Formalisation of the survival task

Contents

1 The robot

The robot evolves in a flat green environment. Some area are painted on the floor with a blue or red color. These are respective water and food areas. In the following, the time is denoted by t. Interactions between the robot and the world, i.e input acquisition and action execution, is performed every Δt seconds.

1.1 Robot sensors

The robot sees the world through a fixed front camera. The image is a colored image named Cam, pixels coordinates are in $[-.5, .5]^2$ (even if the real image is of course a discrete grid of colored pixels), such as the image width is 1. The robot can move an attention point, i.e. a focus, into the current video image. The position of the focus is denoted by $\mathrm{F}_t = (\mathrm{F.x}_t, \mathrm{F.y}_t) \in [-.5, .5]^2$.

Around the focus of attention, a distorted local view of the visual input is computed as a second colored image, denoted by LGN, where pixels coordinates are in $[-.5, .5]^2$ as well. This image is turned into a Boolean image denoted by Att (the attentional input) thanks to a color filter. The color filter will be denoted by the function $\mathrm{Col}_{h,s,v}$ defined by equation 1, where (h, s, v) is the color to be detected. Thus $\mathrm{Att}_t = \mathrm{Col}_{h,s,v}(\mathrm{LGN}_t)$.

$$\mathrm{Col}_{h,s,v}(c) = (c.\mathrm{h} \in [h - \mu, h + \mu]) \wedge (c.\mathrm{v} > \nu) \quad (1)$$

Last, an artificial physiology is implemented at the level of the robot. It consists of two scalar variables, $\mathrm{H}_t \in [0, 1]$ and $\mathrm{G}_t \in [0, 1]$, which represent respectively hydration and glycemia. Those physiological variables decay over time, except if they are refilled by external inputs from the world, respectively $\mathrm{H}^{\mathrm{in}}_t \in [0, 1]$ and $\mathrm{G}^{\mathrm{in}}_t \in [0, 1]$. The evolution of the physiology can be something like equation 2, where $[x]_0^1 = \max(\min(1, x), 0)$. The formalism here is a discrete update from one time sample to the next, rather than a differential equation.

$$\mathrm{H}_{t+\Delta t} = \left[\tau_{\mathrm{H}}\mathrm{H}_t + \mathrm{H}^{\mathrm{in}}_t\right]_0^1,\ \mathrm{G}_{t+\Delta t} = \left[\tau_{\mathrm{G}}\mathrm{G}_t + \mathrm{G}^{\mathrm{in}}_t\right]_0^1 \quad (2)$$

To sum up, the robot sensor signals are $\{\mathrm{H}_t, \mathrm{G}_t, \mathrm{Cam}_t, \mathrm{Att}_t, \mathrm{F}_t\}$.

1.2 Robot actuators

1.2.1 Implicit actuators

The focus of attention on the input image Cam is driven by a reflex, which is based on Att_t. Indeed, if some `true` pixels lie in Att_t, the focus is moved such as the patch of such pixels gets centered on Att_t. This is driven by a neural field, not detailed here.

1.2.2 Explicit actuators

First actuator is the selection of some color of interest. It consists of producing a signal $\mathrm{C}_t \in [0,1]^3$ corresponding to the HSV parameter given to the color filter. So $\mathrm{Att}_t = \mathrm{Col}_{\mathrm{C.h}_t, \mathrm{C.s}_t, \mathrm{C.v}_t}(\mathrm{LGN}_t)$.

Second actuator is a velocity twist for the robot motion, denoted by

$$\mathcal{T}_t = (\mathcal{T}.\mathrm{lin}_t, \mathcal{T}.\mathrm{ang}_t) \in [0,\lambda] \times [-\alpha, \alpha]$$

Third actuator is the ingestion, which is a boolean signal $\mathrm{Ing}_t \in \{\mathtt{true}, \mathtt{false}\}$. It allows to compute inputs for glycemia and hydration.

$$\left(\mathrm{H}_t^{\mathrm{in}}, \mathrm{G}_t^{\mathrm{in}}\right) = \begin{cases} (0, \delta_\mathrm{G}) & \text{if } \mathrm{Ing}_t = \mathtt{true} \text{ and the robot is in a red area} \\ (\delta_\mathrm{H}, 0) & \text{if } \mathrm{Ing}_t = \mathtt{true} \text{ and the robot is in a blue area} \\ (0,0) & \text{otherwise} \end{cases} \tag{3}$$

To sum up, the robot actuator signals are $\{\mathrm{C}_t, \mathcal{T}_t, \mathrm{Ing}_t\}$

2 Discretizing

For further use in reinforcement learning, we need to discretize the signals. The notation $\widehat{s}$ recalls that s is a signal with discrete values.

2.1 Discrete sensors

Let us define the discrete signals corresponding to the focus position as

$$\widehat{\mathrm{F.x}_t} = \begin{cases} \mathtt{left} & \text{if } \mathrm{F.x}_t < -\beta \\ \mathtt{right} & \text{if } \mathrm{F.x}_t > \beta \\ \mathtt{middle} & \text{otherwise} \end{cases}, \quad \widehat{\mathrm{F.y}_t} = \begin{cases} \mathtt{near} & \text{if } \mathrm{F.x}_t < \beta' \\ \mathtt{far} & \text{otherwise} \end{cases} \tag{4}$$

The physiological variable are roughly discretized as well:

$$\widehat{\mathrm{H}}_t = \begin{cases} \mathtt{comfort} & \text{if } \mathrm{H}_t > \varphi \\ \mathtt{shortage} & \text{otherwise} \end{cases}, \quad \widehat{\mathrm{G}}_t = \begin{cases} \mathtt{comfort} & \text{if } \mathrm{G}_t > \varphi \\ \mathtt{shortage} & \text{otherwise} \end{cases} \tag{5}$$

For the visual input, we only rely on Att. We compute a signal from it, telling whether there is something seen or not. Remember that when something is present in Att, it is implicitly focused on. Thus, let us denote by $\overline{\mathrm{Att}}_t$ the ratio of `true` pixels in Att_t. We can define the discrete information got from the focus point as $\widehat{\mathrm{Att}}_t = \overline{\mathrm{Att}}_t > \xi$.

Let us denote by $o_t \in \mathcal{O}$ the current discretized sensor information available to the robot. The following stands:

$$o_t = \left(\widehat{\mathrm{F.x}_t}, \widehat{\mathrm{F.y}_t}, \widehat{\mathrm{H}}_t, \widehat{\mathrm{G}}_t, \widehat{\mathrm{Att}_t}\right) \tag{6}$$

$$\mathcal{O} = \{\mathtt{left}, \mathtt{middle}, \mathtt{right}\} \times \{\mathtt{near}, \mathtt{far}\} \times \{\mathtt{comfort}, \mathtt{shortage}\}^2 \times \{\mathtt{true}, \mathtt{false}\} \tag{7}$$

$$|\mathcal{O}| = 3 \times 2 \times 2 \times 2 \times 2 = 48 \tag{8}$$

2.2 Discrete actuators

Let us define four discrete twists, such as $\widehat{\mathcal{T}_t} \in \{\texttt{go}, \texttt{stop}, \texttt{turn\_left}, \texttt{turn\_right}\}$, where $\texttt{go} = (\lambda, 0)\,, \texttt{stop} = (0, 0)\,, \texttt{turn\_left} = (0, \alpha)\,, \texttt{turn\_right} = (0, -\alpha)$.

Apart from robot motion, actuators are ingestion Ing_t, which is by definition a discrete signal already, and also the choice of the color for the filter. Let us use only two colors, so that $\widehat{\mathrm{C}}_t \in \{\texttt{blue}, \texttt{red}\}$, where $\texttt{blue} = (h_{\texttt{blue}}, 1, 1)$ and $\texttt{red} = (h_{\texttt{red}}, 1, 1)$.

To sum up, the action $a_t \in \mathcal{A}$ is such as:

$$
\begin{aligned}
a_t &= \left(\widehat{\mathcal{T}_t}, \mathrm{Ing}_t, \widehat{\mathrm{C}}_t\right) && (9)\\
\mathcal{A} &= \{\texttt{go}, \texttt{stop}, \texttt{turn\_left}, \texttt{turn\_right}\} \times \{\texttt{true}, \texttt{false}\} \times \{\texttt{blue}, \texttt{red}\} && (10)\\
|\mathcal{A}| &= 4 \times 2 \times 2 = 16 && (11)
\end{aligned}
$$

3 Toward reinforcement learning

3.1 From time to events

Let us denote by u the time used in RL. RL time is rather called a step in the following. When performing an action at step$u$, the controlled system goes to next step $u + 1$. Let us stress here that t and u are fundamentally different. The problem for applying RL is to identify the succession of steps in the behaviour $(u, u+1, u+2, \cdots)$ while real-life time is sampled as $t, t+\Delta t, t+2\Delta t, \cdots$ that may not match steps!

The same stands for actions. Even if they are discrete, they need to start and end. For example, action `turn_left` is an elementary rotation, i.e. the twist `turn_left` applied during a certain duration. After that duration, the action `turn_left` performed at step u is considered to be performed, and the controller skips to next step $u + 1$.

Such consideration require to set up events, allowing to identify steps u from the temporal signals.

3.2 States and action

States have to be Markovian for applying the RL theory. Is o a Markovian representation that a controller should rely on for playing the role of states ? Let us answer positively for first implementations.

3.3 Reward

Reward is a hard-wired process telling the robot what it is supposed to do, since the behavior computed by RL is the one that maximized the accumulation of rewards[1] along the robot's life. Reward is provided after each transition from step u to $u + 1$, it thus requires an event-based implementation as well. For example, we could consider the reward signal r_t as follows:

$$
r_t = \begin{cases} -1 & \text{if } \widehat{\mathrm{H}}_t = \texttt{comfort} \text{ and } \widehat{\mathrm{G}}_t = \texttt{shortage} \\ -1 & \text{if } \widehat{\mathrm{H}}_t = \texttt{shortage} \text{ and } \widehat{\mathrm{G}}_t = \texttt{comfort} \\ -2 & \text{if } \widehat{\mathrm{H}}_t = \texttt{shortage} \text{ and } \widehat{\mathrm{G}}_t = \texttt{shortage} \\ 0 & \text{otherwise} \end{cases} \tag{12}
$$

[1] A γ-discounted accumulation indeed.