

Laporan Praktikum Kontrol Cerdas

Nama : Kevin Bayu Pradana
NIM : 224308009
Kelas : TKA - 6A
Akun Github (Tautan) : <https://github.com/Kevinbyu84>
Student Lab Assistant : Rizky Putri Ramadhani

1. Judul Percobaan

Week 4: Reinforcement Learning for Autonomous Control

2. Tujuan Percobaan

Tujuan dari praktikum “Reinforcement Learning for Autonomous Control”, mahasiswa diharapkan mampu:

1. Memahami Memahami konsep dasar Reinforcement Learning (RL) dalam sistem kendali.
2. Mengimplementasikan agen RL menggunakan algoritma Deep Q-Network (DQN).
3. Menggunakan OpenAI Gym sebagai simulasi lingkungan untuk pelatihan RL.
4. Melatih dan menguji agen RL untuk mengontrol lingkungan secara otonom.
5. Menggunakan GitHub untuk version control dan dokumentasi praktikum.

3. Landasan Teori

- **Reinforcement Learning (RL)** adalah cabang dari machine learning di mana agen belajar mengambil tindakan dalam lingkungan untuk memaksimalkan reward kumulatif. Proses pembelajaran dalam RL terdiri dari iterasi antara agen dan lingkungan, di mana agen menerima umpan balik dalam bentuk reward berdasarkan tindakan yang diambil. RL memiliki aplikasi luas dalam berbagai bidang seperti robotika, permainan video, dan pengendalian sistem dinamis.
- **Deep Q-Network (DQN)** adalah pengembangan dari algoritma Q-Learning yang menggunakan jaringan saraf tiruan (neural network) sebagai fungsi aproksimasi untuk memetakan state ke nilai Q. DQN diperkenalkan oleh DeepMind dan berhasil mencapai performa manusia dalam berbagai permainan Atari. Algoritma ini menggunakan dua teknik utama untuk meningkatkan stabilitas dan konvergensi, yaitu pengalaman replay (experience replay) dan target network.
- **OpenAI Gym** adalah toolkit open-source yang dirancang untuk membangun dan mengevaluasi algoritma reinforcement learning. Platform ini menyediakan berbagai environment standar yang memungkinkan pengujian dan perbandingan berbagai metode RL. OpenAI Gym memudahkan replikasi eksperimen dan pengembangan algoritma baru dalam komunitas penelitian RL.
- Environment **CartPole** adalah salah satu lingkungan klasik di OpenAI Gym yang digunakan untuk menguji algoritma RL. Tugas dalam CartPole adalah menyeimbangkan tiang (pole) yang terpasang pada kereta (cart) di sepanjang jalur horizontal. Agen menerima reward +1 untuk setiap langkah di mana tiang tetap seimbang, dan episode berakhir jika tiang miring lebih dari sudut tertentu atau jika cart bergerak keluar dari batas jalur.
- Environment **MountainCar** adalah lingkungan di OpenAI Gym di mana agen mengendalikan mobil di antara dua bukit. Tugasnya adalah mendorong mobil ke puncak bukit di sebelah kanan. Tantangan utama di lingkungan ini adalah mobil tidak memiliki daya dorong yang cukup untuk langsung mencapai puncak, sehingga agen harus mempelajari strategi osilasi untuk membangun momentum. Reward diberikan hanya ketika agen berhasil mencapai puncak bukit, menjadikannya tugas eksplorasi yang sulit.

4. Analisis dan Diskusi

Analisis Hasil:

- **Bagaimana performa agen dalam mengontrol environment CartPole?**
Agen RL belajar menyeimbangkan tiang pada CartPole melalui banyak percobaan. Awalnya, sering gagal, namun seiring waktu agen menjadi lebih baik dan mampu menyeimbangkan tiang lebih lama.
- **Bagaimana perubahan parameter (misal: gamma, epsilon, learning rate) mempengaruhi kinerja agen?**
 - **Gamma (Discount Factor)** Semakin tinggi nilai gamma, agen mempertimbangkan reward jangka panjang. Jika gamma terlalu rendah, agen hanya fokus pada reward langsung.
 - **Epsilon (Exploration Rate)** Nilai epsilon tinggi membuat agen mencoba berbagai aksi secara acak untuk menemukan strategi terbaik. Secara bertahap, nilai epsilon diturunkan agar agen lebih sering memilih aksi yang terbukti efektif.
 - **Learning Rate** menentukan seberapa cepat agen mengubah perkiraan nilainya. Nilai yang terlalu tinggi bisa membuat pembelajaran tidak stabil, sedangkan nilai yang terlalu rendah membuat proses belajar menjadi lambat.
- **Apa tantangan yang muncul selama pelatihan agen RL?**
Agen kadang mengalami fluktuasi reward yang menyebabkan pembelajaran tidak stabil. Menentukan kapan harus mencoba aksi baru (eksplorasi) dan kapan harus menggunakan aksi yang sudah diketahui efektif (eksploitasi) merupakan tantangan tersendiri. Proses pelatihan, terutama dengan algoritma seperti DQN, membutuhkan banyak waktu dan sumber daya komputasi.

Diskusi:

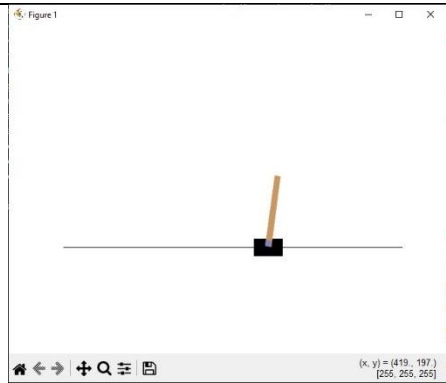
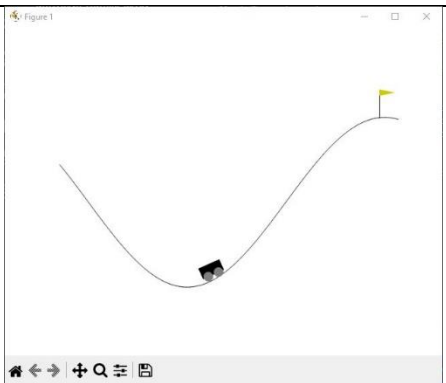
- **Perbedaan Utama antara Reinforcement Learning dan Supervised Learning:**
Supervised learning menggunakan data yang sudah dilabeli, sedangkan RL belajar dari interaksi langsung dengan lingkungan tanpa label. Dalam supervised learning, model belajar dari contoh yang benar. Di RL, agen belajar melalui trial-and-error dengan mendapatkan reward atau hukuman. RL berfokus pada mendapatkan reward kumulatif maksimal, sedangkan supervised learning fokus pada akurasi prediksi.
- **Optimasi Strategi Eksplorasi dan Eksploitasi:**
Eksplorasi, Agen mencoba berbagai aksi untuk menemukan strategi terbaik.
Eksploitasi, Agen menggunakan strategi yang sudah terbukti memberikan reward tinggi.
Optimalisasi, Pengurangan nilai epsilon secara bertahap (epsilon decay) membantu menyeimbangkan antara eksplorasi (mencoba hal baru) dan eksploitasi (menggunakan strategi yang sudah diketahui efektif).
- **Potensi Aplikasi RL dalam Sistem Kendali Nyata:**
Robotika, Untuk mengontrol pergerakan robot dalam navigasi atau manipulasi objek.
Kendaraan Otonom, Mengoptimalkan keputusan pada mobil self-driving.

5. Assignment

Tahap awal pengerjaan assignment dimulai dengan melakukan studi literatur untuk memahami konsep dasar dan teknik yang digunakan dalam Reinforcement Learning (RL), khususnya algoritma **Deep Q-Network (DQN)**, serta lingkungan yang disediakan oleh **OpenAI Gym**, termasuk **CartPole-v1** dan **MountainCar-v0**. Pada tahap ini, dilakukan modifikasi pada program latih dan uji untuk menyesuaikan environment dari **CartPole-v1** menjadi **MountainCar-v0** dan menambahkan fitur baru untuk meningkatkan stabilitas pelatihan. Untuk meningkatkan stabilitas pelatihan, fitur **target network** ditambahkan dan diperbarui setiap **10 episode**. Target network membantu mengurangi perubahan drastis dalam estimasi Q-value dengan memisahkan jaringan utama (yang belajar secara aktif) dari jaringan target (yang menjadi acuan tetap dalam waktu tertentu). Dengan pembaruan target network secara berkala, pelatihan menjadi lebih stabil dan akurat. Agen RL menunjukkan performa yang cukup baik pada environment CartPole, tetapi kesulitan mengontrol MountainCar secara efektif karena skor pelatihan masih tinggi. Hal ini mengindikasikan bahwa agen membutuhkan lebih banyak eksplorasi atau modifikasi parameter agar bisa mencapai hasil yang lebih baik.

6. Data dan Output Hasil Pengamatan

Data dan hasil yang diperoleh selama percobaan.

No	Variabel	Hasil Pengamatan
1	Data hasil uji environment CartPole	<pre> Target Network updated! Episode: 92/100, Score: 13, Epsilon: 0.41 Episode: 93/100, Score: 15, Epsilon: 0.40 Episode: 94/100, Score: 25, Epsilon: 0.40 Episode: 95/100, Score: 14, Epsilon: 0.40 Episode: 96/100, Score: 16, Epsilon: 0.39 Episode: 97/100, Score: 15, Epsilon: 0.39 Episode: 98/100, Score: 69, Epsilon: 0.38 Episode: 99/100, Score: 29, Epsilon: 0.38 Episode: 100/100, Score: 18, Epsilon: 0.38 </pre>
2	Hasil envirointment CartPole.	
3	Data hasil uji environment MountainCar	<pre> Episode: 1/50, Score: 199, Epsilon: 1.00 Target Network updated! Episode: 2/50, Score: 199, Epsilon: 0.99 Episode: 3/50, Score: 199, Epsilon: 0.99 Episode: 4/50, Score: 199, Epsilon: 0.99 Episode: 5/50, Score: 199, Epsilon: 0.98 Episode: 6/50, Score: 199, Epsilon: 0.98 Episode: 7/50, Score: 199, Epsilon: 0.97 Episode: 8/50, Score: 199, Epsilon: 0.97 Episode: 9/50, Score: 199, Epsilon: 0.96 Episode: 10/50, Score: 199, Epsilon: 0.96 Episode: 11/50, Score: 199, Epsilon: 0.95 Target Network updated! Episode: 12/50, Score: 199, Epsilon: 0.95 </pre>
4	Hasil envirointment MountainCar	

7. Kesimpulan

- Agen RL menunjukkan peningkatan kemampuan di lingkungan CartPole setelah beberapa episode pelatihan.
- Parameter gamma, epsilon, dan learning rate memengaruhi stabilitas dan kinerja agen:
- Gamma yang rendah membuat agen mengabaikan reward jangka panjang.
- Epsilon yang tinggi mendorong eksplorasi berlebih.
- Learning rate yang sesuai penting untuk menjaga kestabilan pembelajaran.
- Tantangan utama meliputi fluktuasi reward, kestabilan pelatihan, dan keseimbangan eksplorasi-eksploitasi.
- Performa agen di lingkungan MountainCar masih memerlukan perbaikan untuk mengatasi tantangan eksplorasi yang kompleks.

8. Saran

- Eksplorasi teknik peningkatan stabilitas seperti Prioritized Experience Replay atau Double DQN.
- Sesuaikan parameter gamma, epsilon decay, dan learning rate secara dinamis sesuai kebutuhan lingkungan.
- Gunakan hardware akselerasi seperti GPU untuk mempercepat pelatihan dan memungkinkan eksperimen yang lebih kompleks.

9. Daftar Pustaka

1. Abadi, Arulkumaran, K., Deisenroth, M. P., Brundage, M. & Bharath, A. A., 2017. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6), pp. 26-38.
2. Barto, A. G., Sutton, R. S. & Anderson, C. W., 1983. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-13(5), pp. 834-846.
3. Mnih, V. et al., 2015. Human-level control through deep reinforcement learning. *Nature*, 518(7540), pp. 529-533.