupun setelah membawa gold, dengan pola gerakan yang sama. Perbedaan *on-policy* vs *off-policy* tidak memengaruhi struktur Final policy, kemungkinan karena lingkungan kecil membatasi variasi kebijakan.

3. Jalur yang Ditempuh: Kedua algoritma memiliki skor risiko yang sama (28.00), menunjukkan jalur yang ditempuh tidak berbeda signifikan dalam hal risiko. Meskipun secara teori Q-Learning cenderung mengambil rute berisiko lebih tinggi karena sifat *off-policy*-nya, area kecil Wumpus World memaksa agen melewati path berisiko (seperti dekat pit atau wumpus) terlepas dari algoritma sehingga perbedaan risiko tidak terlihat jelas.