Embodied Reinforcement Learning

René Ahn, Loe Feijs, Saskia Bakker, Sibrecht Bouwstra, Jos Verbeek, Bram van der Vlist, Arne Wessels en Rick van de Westelaken. Eindhoven University of Technology, Department of Industrial Design.

Q-learning theory

In both robots, we demonstrate the well-known one-step Q-learning control algorithm. It is a Temporal Difference (TD) method. TD combines Dynamic Programming (DP) and Optimal Control (OC).

We write ϵ for the exploration factor, γ for discounting factor, α for learning factor, π for policy (ϵ-greedy), s for state, a for action, t for (discrete) time, A(s) for action set, Q(s,a) for expected return in state s after action a, under current policy, Q*(s,a) for expected return in state s after action a, under optimal policy, and $\mathcal{E}$ for expectation.

The Designed Intelligence Group investigates embodied interaction for intelligent systems, products and related services. Q-learning could give more autonomy to robots so they perform better in diffcult places and under conditions where full environment modelling and full tele-control are impossible or impractical.

$$\pi(s) : \mathcal{A}(s) \rightarrow [0,1]$$

$$\forall_s \Sigma_{a \in \mathcal{A}(s)} \pi(s,a) = 1$$

$$Q^\pi(s,a) = \mathcal{E}_\pi \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s, a_t = a \right\}$$

$$Q^*(s,a) = \max_\pi Q^\pi(s,a)$$

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha \left[r_{t+1} + \gamma \max_a Q(s_{t+1},a) - Q(s_t,a_t) \right]$$

$$\pi(s,a) = \begin{cases} \frac{\epsilon}{|\mathcal{A}(s)|} & \text{if } a \neq \text{arg } \max_{a'} Q(s,a') \\ 1 - \epsilon + \frac{\epsilon}{|\mathcal{A}(s)|} & \text{if } a = \text{arg } \max_{a'} Q(s,a') \end{cases}$$

calculation of Q-values for policy π

The Crawler

This crawler has wheels to freely move forward and backward. In order to move itself, the crawler can only use its arm, which has two joints under motor control. The Crawler has sensors to measure the position of the joints of the arm and also one distance sensor which "sees" the distance from a wall or another reference object. Inside The Crawler is an NXT control brick, an embedded processor programmed in Java to execute the reinforcement learning algorithm (Q-learning). It is rewarded if it moves forward. It explores its possibilities and learns how it should move to accumulate the maximal rewards. The demo shows The Crawler starting from seemingly random movements, but after a few minutes really finding a kind of rhythm how to move the arm and efficiently move forward. Usually this type of algorithms is demonstrated through screen demos and applets, but here the potential of embodied learning is made visible in a truly embodied model. From a semantic point of view, it is very interesting to observe and interpret the behaviour. A human observer recognises the primitive but charming initial attempts, the gradual progress and the surprising final achievements.

crawler moving itself forward

crawler moving itself forward

crawler moving itself forward