Hierarchical goal abstraction for sensorimotor agency

The model – Draft 0.1 February 14, 2017

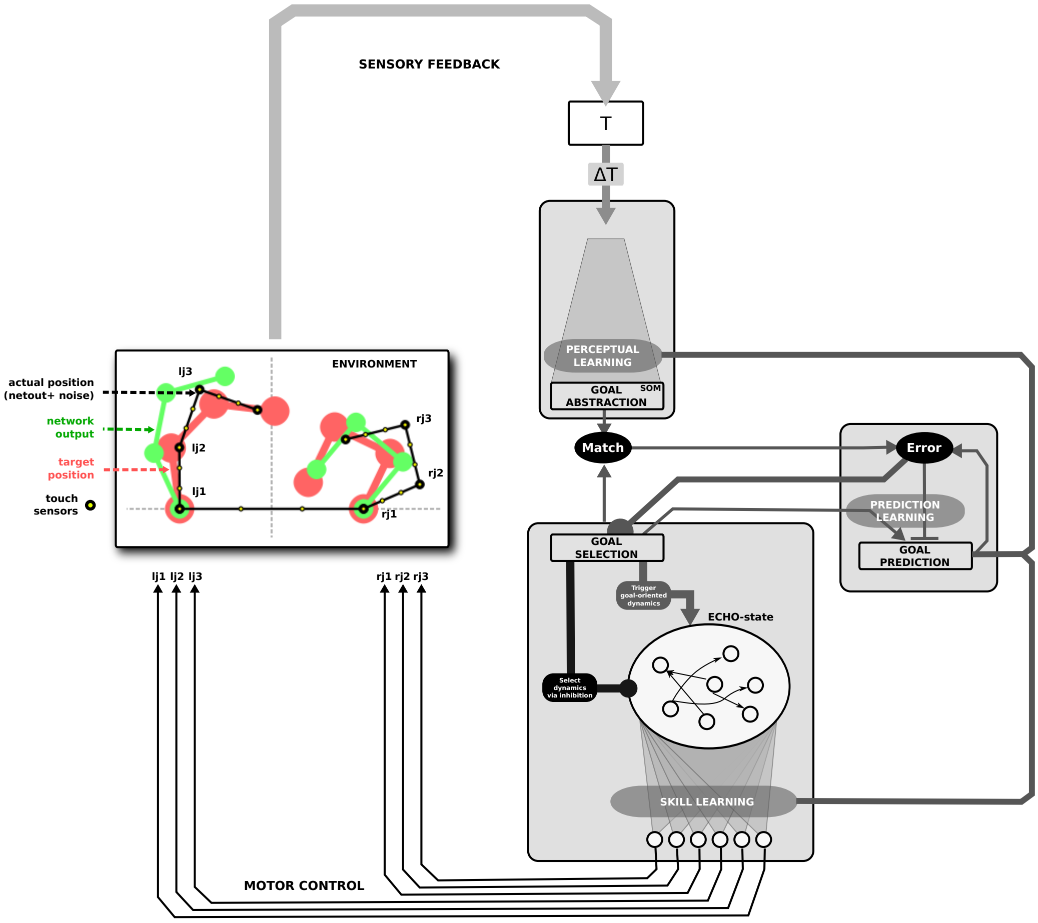

Figure 1: Architecture of the model.

1 The model

1.1 The task

These simulations are a simplified version of the buzzer experiments. We use a simple 2-dimensional agent composed of 2 arms (2 segments with 3 joints each), and a torso connecting them. On the whole body of the agent are positioned N touch sensors, activated when reached by one of the two edges of the arms (the “hands”).

Simulations are divided in two phases, each composed of X trials. A trial ends if a determined amount of time is reached or if a match event (see sec. 2) occurs.

On the first phase, the agent starts exploring its body space with random movements. When a sensor is activated, the system starts learning the proper posture (end-point) needed to determine that activation.

In particular, the learning process is driven by competence-based intrinsic motivations (CB-IMs), so that the system continues to focus on that particular posture as long as it is improving its ability to achieve it. The activations of the touch sensors trigger the formation of internal abstractions together with the movements (end-point postures) necessary to determine those activations.

In the second phase, a token is put on one of the sensors of the agent (to simulate the buzzer touching that part of the body). On the basis of previosly acquired skills, the agent has to learn to reach for the token on its body.

1.2 Expected behaviour

In the first phase, we expect the system to explore its body space and its actions creating a complete sensory-motor abstraction map. We also expect that the more the system is getting close to complete the map, the slower become the learning process: when the agent has learnt to reach all the different part of its body it has formed a sensory-motor map that is no more modified during the first phase of the experiment.

In the second phase, we expect the system to exploit the autonomously acquired knowledge/competence to reach the place where the token/buzzer is positioned.

2 Definition of key terms

goal:

- A state of the world (a particular disposition of the touch sensors, in this version) that the **goal-abstraction layer** can identify and distinguish from others (section 4.1).
- A configuration of the **action-selection layer** that can be linked to a proper action (section 4.2).

```
 Note: the goal-abstraction layer has the same dimensionality of
the goal-selection layer so that they can be compared.
```

match:

- The occurrence of an identical configuration of the goal-abstraction and goal-selection layers.

prediction:

- Computation of the probability that given a configuration of the goal-selection layer the agent will be able to obtain a state in which the goal-abstraction layer has the same configuration (*match* event).

error:

- Difference between the prediction and the real occurrence of a *match* event within a trial.

3 Simulator

- **Environment:** the simulation takes place in a 2-dimensional space without physics.
- **Agent:**
  - The body of the agent is composed of the 2 arms plus a segment joining their origins
  - Each arm has 3 joints (3DoF)
  - N touch sensors all over the body.
- **Agent sensory information**

  In this first version of the model, the system only receives information from the touch sensors. In the final version visual information and proprioception will be added to the sensory input of the system.

  - `input from the touch sensors (T):` a 20x20 pixels retina on which the current activations of the N touch sensors are represented.

4 Controller

The controller takes the sensory information at each time step and produces a motor command, consisting in the requested position of the 6 joint angles (3 for each arm).

More precisely, the actual motor command is a composition between the output of the controller and a noise signal. The more the system learn its skills, the more the actual motor command rely only on the output of the controller (see sec. 4.4 for more details).

4.1 Abstraction

The sensory information provided at each time step is further processed by computing the finite difference with the information given at the previous time step. As a result, the input is composed by a retina containing the *derivative* of the (T) described above.

The result of this processing is sent as input to a self-organizing map (SOM). The output of the SOM is further processed so that all output units have 0 activation except the one whose weights are the closest to the current input, which has 1 activation.

This process generates high-level abstractions of the sensory input that can be compared with the abstractions of the end-point postures. In this way a sensory-motor map can be acquired.

The learning of the SOM is also modulated by the activity of the goal-prediction layer (see sec. **??**)

4.2 Goal selection

The goal-selection layer is a matrix of units with the same numerosity of the abstraction layer. At the beginning of each trial one of the goal-units is selected. The probability for each unit to be selected depends on its value (plus noise), determined by the running averages of the prediction error related to that goal (see section 4.3). Each average is updated at the end of the trials in which the corresponding unit is actually selected: as a result, these running averages convey information about the current amount of error related to each goal-unit.

While the agent is learning, each unit is associated with the production of one particular end-point posture.

4.3 Goal prediction

The **goal-prediction layer** takes the activation of the goal-selection layer as input and return a **prediction of a *match* event** (i.e. a prediction of the achievement of the selected goal). This prediction is compared with the actual occurrence of a *match* event at the end of each trial. The result of this comparison (the *error*) is used both to update the parameters of the goal-prediction layer and to determine the activation of the corresponding unit in the goal-selection layer (see section 4.2).

Moreover, the activity of the goal-prediction layer is used to modulate the learning of the abstraction layer and of the motor controller.

4.4 Action triggering

The **motor output** of the controller is represented by the activation of 6 readout units of an echo-state network (ESN, see the appendix). Each readout unit gives the current amplitude of a joint angle of the 2-arm agent.

The input to the ESN is provided by the goal-selection layer. The activation of the goal-selection layer is connected to the ESN in two ways:

1. It steadily excites distinct subpopulations of the ESN reservoir based on the current selection during a trial. This excitation triggers the proper dynamics of the reservoir and guaranties that the reservoir activity fades to a distinct fixed point depending on the selection.

2. It inhibits distinct subpopulations of the ESN reservoir based on the current selection. This inhibition allows to select different dynamics in the network.

The weights connecting the reservoir to the readout units of the ESN are updated via a reward-based online learning rule, with **the *match* event triggering the reward signal.** Also this learning process is modulated by the activity of the goal-prediction layer (i.e. on the basis of intrisic motivations)

The actual motor output is composed of the activation of the 6 readout units plus some **exploratory noise** noise (6 sinusoidal oscillators whose parameters are set randomly at each trial). The amplitude of the noise depends on the amount of previous *match* events given the same goal selection. As soon as matches become frequent the motor output becomes completely dependent on the activity of the readout units (**exploitation** with no noise).

5 Appendix: the Echo-State Network

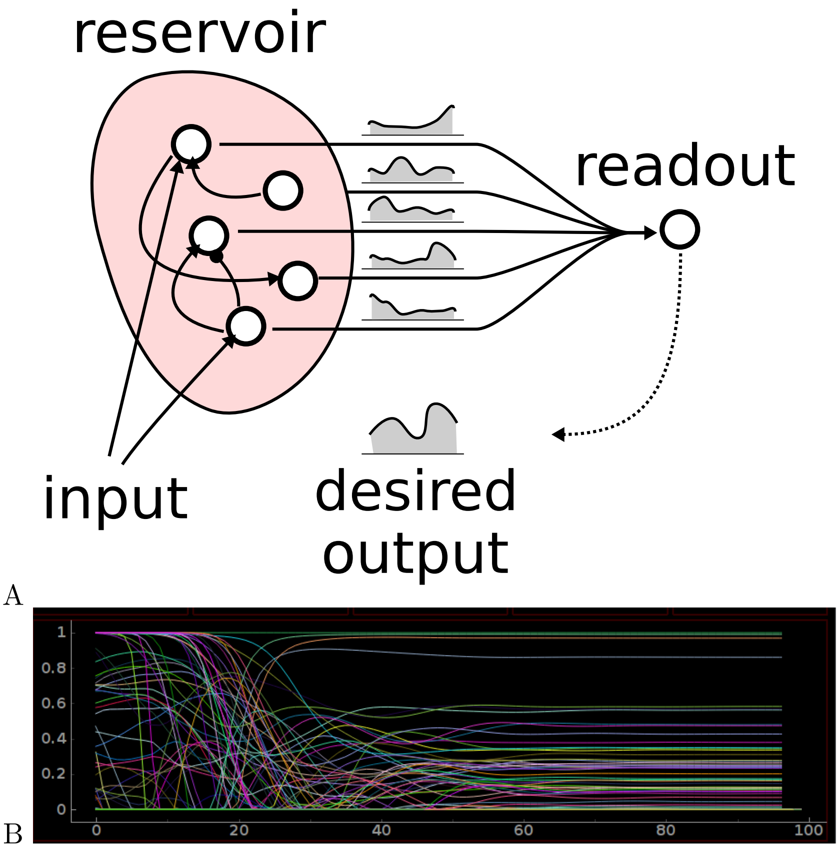

Figure 2: The echo-state network.

Figure 2 shows a general schema of the functioning of an ESN. inputs reach sparsely the internal units of the reservoir.

Internal units are connected with each other via sparse random connections. Learning consists in the update of the external weights connecting the reservoir to one or more readout units. After learning, the readout units reproduce a learnt trajectory in relation to a specific input.

Figure 2B shows the typical activation of the reservoir in the model. During the second half of the trial the ESN reaches a fixed point that is peculiar of for the current input (a selected goal).