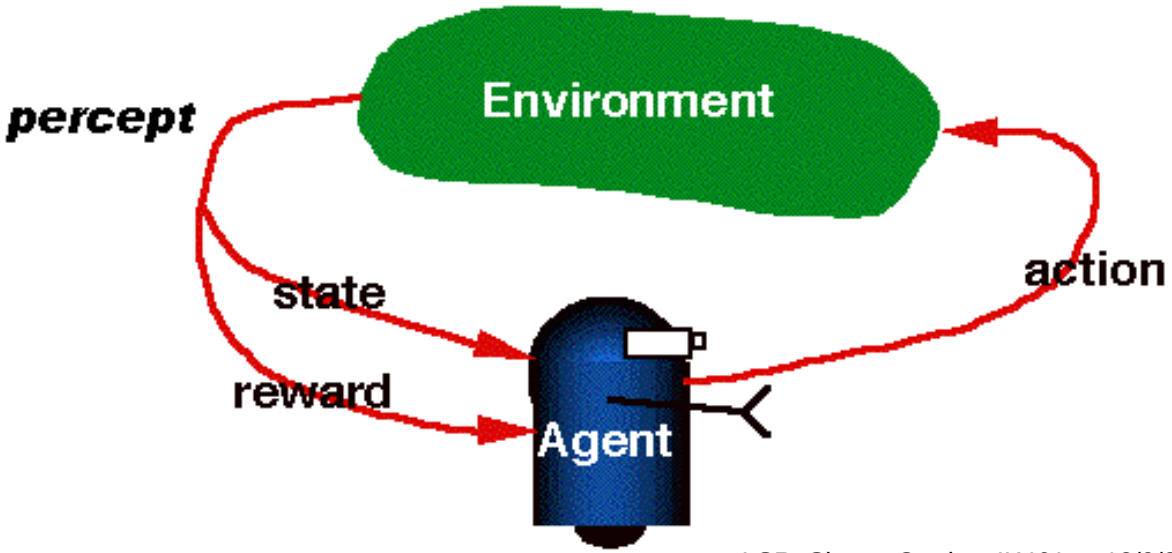
Reinforcement Learning

Pendahuluan RL

- Dari teori dalam ilmu psikologi (Reinforcement Theory).
- Merupakan hasil kompilasi dari pengalaman-pengalaman terdahulu (consequences influence behavior).
- Contoh: seorang pawang mendidik lumba-lumba, monyet.
- A mobile robot decides whether it should enter a new room in search of more trash to collect or start trying to find its way back to its battery recharging station.

Pendahuluan RL

- Dari Reinforcement Theory, terdapat 3 konsekuensi yang berbeda, yaitu:
  - 1. Konsekuensi yang memberikan reward
  - 2. Konsekuensi yang memberikan punishment
  - 3. Konsekuensi yang tidak memberikan apa-apa.

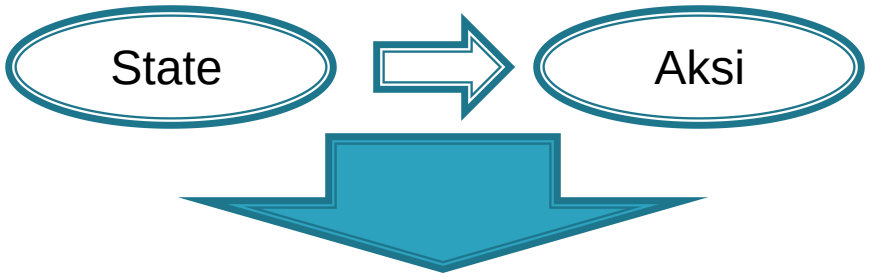


Sejarah

- Diawali munculnya prinsip psikologi klasik oleh Thorndike dalam teori “Law of Effect” tahun 1911.
- Law of effect diaplikasikan dalam computational field, tahun 1954 oleh Minsky.
- Tahun 1960, Donald Michie membuat MENACE (for Matchbox Educable Noughts and Crosses Engine) yg dapat bermain Tic-Tac-Toe.
- Tahun 1963, Andrea membuat STeLLA.
- Tahun 1968, Michie dan Chambers menyempurnakan MENACE yang dinamakan GLEE (Game Learning Expectimaxing Engine).

Konsep Dasar

- RL dibangun dari proses mapping (pemetaan) dari state ke aksi sedemikian hingga diperoleh reward maksimum



Komponen RL

4 Komponen dasar:

1. Policy: Kebijaksanaan (bertugas memetakan state ke aksi yang dipilih).
2. Reward function. (fungsi untuk memaksimalkan reward setelah agent beraksi pada **kurun waktu tertentu**).
3. Value function. (fungsi akumulasi dari **total reward** yang didapat oleh agent).
4. Model of environtment. (aksi yang mungkin dilakukan oleh agent).

Evaluate Feedback

- Evaluate feedback ≈ Fitness Function (GA).
- Evaluate feedback tidak untuk menentukan apakah suatu state terbaik atau tidak tapi untuk memberikan instruksi aksi apa yang akan diambil.
- Jika agent melakukan aksi "a" untuk mencapai goal → diperoleh reward Ri, dgn i adalah waktu tiap kali melakukan aksi sehingga aksi:

$$Q_t(a) = (R_1 + R_2 + \ldots + R_n)/n$$

dimana t adalah waktu yang diperlukan untuk melakukan n aksi. Pada t=0 ➔$Q_0(a) = 0$ karena blm melakukan apa-apa.

Incremental Implementation

- Digunakan untuk menghemat memori (tidak perlu mencatat semua reward dan aksinya!).

$$Q(k+1) = (\sum R(i)) / (k+1) \qquad \text{dimana} \quad i=1..k+1$$

$$= [R(k+1) + \sum R(i)\ ] / (k+1) \qquad \text{dimana } i=1..k$$
$$= [R(k+1) + k\ .Q\,(k) + Q\,(k) - Q(k)] / (k+1)$$
$$= [R(k+1) + (k+1)\ .\ Q(k) - Q(k)] / (k+1)$$
$$= Q(k) + \{[R(k+1) - Q(k)] / (k+1)\}$$

Reward

$$\text{NewEstimate} \leftarrow \text{OldEstimate} + \text{StepSize}\ [\text{Target} - \text{OldEstimate}]$$

aksi

Cat: stepsize[0...1]

Incremental Imp (con't)

$$Q_{k+1} = Q_k + \alpha\Big[r_{k+1} - Q_k\Big],$$

$$\begin{aligned}
Q_k &= Q_{k-1} + \alpha\Big[r_k - Q_{k-1}\Big] \\
&= \alpha r_k + (1-\alpha)Q_{k-1} \\
&= \alpha r_k + (1-\alpha)\alpha r_{k-1} + (1-\alpha)^2 Q_{k-2} \\
&= \alpha r_k + (1-\alpha)\alpha r_{k-1} + (1-\alpha)^2 \alpha r_{k-2} + \\
&\quad \cdots + (1-\alpha)^{k-1}\alpha r_1 + (1-\alpha)^k Q_0 \\
&= (1-\alpha)^k Q_0 + \sum_{i=1}^{k} \alpha(1-\alpha)^{k-i} r_i.
\end{aligned}$$

Cat: α = stepsize, Q_0 = initial action/estimate