

Tugas RL 02 Kelompok 18

Nama : - ENJELIN YENI DWI LESTARI (ID 20_ALAN TURING)
- GRACE D. SITANGGANG (ID 19_VISIONER)
- DHEA ANGGITA (ID 4_PERSEVERE)
- SYAFRAJI (ID _BETTER)
- JAVANDRA SAVINSAKA (ID _ATLAS)

STUDENT ACTIVITIES

1. Kasus di environment OpenAI yang bisa diselesaikan dengan Monte Carlo

Kasus environment OpenAI yang diselesaikan dengan Monte Carlo yang kami ambil adalah BlackJack Environment .

2. Pemahaman dari mekanisme kerja dari program tersebut

- a. **Summary dari BlackJack**

Blackjack adalah permainan kartu yang dimainkan melawan dealer. Pada awal putaran, baik pemain dan dealer dibagikan 2 kartu. Pemain hanya dapat melihat salah satu kartu dealer. Tujuan dari permainan ini adalah untuk mendapatkan nilai kartu kita sedekat mungkin dengan 21, tanpa melewati 21. Nilai setiap kartu tercantum di bawah ini.

- 10/Jack/Ratu/Raja → 10
- 2 sampai 9 → Nilai yang sama dengan kartu
- As → 1 atau 11 (Pilihan pemain)

- Agent : Pemain
- Environment : Himpunan semua kemungkinan situasi yang dapat berinteraksi dengan agen, tindakan yang tersedia dalam setiap situasi, dan hasil (rewards + punishments) yang terkait dengan masing-masing situasi ini. Di Blackjack, ini adalah kumpulan semua kemungkinan tangan pemain, kartu up dealer, aksi pemain (hit atau stand), dan hasil (win/lose/tie).
- Objective :
 - Membuat jumlah kartu lebih besar dari dealer tanpa melebihi 21.
 - Semua kartu dihitung sebagai 10, dan kartu As dapat dihitung sebagai 1 atau 11.
- States :
 - jumlah saat ini (12-21)
 - kartu yang ditunjukkan dealer (As, 2-10)
 - apakah saya punya kartu As yang bisa digunakan?
- Reward :
 - +1 untuk menang, 0 untuk seri, -1 untuk kalah (bukan

- diskon)
- Actions :
 - stand (menghentikan menerima kartu), hit (menerima kartu lain)
 - Policy :
 - stand jika jumlah dari kartu 20 atau 21, jika tidak hit

Jika pemain memiliki kurang dari 21, mereka dapat memilih untuk "hit" dan menerima kartu acak dari dek. Mereka juga dapat memilih untuk "stand" dan menyimpan kartu yang mereka miliki. Jika pemain melebihi 21, mereka "bust" dan secara otomatis kalah. Jika pemain memiliki tepat 21, mereka secara otomatis menang. Jika tidak, pemain menang jika mereka lebih dekat ke 21 daripada dealer.

b. Bagaimana Komponen Ini Bekerja Bersama Di Blackjack

- Putaran Blackjack dimulai: 2 kartu dibagikan kepada pemain dan dealer, dan agen hanya melihat kartunya dan salah satu kartu dealer. Lingkungan memodelkan ini dengan mengirimkan agen status awal (nilai tangan pemain + nilai kartu atas dealer).



- Pemain memiliki dua tindakan: hit atau stand. Agen memilih tindakan berdasarkan status saat ini menggunakan kebijakannya



- State + action yang dipilih oleh agen dikirim kembali ke environment.



- Environment secara internal memproses tindakan agen yang diberikan keadaan. Ini menangani kartu baru untuk "hit". Ini menghitung siapa yang memenangkan ronde jika sesuai (emain berdiri, Blackjack!, atau gagal).



- Jika ronde sekarang berakhir, environment mengirim agen **new state** yang mewakili ronde Blackjack berikutnya, bersama dengan hadiah dari ronde sebelumnya. Agen menggunakan hasil ini untuk memperbarui **policynya**.



- Jika ada lebih banyak actions yang harus dilakukan agen di ronde saat ini, environment mengirim agen status dengan nilai kartu pemain yang diperbarui dan reward +\$0 karena ronde belum berakhir.



c. Hands on

Untuk eksplorasi langsung dengan code dari penerapan Monte Carlo pada studi kasus [BlackJack Environment](#).

3. Jelaskan perbedaan antara DP dan MC dalam menyelesaikan Frozen Lake Problem

a. Dynamic Programming

Agent berjalan menuju tujuan yang jalurnya berupa es. Namun ada beberapa lubang. Esnya licin, sehingga agent tidak bisa selalu bergerak sesuai arah yang diinginkannya.

- Satu episode berhenti ketika agent mencapai tujuan (goal)
- Agent mendapat 1 reward ketika mencapai tujuan (goal) dan 0 untuk kondisi yang lain.
- Jika masuk ke lubang, agent hanya bisa bergerak searah dengan arah sebelum masuk lubang (peluang 1/3), dan kedua arah tegak lurus (masing-masing peluangnya 1/3).

b. Monte Carlo

- Seluruh episode dipertimbangkan dalam MC.
- Hanya satu pilihan setiap perpindahan state di MC, sedangkan DP mempertimbangkan semua probabilitas transisi pada setiap perpindahan state.
- Estimasi-estimasi untuk semua state adalah independent di MC, tidak bootstrap.
- Waktu yang dibutuhkan untuk mengestimasi suatu state tidak bergantung pada jumlah total state.