Remedial Reinforcement Learning

Nama : Muhammad Ikhwan Fathulloh

Kelas : Better

Grup : 2

1. Bikin resume materi RL dari pertemuan 1-6, masing2 per materi 1 halaman word, jadi total ada 6 halaman.

Pertemuan 1

Sejarah Reinforcement Learning (RL)

Reinforcement Learning termasuk dalam Artificial Intelegent pada cabang machine learning, dimana untuk pembelajaran mesin, didalamnya terdapat supervised learning, unsupervised learning, dan Reinforcement Learning itu sendiri, bedanya Reinforcement Learning tidak memerlukan dataset disbanding dengan 2 contoh lainnya karena mesin akan senantiasa belajar dari data yang ditemui.

1972 Reinforcement Learning mulai dikembangkan secara masif memiliki struktur agen merupakan actor itu sendiri seperti mesin, action tindakan yang dilakukan oleh mesin, dan reward adalah hadiah yang diberikan atas tindakan mesin yang akan bernilai positif ataupun negative. Ada juga environment berupa lingkungannya, dan state kemampuan yang dimiliki mesin untuk belajar.

Algoritma RL

Menghendaki pertukaran state-action-reward (st , at , rt) antara agent dan environment. Proses ini dinamakan sequential decision-making process, dan (st , at , rt) dinamakan experience

Elements & Environment dari RL

Di luar dari Agent dan Environment yang merupakan sebagai element utama, ada 4 sub elemen sebagai penyusun utama sistem Reinforcement Learning:

Elemen utama:

1. Agent

2. Environment

Sub Elemen utama:

1. Policy

2. Reward signal

3. Value function

4. Model environment (optional)

Aplikasi dari RL

Berikut beberapa contoh dari penerapan Reinforcement Learning untuk Optimasi:

1. Menentukan luasan suatu area dengan pendekatan optimasi monte carlo prediction.

2. Menentukan model suatu fungsi matematika dengan pendekatan optimasi monte carlo prediction

3. Membuat solusi dari permasalahan kasus Traveling Salesman Problem.

4. Membuat suatu sistem Simultaneous Locallization and Mapping (SLAM) untuk mengetahui posisi object terhadap lingkungan dalam koordinat lokal maupun koordinat global

Pertemuan 2

Mengetahui Markov Decision Proses

Markov Decision Process merupakan sebuah tuple (S, A, P, R, γ). Dimana:

•S merupakan state

•A merupakan action

•P merupakan state transition probability function (transition probability)

•R merupakan reward function

•γ merupakan discount factor (γ ϵ [0, 1])

Bisa menyusun MDP dalam merumuskan RL secara formal

• Markov Decision Processes mendeskripsikan secara formal lingkungan untuk RL

• Secara spesifik biasanya dibuat saat environment fully observable

• Hampir semua RL problems dapat diformalisasi menggunakan MDP

• Optimal Control

• Partially Observable problems

• Bandits problem

Keterangan :

• States: himpunan dari states

• $Sstart \in States$: state awal

• Actions(s): actions yang tersedia dari state s

• $P\ s, a, s'$: probabilitas ke s' jika mengambil action a dari state s

• Reward (s, a, s') : idem dengan diatas

• isEnd(s) : apakah state terakhir

• $0 \leq \gamma \leq 1$ faktor diskon

Menentukan Value sebuah Policy

• Mengikuti policy akan bisa menghasilkan jalur yang acak (mengapa?)

• Return (utility) dari sebuah policy adalah Jumlah dari reward selama mengikuti jalur (nilai yang acak)

Mencoba membuat Dynamic Programming tipe Value Iteration

• Dynamic Programming jenis Value Iteration

- Algoritme menggunakan iterasi:
- Inisialisasi $Vopt$ 0 $s \longleftarrow$ 0 untuk semua state

Pertemuan 3

Mengetahui Monte Carlo Prediction

• MC tidak mengambil pengetahuan lengkap dari environment.

• MC belajar dari experience, episode per episode, baik itu experience aktual, maupun simulasi.

• MC belajar dari episode-episode secara utuh dan independent, tidak bootstrapping.

Mengetahui Monte Carlo Estimation & Control

• Karena model tidak tersedia, perlu untuk mengestimasi action juga, selain hanya mengestimasi state-nya.

• Tidak seperti DP, pada Model Free Algorithm, MC, state saja tidak cukup untuk menentukan policy.

• Pada MC, action diperlukan dalam menentukan policy. Policy Evaluation Problem.

• Policy evaluation problem mengestimasi $q\pi\ s, a$, yaitu expected return ketika mulai dari state s, melakukan action a, dan mengikuti policy π.

• First–visit MC mengestimasi value dari pasangan state-action s, a dengan merata-rata returns, saat pertama kali menemui (visit) state s dan mengambil action a dalam suatu episode.

• Every–visit MC mengestimasi value dari pasangan state-action s, a dengan merata-rata returns, setiap menemui (visit) state s dan mengambil action a dalam suatu episode.

• Implikasinya, tidak semua state-action s, a akan ditemui (visited). Apakah masih bisa mendekati expected return, $q\pi\ s, a$?

• Solusi: maintaining exploration

Mengetahui On-Policy dan Off-Policy Monte Carlo

• On–policy MC adalah algoritme Monte Carlo yang melakukan evaluasi atau improvisasi dari policy yang digunakan untuk membuat keputusan-keputusan.

• Pembahasan MC sebelumnya adalah on-policy MC.

• Lalu disini akan dibahas on-policy MC dengan policy dengan nama $\varepsilon - greedy$.

• Dalam $\varepsilon - greedy$ policy, hampir semua action yang dipilih, mempunyai action value estimasi yang maksimal, tetapi dengan probabilitas ε dalam memilih action, tidak dengan random/acak.

Mengetahui Dasar Pemrograman Monte Carlo

1. Monte Carlo bekerja dengan pengalaman dari sample, dan menggunakannya untuk belajar secara langsung tanpa model.

2. Pada first–visit MC, value function dihitung saat pertama kali mengunjungi state s pertama dalam setiap episode.

3. Pada every–visit MC, value function dihitung setiap mengunjungi state s dalam setiap episode.

4. Pada on-policy MC, agent berkomitmen untuk selalu eksplorasi dan mencoba mencari policy terbaik.

5. Pada off-policy MC, dilakukan evaluasi atau improvisasi dari policy yang berbeda dalam meng-generate data (behavior policy), sedangkan target policy-nya sama.

Pertemuan 4

Mengetahui Q Learning

Q-Learning merupakan pengembangan RL yang menggunakan Q-values (disebut juga action-values) untuk meningkatkan kemampuan agent belajar agent berulang-ulang.

Konsep dasar Q-Learning:

• Terinspirasi dari value iteration

• Sample an action

• Observe the reward and the next state

• Take the action with the highest Q (Max Q)

Action dari setiap step dapat dihitung untuk menemukan action terbaik (best action). Untuk keperluan ini digunakan Q-Table.

Memahami bagaimana Q Learning bekerja

1. Tentukan current state = initial state.

2. Dari current state, cari dengan nilai Q terbesar.

3. Tentukan current state = next state.

4. Ulang Langkah (2) dan (3) hingga current state = goal state.

Policy (π) : Adalah sebuah fungsi yg memetakan setiap state ke action. Sehingga dengan mengikuti policy, agent bisa memilih action apa yg akan dia ambil pada state tertentu.

Reward (r) : feedback atau umpan balik untuk agent. Reward dapat bernilai positif (berupa hadiah) atau negatif (berupa hukuman) dan juga nol (tidak ada tindakan apapun terhadap agent).

Episodes: Ketika agent berakhir pada terminating state dan tidak bisa mengambil action apapun pada proses learning.

Mengetahui Deep Q Learning

Deep Q Network adalah sebuah NN yang menerima states yg diberikan oleh environment sebagai input, lalu DQN akan menghasilkan output estimasi Q Values pada setiap actions yang dapat diambil pada state tersebut. Tujuan dari NN ini adalah untuk menghasilkan aproksimasi Q Function yg optimal.

Loss pada NN ini adalah dengan membandingkan antara Q-Values dari output dengan target Q-Values yang didapatkan dari persamaan diatas.

Memahami bagaimana Deep Q Learning Bekerja

Goal dari NN ini adalah untuk minimize loss, lalu setelah menghitung loss bobot pada network akan diupdate menggunakan stochastic gradient descent dan backpropagation seperti neural network pada umumnya.

Dalam proses training DQN kita akan menggunakan sebuah teknik yg dinamakan dengan experience replay. Dengan experience replay, kita menyimpan experience dari agent untuk setiap time step ke dalam sebuah wadah yang bernama replay memory.

Secara teori seluruh experience agent pada setiap time step akan disimpan pada replay memory. Sebenarnya dalam praktiknya, kita akan mendefinisikan besaran experience yang dapat ditampung oleh replay memory sebesar N, dan kita hanya akan menyimpan N terakhir experience dari agent

Pertemuan 5

Mengetahui Dasar dari Robotika

Robotics adalah suatu disiplin ilmu yang mempelajari tentang konsep suatu robot

Robot merupalan mesin yang beroperasikan secara otomatis yang menggantikan usaha manusia, meskipun mungkin tidak menyerupai manusia dalam penampilan atau melakukan fungsi dengan cara yang mirip manusia.

Mengetahui Kegunaan Konsep Machine Learning untuk Robotika

Pada dasarnya dalam algoritma Q-Learning, agent akan memilih action berdasarkan Q-Table. Agent akan memilih action yang mempunyai Q-Value paling besar berdasarkan state saat ini.

Eksplorasi adalah suatu metode pemilihan action dimana agent akan melakukan pemilihan action secara random dengan tujuan ia bisa mengetahui informasi tentang environment secara mendalam. Sedangkan Eksploitasi adalah sebuah metode pemilihan action dengan memilih action yang mempunyai return (dalam hal ini Q-Value) paling besar.

Secara teori seluruh experience agent pada setiap time step akan disimpan pada replay memory. Sebenarnya dalam praktiknya, kita akan mendefinisikan besaran experience yang dapat ditampung oleh replay memory sebesar N, dan kita hanya akan menyimpan N terakhir experience dari agent

Mengimplementasikan Reinforcement Learning pada Robotika

Pertemuan 6

Mengetahui Dasar-Dasar RL (MDP, Bellman Eq.)

• Algoritme RL menghendaki pertukaran state-action-reward (st , at , rt) antara agent dan environment.

• RL belajar (learn) dari interaksi agent dengan environment menggunakan proses eksploitasi/eksplorasi dan sistem reward, untuk memperkuat (reinforce) action positif.

• Eksploitasi dan eksplorasi perlu diatur sedemikian sehingga tujuan dapat tercapai.

• Eksploitasi adalah action RL dalam menggunakan solusi terbaik sebelumnya.

• Eksplorasi adalah action RL dalam melakukan action berbeda untuk mencari solusi yang lebih baik.

• Algoritme RL yang lebih dominan eksploitasi, memungkinkan terlewatkannya kemungkinan peluang solusi lain yang lebih baik

• Algoritme RL yang lebih dominan eksplorasi, memungkinkan agent RL tersebut akan mendapatkan banyak peluang solusi buruk

Mengetahui DP, MC, TD (SARSA), QL dan DQN

Monte Carle

• MC tidak mengambil pengetahuan lengkap dari environment (model free).

• MC belajar dari experience, episode per episode, baik itu experience aktual, maupun simulasi.

• MC belajar dari episode-episode secara utuh dan independent, tidak bootstrapping.

• MC didefinisikan untuk jenis episodic environment.

• Ide utama dari MC: Value didapatkan dari rata-rata returns.

• Dengan semakin banyak returns, nilai rata-ratanya diharapkan konvergen pada expected value.

Q-Learning

1. Tentukan current state = initial state.

2. Dari current state, cari dengan nilai Q terbesar.

3. Tentukan current state = next state.

4. Ulang Langkah (2) dan (3) hingga current state = goal state.

Deep Q-Learning

Deep Q Network adalah sebuah NN yang menerima states yg diberikan oleh environment sebagai input, lalu DQN akan menghasilkan output estimasi Q Values pada setiap actions yang dapat diambil pada state tersebut.

Mengetahui implementasi RL pada robotics

Robot merupalan mesin yang beroperasikan secara otomatis yang menggantikan usaha manusia, meskipun mungkin tidak menyerupai manusia dalam penampilan atau melakukan fungsi dengan cara yang mirip manusia.

Dapat melakukan pemrograman RL dengan python

2. Buatlah satu contoh aplikasi RL dalam bidang apapun dan ceritakan bagaimana aplikasi itu di terapkan (tanpa codingan).

Robot Obstacle Avoiding

Agen : Robot

Lingkungan/env : Lintasan Robot

Aksi : Aksi Lurus, Belok kanan dan Kiri

State : Sensor Jarak Robot

Reward : Saat tidak nabrak akan dapat hadiah positif dan saat nabrak akan dapat hadiah negative

Kerja Robot :

Robot akan bergerak lurus sampai mendapati halangan yang akan di deteksi oleh sensor jarak, lalu akan melakukan aksi berpindah kanan atau kiri jika salah satu aksi tidak ada halangan lagi maka akan bergerak lurus kembali, jika robot tidak menabrak halangan maka akan mendapat reward positif dan jika menabrak akan mendapat hadiah negative.

3. Buat tabel perbandingan dari metode Dynamic Programming, Monte Carlo, SARSA, Q-Learning, dan Deep Q Learning.

Dynamic Programming	Monte Carlo	SARSA	Q-Learning	Deep Q Learning
Salah satu teknik perancangan algoritma yang dikembangkan untuk menyelesaikan permasalahan yang sangat kompleks dengan memecah permasalahan tersebut menjadi banyak sub-permasalahan.	Algoritme komputasi untuk mensimulasikan berbagai perilaku sistem fisika dan matematika. Penggunaan klasik metode ini adalah untuk mengevaluasi integral definit, terutama integral multidimensi dengan syarat dan batasan yang rumit.	Algoritma untuk mempelajari kebijakan proses keputusan Markov, yang digunakan dalam area pembelajaran penguatan pembelajaran mesin.	Algoritma pembelajaran penguatan model-bebas untuk mempelajari nilai suatu tindakan dalam keadaan tertentu. Itu tidak memerlukan model lingkungan, dan dapat menangani masalah dengan transisi stokastik dan penghargaan tanpa memerlukan adaptasi.	Algoritma pembelajaran penguatan model-bebas untuk mempelajari nilai suatu tindakan dalam keadaan tertentu. Itu tidak memerlukan model lingkungan, dan dapat menangani masalah dengan transisi stokastik dan penghargaan tanpa memerlukan adaptasi dengan memaksimalkan nilai reward komulatif

4. Buat dan cari salah satu contoh code dari aplikasi algortima Dynamic Programming, Monte Carlo, SARSA, Q-Learning, dan Deep Q Learning, pilih salah satu yang paling mudah, dan jelaskan cara kerja programnya.

https://github.com/Muhammad-Ikhwan-Fathulloh/Task-Reinforcement-Learning-Orbit-AI-Mastery

Untuk kode saya lampirkan github, terima kasih mohon maaf bila masih banyak kekurangan.