5.5.2 Deep Learning RL

Tags

Deep Learning with Reinforcement Learning

Pembelajaran penguatan mendalam menggabungkan jaringan saraf tiruan dengan kerangka pembelajaran penguatan yang membantu agen perangkat lunak belajar bagaimana mencapai tujuan mereka. Artinya, ini menyatukan perkiraan fungsi dan pengoptimalan target, memetakan status dan tindakan ke imbalan yang dihasilkannya.

Anda mungkin tidak memahami semua istilah itu, tetapi istilah-istilah tersebut akan dijelaskan di bawah ini, dengan bahasa yang lebih mendalam dan lebih sederhana, berdasarkan pengalaman pribadi Anda sebagai individu yang bergerak di dunia.

Sementara jaringan saraf bertanggung jawab atas terobosan AI baru-baru ini dalam masalah seperti visi komputer, terjemahan mesin, dan prediksi deret waktu – mereka juga dapat menggabungkan dengan algoritme pembelajaran penguatan untuk menciptakan sesuatu yang menakjubkan seperti Deepmind'sAlphaGo, algoritme yang mengalahkan juara dunia papan Go permainan. Itu sebabnya Anda harus peduli dengan RL yang dalam.

💡 Pembelajaran penguatan mengacu pada algoritme berorientasi tujuan, yang mempelajari bagaimana mencapai tujuan (sasaran) yang kompleks atau bagaimana memaksimalkan sepanjang dimensi tertentu melalui banyak langkah; misalnya, mereka dapat memaksimalkan poin yang dimenangkan dalam permainan melalui banyak gerakan. Algoritme pembelajaran penguatan dapat dimulai dari papan tulis kosong, dan dalam kondisi yang tepat, mencapai kinerja manusia super. Seperti hewan peliharaan yang diberi insentif oleh omelan dan suguhan, algoritme ini dihukum ketika mereka membuat keputusan yang salah dan diberi penghargaan ketika mereka membuat keputusan yang benar – ini adalah penguatan.

Algoritme penguatan yang menggabungkan jaringan saraf dalam dapat mengalahkan ahli manusia yang memainkan banyak video game Atari, Starcraft II, dan Dota-2. Meskipun itu mungkin terdengar sepele bagi non-gamer, ini adalah peningkatan besar atas pencapaian pembelajaran penguatan sebelumnya, dan keadaan seni berkembang pesat.

Pembelajaran penguatan memecahkan masalah sulit menghubungkan tindakan segera dengan hasil tertunda yang mereka hasilkan. Seperti manusia, algoritma pembelajaran penguatan terkadang harus menunggu untuk melihat buah dari keputusan mereka. Mereka beroperasi dalam lingkungan pengembalian yang tertunda, di mana mungkin sulit untuk memahami tindakan mana yang mengarah ke hasil mana dalam banyak langkah waktu.

Various Deep Learning Agents

Agen pembelajaran mendalam adalah sistem berbasis AI otonom atau semi-otonom yang menggunakan pembelajaran mendalam untuk melakukan dan meningkatkan tugasnya. Sistem (agen) yang menggunakan pembelajaran mendalam termasuk chatbot, mobil self-driving, sistem pakar, program pengenalan wajah, dan robot.

Pembelajaran mendalam menggunakan sistem lapisan di mana input diproses dan kemudian output yang diproses diteruskan sebagai input ke lapisan berikutnya, berfungsi seperti neuron di otak manusia. Melalui sistem pemrosesan input dan output

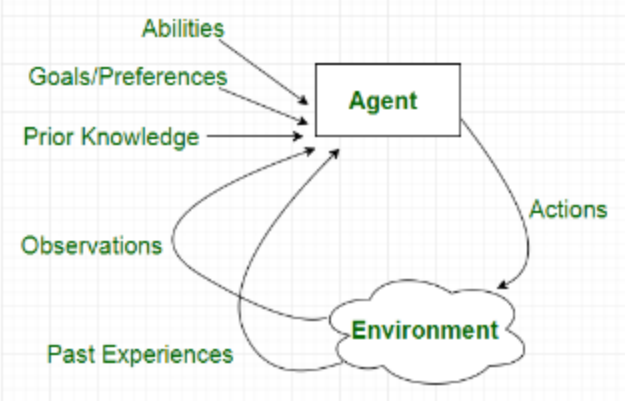
ini, agen pembelajaran mendalam memodelkan pemikiran abstrak dalam data. Sistem pembelajaran mendalam secara fungsional dapat dipecah menjadi dua kategori utama: model pengambilan dan generatif.

Examples of Agent:

Agen perangkat lunak memiliki penekanan tombol, konten file, paket jaringan yang diterima yang bertindak sebagai sensor dan tampilan di layar, file, paket jaringan yang dikirim yang bertindak sebagai aktuator.

Agen manusia memiliki mata, telinga, dan organ lain yang bertindak sebagai sensor, dan tangan, kaki, mulut, dan bagian tubuh lainnya bertindak sebagai aktuator.

Agen robotik memiliki Kamera dan pencari jangkauan inframerah yang bertindak sebagai sensor dan berbagai motor yang bertindak sebagai aktuator.



Types of Agents

Agen dapat dikelompokkan menjadi empat kelas berdasarkan tingkat kecerdasan dan kemampuan yang dirasakan:

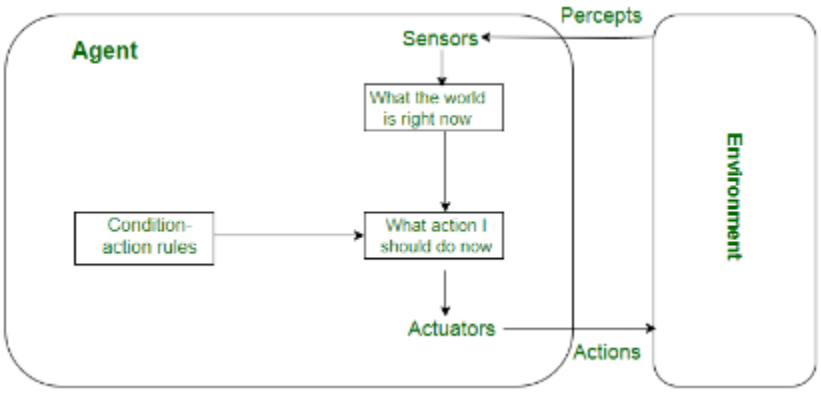
- Simple Reflex Agents
- Model-Based Reflex Agents
- Goal-Based Agents
- Utility-Based Agents
- Learning Agent

Simple Reflex Agents

Agen refleks sederhana mengabaikan sisa sejarah persepsi dan bertindak hanya berdasarkan persepsi saat ini. Percept history adalah sejarah dari semua yang dirasakan oleh agen sampai saat ini. Fungsi agen didasarkan pada aturan kondisi-aksi. Aturan kondisi-tindakan adalah aturan yang memetakan keadaan yaitu, kondisi ke tindakan. Jika kondisinya benar, maka tindakan dilakukan, jika tidak. Fungsi agen ini hanya berhasil ketika lingkungan sepenuhnya dapat diamati. Untuk agen refleks sederhana yang beroperasi di lingkungan yang dapat diamati sebagian, loop tak terbatas sering kali tidak dapat dihindari. Dimungkinkan untuk melarikan diri dari loop tak terbatas jika agen dapat mengacak tindakannya.

Masalah dengan agen refleks sederhana adalah:

- Kecerdasan yang sangat terbatas.
- Tidak ada pengetahuan tentang bagian non-persepsi dari state.
- Biasanya terlalu besar untuk dibuat dan disimpan.
- Jika terjadi perubahan dalam lingkungan, maka kumpulan aturan perlu diperbarui.

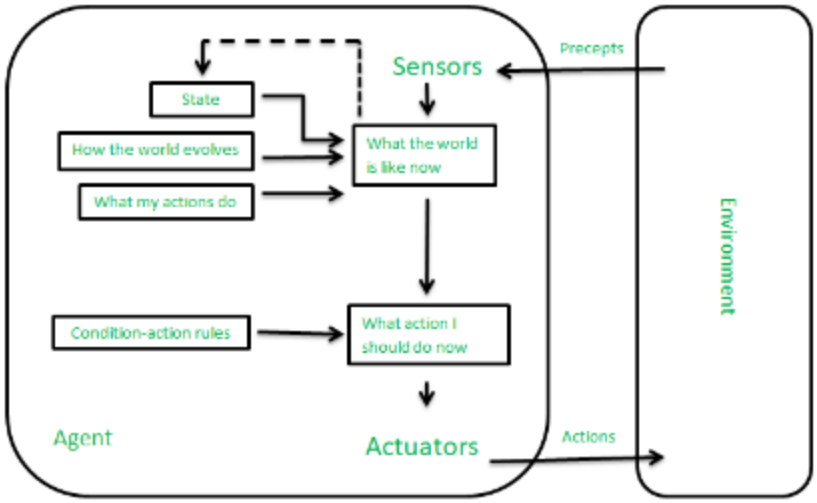


Model-Based Reflex Agents

Ia bekerja dengan mencari aturan yang kondisinya sesuai dengan situasi saat ini. Agen berbasis model dapat menangani sebagian lingkungan yang dapat diamati dengan menggunakan model tentang dunia. Agen harus melacak keadaan internal yang disesuaikan oleh masing-masing persepsi dan itu tergantung pada sejarah persepsi. Keadaan saat ini disimpan di dalam agen yang mempertahankan semacam struktur yang menggambarkan bagian dunia yang tidak dapat dilihat.

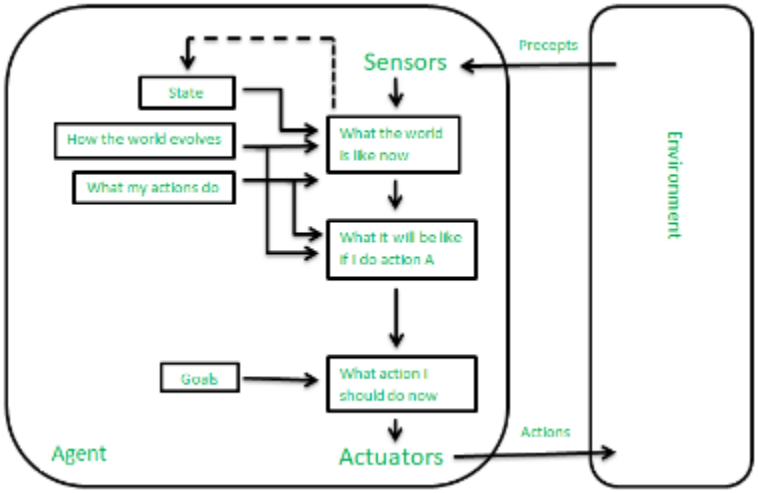
Memperbarui status memerlukan informasi tentang:

- bagaimana dunia berevolusi secara independen dari agen, dan
- bagaimana tindakan agen mempengaruhi dunia.



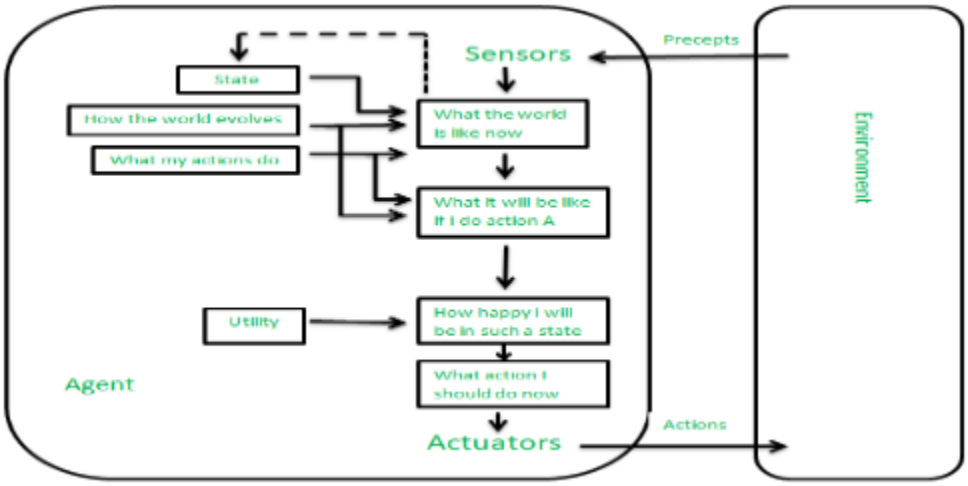
Goal-Based Agents

Agen semacam ini mengambil keputusan berdasarkan seberapa jauh mereka saat ini dari tujuan mereka (deskripsi situasi yang diinginkan). Setiap tindakan mereka dimaksudkan untuk mengurangi jaraknya dari gawang. Hal ini memungkinkan agen cara untuk memilih di antara beberapa kemungkinan, memilih salah satu yang mencapai keadaan tujuan. Pengetahuan yang mendukung keputusannya direpresentasikan secara eksplisit dan dapat dimodifikasi, yang membuat agen ini lebih fleksibel. Mereka biasanya membutuhkan pencarian dan perencanaan. Perilaku agen berbasis tujuan dapat dengan mudah diubah.



Utility-Based Agents

Agen yang dikembangkan memiliki kegunaan akhir sebagai blok bangunan disebut agen berbasis utilitas. Ketika ada beberapa alternatif yang mungkin, maka untuk memutuskan mana yang terbaik, agen berbasis utilitas digunakan. Mereka memilih tindakan berdasarkan preferensi (utilitas) untuk setiap negara bagian. Terkadang mencapai tujuan yang diinginkan tidak cukup. Kita mungkin mencari perjalanan yang lebih cepat, lebih aman, lebih murah untuk mencapai tujuan. Kebahagiaan agen harus dipertimbangkan. Utilitas menggambarkan betapa "bahagia" agen itu. Karena ketidakpastian di dunia, agen utilitas memilih tindakan yang memaksimalkan utilitas yang diharapkan. Fungsi utilitas memetakan keadaan ke bilangan real yang menggambarkan tingkat kebahagiaan yang terkait.

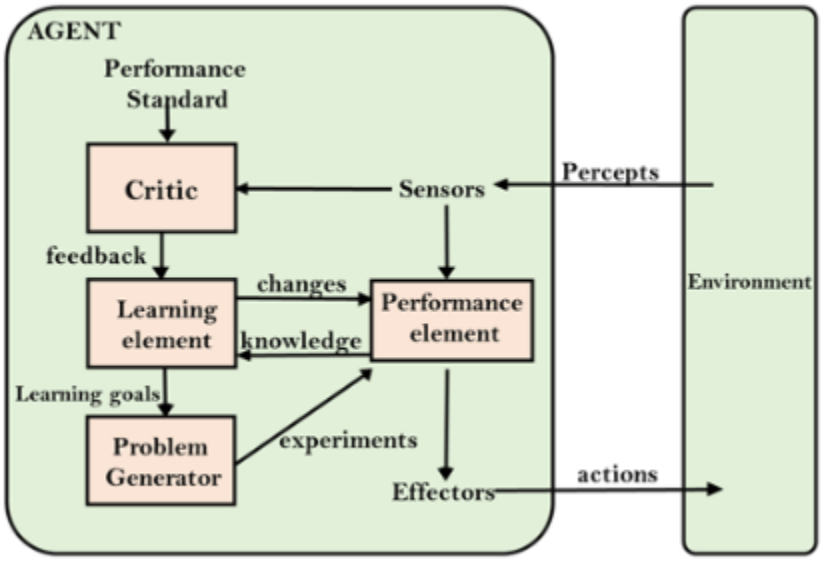


Learning Agent

Agen pembelajaran dalam AI adalah jenis agen yang dapat belajar dari pengalaman masa lalunya atau memiliki kemampuan belajar. Ia mulai bertindak dengan pengetahuan dasar dan kemudian mampu bertindak dan beradaptasi secara otomatis melalui pembelajaran.

Seorang agen pembelajaran terutama memiliki empat komponen konseptual, yaitu:

1. Elemen pembelajaran: Bertanggung jawab untuk melakukan perbaikan dengan belajar dari lingkungan
2. Kritik: Elemen pembelajaran menerima umpan balik dari kritikus yang menggambarkan seberapa baik kinerja agen sehubungan dengan standar kinerja tetap.
3. Elemen kinerja: Bertanggung jawab untuk memilih tindakan eksternal
4. Pembangkit Masalah: Komponen ini bertanggung jawab untuk menyarankan tindakan yang akan mengarah pada pengalaman baru dan informatif.



Q Learning and Deep Neural Networks

Di mana jaringan saraf cocok?

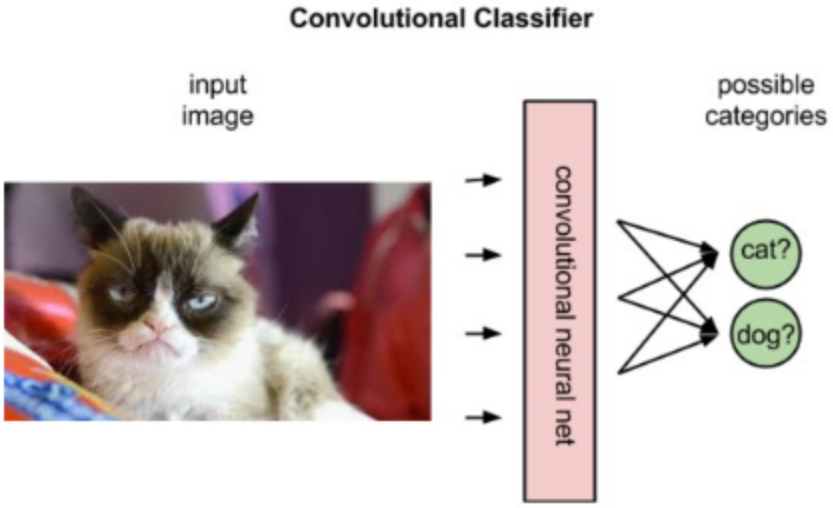
Jaringan saraf adalah pendekatan fungsi, yang sangat berguna dalam pembelajaran penguatan ketika ruang keadaan atau ruang tindakan terlalu besar untuk diketahui sepenuhnya.

Jaringan saraf dapat digunakan untuk memperkirakan fungsi nilai, atau fungsi kebijakan. Artinya, jaringan saraf dapat belajar memetakan status ke nilai, atau pasangan tindakan status ke nilai Q. Daripada menggunakan tabel pencarian untuk menyimpan, mengindeks, dan memperbarui semua status yang mungkin dan nilainya, yang tidak mungkin dilakukan dengan masalah yang sangat besar, kita dapat melatih jaringan saraf pada sampel dari status atau ruang tindakan untuk belajar memprediksi seberapa berharganya relatif terhadap target kami dalam pembelajaran penguatan.

Seperti semua jaringan saraf, mereka menggunakan koefisien untuk memperkirakan fungsi yang berhubungan dengan input ke output, dan pembelajaran mereka terdiri untuk menemukan koefisien yang tepat, atau bobot, dengan secara iteratif menyesuaikan bobot tersebut di sepanjang gradien yang menjanjikan lebih sedikit kesalahan.

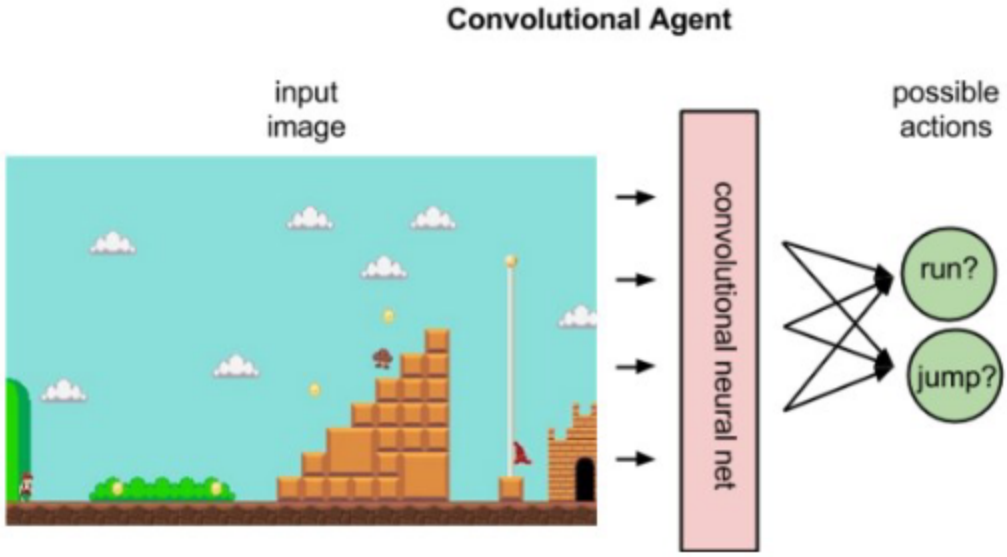
Dalam pembelajaran penguatan, jaringan konvolusi dapat digunakan untuk mengenali status agen ketika inputnya visual; misalnya layar tempat Mario berada, atau medan sebelum drone. Artinya, mereka melakukan tugas khas mereka dalam pengenalan gambar.

Tetapi jaringan konvolusional memperoleh interpretasi yang berbeda dari gambar dalam pembelajaran penguatan daripada dalam pembelajaran yang diawasi. Dalam pembelajaran terawasi, jaringan menerapkan label pada gambar; yaitu, mencocokkan nama dengan piksel.



Bahkan, itu akan memberi peringkat pada label yang paling sesuai dengan gambar dalam hal probabilitasnya. Ditampilkan gambar keledai, mungkin diputuskan bahwa gambar tersebut kemungkinan 80% adalah keledai, 50% kemungkinan kuda, dan 30% kemungkinan anjing.

Dalam pembelajaran penguatan, diberikan gambar yang mewakili keadaan, jaring konvolusi dapat memberi peringkat tindakan yang mungkin dilakukan dalam keadaan itu; misalnya, mungkin memprediksi bahwa berlari ke kanan akan mengembalikan 5 poin, melompat 7, dan berlari ke kiri tidak ada.



Gambar di atas mengilustrasikan apa yang dilakukan agen kebijakan, memetakan keadaan ke tindakan terbaik.

$$\alpha = \pi(S)$$

Kebijakan memetakan keadaan ke tindakan.

Jika Anda ingat, ini berbeda dari Q, yang memetakan pasangan tindakan state bagian ke reward.

Untuk lebih spesifik, Q memetakan pasangan tindakan keadaan ke kombinasi tertinggi dari hadiah langsung dengan semua hadiah di masa depan yang mungkin diperoleh oleh tindakan selanjutnya dalam lintasan. Berikut adalah persamaan untuk Q,:

$$Q(s_t, a_t) \leftarrow (1-\alpha) \cdot \underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \overbrace{\left(\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{estimate of optimal future value}} \right)}^{\text{learned value}}$$

Setelah menetapkan nilai ke imbalan yang diharapkan, fungsi Q hanya memilih pasangan keadaan-tindakan dengan nilai Q tertinggi yang disebut.

Pada awal pembelajaran penguatan, koefisien jaringan saraf dapat diinisialisasi secara stokastik, atau secara acak. Dengan menggunakan umpan balik dari lingkungan, jaringan saraf dapat menggunakan perbedaan antara imbalan yang diharapkan dan imbalan kebenaran dasar untuk menyesuaikan bobotnya dan meningkatkan interpretasinya terhadap pasangan keadaan-tindakan.

Putaran umpan balik ini analog dengan propagasi balik kesalahan dalam pembelajaran terawasi. Namun, pembelajaran yang diawasi dimulai dengan pengetahuan tentang label kebenaran dasar yang coba diprediksi oleh jaringan saraf. Tujuannya adalah untuk membuat model yang memetakan gambar yang berbeda ke nama masing-masing.

💡 Pembelajaran penguatan bergantung pada lingkungan untuk mengirimkannya nomor skalar sebagai respons terhadap setiap tindakan baru. Imbalan yang dikembalikan oleh lingkungan dapat bervariasi, tertunda atau dipengaruhi oleh variabel yang tidak diketahui, yang menyebabkan gangguan pada loop umpan balik.
Ini membawa kita ke ekspresi fungsi Q yang lebih lengkap, yang memperhitungkan tidak hanya imbalan langsung yang dihasilkan oleh suatu tindakan, tetapi juga imbalan tertunda yang mungkin dikembalikan beberapa langkah lebih dalam dalam urutannya.
Seperti halnya manusia, fungsi Q bersifat rekursif. Sama seperti memanggil metode wetwarehuman() berisi di dalamnya metode lainhuman(), di mana kita semua adalah buahnya, memanggil fungsi Q pada pasangan tindakan-status tertentu mengharuskan kita memanggil fungsi Q bersarang untuk memprediksi nilai berikutnya keadaan, yang pada gilirannya tergantung pada fungsi Q keadaan setelah itu, dan seterusnya

Policy Gradient Methods with Neural Networks

Algoritme pembelajaran penguatan cenderung terbagi dalam dua kategori berbeda: pembelajaran berbasis nilai dan pembelajaran berbasis kebijakan. Q Learning, dan implementasi deep neural network-nya, Deep Q Learning, adalah contoh dari yang pertama. Metode gradien kebijakan, seperti yang bisa ditebak dari namanya, adalah contoh yang terakhir.

Sementara metode berbasis nilai telah menunjukkan kinerja yang memenuhi atau melampaui permainan tingkat manusia di berbagai lingkungan, mereka menderita beberapa kelemahan signifikan yang membatasi kelangsungan hidup mereka di berbagai kemungkinan lingkungan yang lebih luas. Secara khusus, dalam pembelajaran Q, agen menggunakan jaringan yang sama untuk memperkirakan nilai dari pasangan tindakan keadaan tertentu, serta untuk memilih tindakan. Hal ini dapat menyebabkan mengejar target yang bergerak, serta adanya bias maksimalisasi, karena argmax dalam proses pemilihan tindakan.

Batasan lebih lanjut dari metode berbasis nilai terletak pada topologi ruang parameter. Fungsi nilai aksi bisa sangat kompleks, dengan banyak minima lokal yang dapat menjebak algoritma penurunan gradien dan menghasilkan permainan di bawah standar.

Metode berbasis kebijakan tidak mengalami kekurangan seperti itu, karena mereka mencoba mendekati kebijakan agen daripada fungsi nilai tindakan. Dalam banyak kasus, ini adalah fungsi yang jauh lebih sederhana untuk didekati dan oleh karena itu kemungkinan terjebak dalam minimum lokal berkurang.

Dalam metode gradien kebijakan tradisional, agen mengambil sampel imbalan dari lingkungan dengan cara Monte Carlo, dan menggunakannya untuk menghitung imbalan masa depan yang didiskon di akhir setiap episode. Imbalan ini digunakan untuk menimbang penurunan gradien sedemikian rupa sehingga agen memberikan probabilitas yang lebih tinggi untuk tindakan yang mengarah pada pengembalian di masa depan yang lebih tinggi, dan kemungkinan yang lebih kecil untuk tindakan yang mengarah pada pengembalian di masa depan yang lebih rendah. Ini dapat dilihat dengan jelas dalam aturan pembaruan:

$$\boldsymbol{\theta}_{t+1} \doteq \boldsymbol{\theta}_t + \alpha G_t \frac{\nabla_{\boldsymbol{\theta}} \pi(A_t|S_t, \boldsymbol{\theta}_t)}{\pi(A_t|S_t, \boldsymbol{\theta}_t)}.$$

Perhatikan adanya gradien pi dibagi pi. Bagi Anda yang akrab dengan kalkulus vektor, ini adalah vektor satuan dalam ruang kebijakan.

Di luar implementasi Monte Carlo, ada serangkaian algoritma gradien kebijakan yang memusingkan. Ada metode kritik aktor, yang memperkenalkan jaringan tambahan untuk mempelajari fungsi nilai aksi, serta algoritme untuk ruang aksi berkelanjutan. Tentu saja, ruang aksi yang berkesinambungan tidak dapat dipecahkan dengan pembelajaran Q, dan dengan demikian metode kritik aktor secara unik cocok untuk lingkungan seperti itu.

Summary

- Pembelajaran penguatan mendalam menggabungkan jaringan saraf tiruan dengan kerangka pembelajaran penguatan yang membantu agen perangkat lunak belajar bagaimana mencapai tujuan mereka.

- Pembelajaran penguatan mengacu pada algoritma berorientasi tujuan, yang mempelajari bagaimana mencapai tujuan (tujuan) yang kompleks atau bagaimana memaksimalkan sepanjang dimensi tertentu melalui banyak langkah.
- Agen pembelajaran mendalam adalah sistem berbasis AI otonom atau semi-otonom yang menggunakan pembelajaran mendalam untuk melakukan dan meningkatkan tugasnya.
- Jaringan saraf dapat digunakan untuk memperkirakan fungsi nilai, atau fungsi kebijakan. Artinya, jaringan saraf dapat belajar memetakan status ke nilai, atau pasangan tindakan status ke nilai Q.
- Dalam pembelajaran Q, agen menggunakan jaringan yang sama untuk memperkirakan nilai pasangan tindakan keadaan tertentu, serta untuk memilih tindakan.