Journal Title
XX(X):1–15
©The Author(s) 2016
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

Emotion as an emergent phenomenon of the neuro-computational energy regulation mechanism of a cognitive agent in a decision-making task.

Murat Kirtay[1], Lorenzo Vannucci[1], Ugo Albanese[1], Cecilia Laschi[1], Erhan Oztop[2] and Egidio Falotico [1]

Abstract

Biological agents need to complete perception-action cycles to perform various cognitive and biological tasks such as maximizing their well-being and their chances of genetic continuation. However, the processes performed in these cycles come at a cost. Such cost forces the agent to evaluate a trade-off between the optimality of the decision making and the time and computational effort required to make it. Several cognitive mechanisms that play critical roles in managing this trade-off have been identified. Among these mechanisms, there are: adaptation, learning, memory, attention, and planning. One of the often overlooked outcomes of these cognitive mechanisms, in spite of the critical effect that it may have on the perception-action cycle of organisms, is 'emotion.' In this study, we hold that emotion can be considered as an emergent phenomenon of a plausible neuro-computational energy regulation mechanism, which generates an internal reward signal to minimize the neural energy consumption of a sequence of actions (decisions), where each action triggers a visual memory recall process. To realize an optimal action selection over a sequence of actions in a visual recalling task, we adopted a model-free reinforcement learning framework, in which the reward signal – i.e., the cost– was based on the iterations steps of the convergence state of an associative memory network. The proposed mechanism has been implemented in simulation and on a robotic platform: the iCub humanoid robot. The results show that the computational energy regulation mechanism enables the agent to modulate its behavior to minimize the required neuro-computational energy in performing the visual recalling task.

Keywords

Emotion, Energy Regulation, Emergent Behavior, Visual Recalling, Decision Making

Introduction

Environmental changes (e.g., food depletion and worsening climatic conditions) continuously force biological agents to evaluate a trade-off between the optimality of the decision making, the time and computational load needed for making such decision. Managing this trade-off comes at a cost in environments where predators are plenty, and the computational power is scarce. One of the outcomes of managing this trade-off, albeit not fully understood, is emotion. It plays potent roles in various cognitive functions of biological agents, like attention, memory recall, decision making, and reward extraction (Arbib and Fellous 2004; Murray 2007).

In this study, we build upon the idea that the emergence of emotion can be explained by the neuro-computational energy regulation need of an organism. To be concrete, we propose the following hypothesis: The computational short-cut mechanisms on cognitive processes to facilitate energy economy give rise to what we define as Emotions (Kirtay and Oztop 2013). Here, we use the term emotion to indicate high-level emotions (e.g., boredom) which play a role in low-cost computations rather than basic (reflex-like) emotions such as disgust, fear, and surprise. By leveraging this idea, we aim to show that the consumption of neuro-computational energy (i.e., the neural cost of cognitive processing) for visual recalling can be employed to generate an internal reward signal for action selection. As such, the selection of an optimal action can be carried out by minimizing the neural energy consumption over a sequence of decisions (actions) (Kirtay et al. 2016).

To test the proposed system and its interactions with a set of cognitive components (i.e., as information processing modules), we adopted a model-free reinforcement learning (RL) framework to guide the behavior of a simulated agent and a humanoid robot: a simple cognitive architecture involving the functions of visual perception, memory recall, action learning, and decision making. In this architecture, the actions performed by the agent to explore the environment are, for the robotic agent, movements of the neck and the eyes to direct the gaze while for a simulated agent they consist in the visiting of state-action pairs.

[1]The BioRobotics Institute, Scuola Superiore Sant'Anna, 56025, Pontedera (PISA), Italy.
[2]Department of Computer Science, Ozyegin University, 34794, Istanbul, Turkey.

Corresponding author:
Murat Kirtay, The BioRobotics Institute, Scuola Superiore Sant'Anna, 56025, Pontedera (PISA), Italy.
Email: murat.kirtay@santannapisa.it

As a cognitive task, we have considered visual recalling, for three reasons. First, visual recalling is a task that can be performed both in a simulation environment and on a robot platform; this enables us to implement the proposed methods in a virtual environment and on an actual hardware setup. In this study, we employed an auto-associative network to form visual memories by using visual patterns. However, other sensory modalities with different types of associative networks can also be integrated to process the stimulus in a multimodal way. Second, in cognitive science literature, the roles of emotion are functionally linked in a number of cognitive visual tasks mechanisms such as visual attention, (visual) stimulus evaluation for action selection and facilitating the storage and recalling memories (Arbib and Fellous 2004; Pessoa 2008; Salzman and Fusi 2010). Lastly, we hold that the implementation of this task on a physically embodied agent will enable us to evaluate the conducted experiment from the onlookers perspective. This way, we can argue that the behavior displayed by the robot can be perceived as an affective state of the agent.

In a visual recalling setting, the goal of the agent is to find a sequence of visual percepts (i.e., states) that minimize the "neural cost" of performing visual recalling as the cognitive task undertaken by an auto-associative neural network. The network dynamics allows the definition of a "neural cost" for the recall process based on the prior experience of the agent and the current visual stimulus. This neural-cost is used by the agent to learn the associations between energy and stimulus and thereby guide the agent's behavior. An onlooker may interpret those visual patterns, whose recall consumes less amount of energy –thus that are more often preferred by the agent– as memories that the agent itself has the more emotional affinity with. This emergent affinity (the actions of the agent) even though merely aimed at reducing the neural recall cost, may be perceived as complex behavior regarding action dynamics that depend on the visual memories of the agent and the current visual input. As such, the neural cost of the computation and its use as an internal reward, constitute the emergent emotion in our proposed system.

The proposed internal reward method might also be associated with the intrinsic motivation (or drive) of the agent: to explore this connection, we analyzed the differences between the concepts of emotion and intrinsic motivation, reviewing the relevant studies. Our agent, in each iteration, will process the visual stimulus to extract the stimulus-specific reward based on the recalling cost. In doing so, it will avoid processing the computationally expensive visual stimuli. Moreover, an intrinsically motivated agent will concentrate on the exploration of the decreasing numeric value of the reward for the exploited states. In our experiment, we used the term emotion for an emergent phenomenon that prevents an agent from searching for the best possible, yet computationally expensive decision rather than driving the agent to continuously engage itself in an activity (e.g., searching salient stimulus in the environment). More importantly, the intrinsically motivated agent will perform the curiosity-driven activities (e.g., play, seeking salient information) to satisfy the predefined emotions such as joy, and surprise (Barto 2013; Ryan and Deci 2000; Oudeyer and Kaplan 2009; Moerland et al. 2018). On the contrary, in our experiment, there is no predefined emotion influencing the cognitive task. Instead, we observe emotion emerging from the behavior displayed by the agent in trying to minimize the required computational energy in making sequential decisions. That is why we interpret these behaviors as emotion-guided, in the next section we provide what is needed to support our claim, drawing from neuroscience studies.

Furthermore, in the related work section, to emphasize the difference between our approach and the state-of-art studies regarding internal reward generation, we have discussed the internal reward generation methods from the intrinsic motivation and affective computing literature.

Overall, the results point out that adopting a modulatory role for neuro-computational cost-based emotion in decision making may pave the way for future designs of cognitive robots that will have the need to optimize for computational time and energy spent, both in physical and in computational. In that, we suggest that our study is the experimental indication that the cognitive robot architectures of the future must involve an emergent function of the emotion that is not only biologically inspired but also computationally justified. Unlike existing studies, we do not adopt a fixed set of basic emotion categories (e.g., fear, anger) or a weighted combination of them, but look for possible evolutionary arguments that might have acted to form neural structures for (high-level) emotional expressions as observable behaviors. We also do not consider any dimensional approach to define the appraisal aspect of the emotion (Scherer 2001). Instead, we hold that emotion might be a behavioral manifestation of the neuro-computational energy regulation of the brain to facilitate fast and cheap neural decision making in the face of computationally expensive problems (e.g., visual search).

On the basis of the results obtained, we have highlighted the contributions of this study as follows. Firstly, we proposed a novel way to extract a reward signal relying on the agents internal (neural) system rather than the one arbitrarily assigned by the designer of the experiment. Importantly, we present that the proposed system with the same implementation procedure leads to similar results in both simulation and real-world environments. Secondly, in simulated and real experiments, we observed that the behavior demonstrated by the robot could be interpreted as emotion-guided and it is non-trivial regarding stimulus-energy-reward associations. Lastly, we emphasize that the hardware realization of the proposed system is also an important aspect of the conducted work. To be concrete, a non-trivial behavior emerges from the robots internal mechanisms (i.e., associative memories and internal reward) while operating in a noisy environment; for example, reflections may substantially influence the processing of visual inputs.

Biological background of the proposed approach

Emotion and its mental processes constitute a vast research area. In this work, we adopt the position that – by accepting that it is not the only possible explanation – some of the emotional mental states of biological organisms may be explained by internal reward mechanisms that regulate computational energy consumption. To support our view,

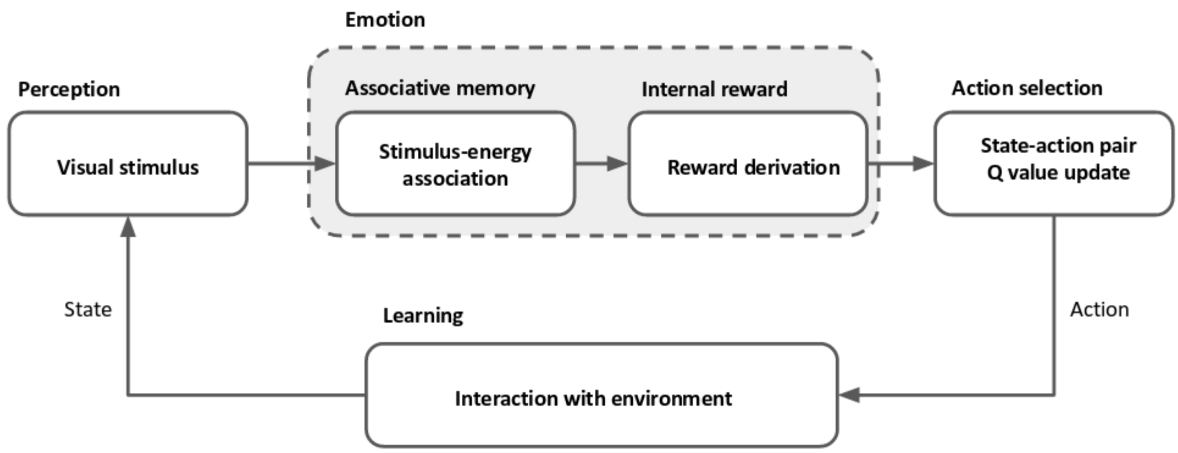


Figure 1. Information flow among components of the agent and their interactions to perform a perception-action cycle in a reinforcement learning framework.

we first introduce the biological literature on decision-making with reference to emotion-energy and energy-reward associations. Then, we present the emulation of a similar association mechanism within the perception-action cycle of an artificial agent.

Emotion-energy and energy-reward associations

Our proposal postulated that certain emotional states are due to the neurocomputational energy regulation mechanism of the neural system of the agent. Further, this regulation mechanism is postulated to be used for forming the internal reward signal for making a series of decisions. To give support for this proposal, we review several neuroscientific studies indicating connections between emotion and internal reward.

The neuroscientific studies on the reward mechanisms are often linked to the neurobiological (e.g., dopamine) and physiological (e.g., learning, emotion (or affect), and motivation) components of the agent (Dayan and Balleine 2002; Berridge and Robinson 2003). Here, we mainly address the coupling between emotion and reward. To be concrete, we have analyzed the decision making and reward circuits in the mammalian brain that has direct and indirect (reciprocal) interactions among different areas, including the Amygdala (Amy), Orbitofrontal Cortex (OFC), Sensory cortex, and basal ganglia, etc (Paton et al. 2006; Haber and Knutson 2010). Some of these brain regions – i.e., the Amy and OFC– also play a potent role in determining the affective (emotional) state of the agent. In particular, in interacting with other regions, they form positive or negative values for a perceived visual stimulus and manage the expectancy of the reward in a decision-making process (Paton et al. 2006; Moren 2002).

Based on this review, we propose that an energy regulation mechanism might be a possible component linking the agents emotional states with the reward values in a reinforcement learning framework. To ground our proposal in neuroscience, we considered energy consumption and its functions from an evolutionary perspective. The bodily energy of living beings (e.g., primates) is a limited resource to be sustained throughout their life spans. The brain consumes a considerable amount of this bodily energy, which has been estimated at 20% (Laughlin et al. 1998). Since the bodily energy is limited and the energy consumption by the brain is non-negligible, the neurons, neuronal codes, and neuronal circuits should have evolved to reduce the (costly) metabolic demands (Laughlin et al. 1998). In that, having a limited energy budget while being exposed to a considerable amount of noisy sensory inputs leads to selective pressure on the sensory systems of the related areas in the brain (Niven and Laughlin 2008). For instance, the nervous system has to manage the trade-off between the need for energy minimization and adaptive behavior generation as a response to changes in an environment (Niven and Laughlin 2008). That is why, from these observations, we infer that the primate brain, as large and complex organ, has been evolved to gain unique energy regulatory functions for maintaining the trade-off between metabolic cost, information processing, and neural-bodily energy economy. For example, a biological agent may not always be able to afford the search for the best option during its lifespan; therefore, the agent needs to adopt a mechanism to regulate neural and bodily resources (Ross and Martin 2006). For some animals, a mechanism to manage this trade-off for decision making is already built into the perceptual system, e.g., in the tectum of the frog (Arbib and Lara 1982). However, for higher functioning species such as humans, the perceptual system must be able to work in high fidelity where other top-down neural mechanisms (e.g., visual attention) modulate its function and allow for decision making based on the environmental context and the state of the agent – that is crucial for a perception-action cycle to sustain the well-being of the organism. In addition to this review, we also noted that further literature findings on the emotion-energy association were provided in our previous studies (Kirtay and Oztop 2013; Kirtay et al. 2016).

Emulation of the perception-action cycle

This study considers emotion as one of the fundamental mechanisms which emerge from managing this trade-off (Kirtay and Oztop 2013). To emulate the proposed mechanism in the reinforcement learning framework as simple cognitive architecture, we hold that the emotion emerges from the interactions between visual perception, associative memory, and the internal reward mechanism.

As illustrated in Figure 1, the agent employs five different cognitive components to carry out a visual recalling task. Here, we refer to components as information processing units in the designed architecture. The agent perceives a visual stimulus, processes it via an associative network, extracts stimulus-energy and energy-reward associations, selects an action to minimize the neural energy consumption and learns the environment to sustain its life cycle. The emulated information flow among these components has been inspired by the neural pathways of the primate brain, including direct and indirect reciprocal interactions among the orbitofrontal cortex, the sensory cortex, and the amygdala (Levine 2009; Murray 2007; Moren 2002).

The implementation of this framework presents how an agent can utilize this mechanism in a simple architecture to extract a reward value from a perceived stimulus and subsequently take a series of actions to find a region where a lesser amount of energy is required to sustain its life cycle. In this way, the agent learns the environment dynamics and how to regulate its energy while performing visual memory recalling as a cognitive task.

Related works

In this section, we first review, from an architectural perspective, the studies on cognitive agents that include emotion as part of their implementations. Then, we evaluate intrinsic motivation studies that consider self (internally) generated rewards in a reinforcement learning framework. For the interested reader, we point out that a more comprehensive review of cognitive architectures, intrinsic motivation, and emotion-related robotics studies can be found in (Kirtay and Oztop 2013; Kirtay et al. 2016; Langley et al. 2009; Barto 2013).

Cognitive Architecture Studies

In study (Gratch 2000), the author proposed an emotional reasoning model to provide planning and reacting modules for an agent. This model was developed based on existing approaches to regulate an elicitation module for altering the agent's behaviors. The elicitation module enables a virtual agent to operate five types of emotion: hope, joy, fear, anger, and distress. The model is employed in a simulator to show that the virtual agent is capable of planning and reasoning by appraising predefined emotions. The authors of study (Marinier and Laird 2008) integrated a reinforcement learning algorithm with appraisal elements, emotions, and moods, to compare a standard reinforcement learning agent with the agent which has appraisal elements. In this work, the task involves a virtual agent in a maze that can evaluate predefined situations such as approaching to the walls or the goal. While taking action in the maze, the software agent was designed to act via well-defined appraisals like suddenness, pleasantness, etc. The work reported in (Lin et al. 2011) presents a design of cognitive-emotional architecture to integrate the generation of emotion and its effects on cognitive processes. The paper follows a unifying approach to selectively merge and combine modified features of several cognitive architectures (Marinier III and Laird 2004; Becker-Asano and Wachsmuth 2010). To be more specific, the appraisal mechanism consists of an arousal and valence node which bidirectionally interacts with the short-term memory component of the architecture. With these interactions, the mood of the agent is derived by averaging arousal and valence values to obtain its effects on the cognitive process. Although this study benefits from a large body of previously developed cognitive architecture literature and cognitive observations, the emotion generation mechanism needs a detailed explanation regarding biological plausibility. Additionally, the proposed system has not been implemented on an agent (either simulated or physically embodied) for quantitative evaluations in real-world. The authors of the study (Franklin et al. 2014) introduced a comprehensive cognitive architecture, LIDA, which has a conceptual appraisal model built on a node linking between emotion and appraisal. These conceptual terms – emotion and feeling– are considered a cognitive motivator for action selection. The cognitive architecture in this model has not been implemented on an agent to understand how the conceptual modeling behaves in a real-world scenario.

As discussed here, the studies on cognitive agents that consider emotions as a component of their designs, lack several features that exist in biological agents.

Firstly, most of these studies focused on a categorical approach to basic emotions and their implications on agent behavior. This approach gives rise to a simplification of the complex nature of emotions and does not answer the functional characteristics of the emotion mechanism in performing a cognitive task (Kirtay et al. 2016). Contrary to this conventional approach, we consider emotions as an emergent phenomenon strictly constrained in the agent's neural system rather than a deterministic input-output process. Integrating categorical emotions into an architecture requires well-defined rules in every state the agent perceives (that can be an external command to direct the agent). Functional integration of emotions considers in what situation emotions should emerge to create computational benefits (i.e., to accelerate learning and take a decision for immediate problems) for an agent.

Secondly, most of the studies mentioned above have limitations regarding the biological plausibility of the emotion mechanisms to describe its interactions with other cognitive components. We propose a simple cognitive architecture that enables an agent to form stimulus-energy associations and employ these associations to extract internal reward values while conducting a cognitive task. To this end, the agent regulates the neural energy consumption and display behaviors that that might be attributed to emotional affinity towards the specific visual stimulus.

Lastly, the experimental realization of most of these architectures on real robot platforms is not available, and the proposed architectures cannot be tested in an experiment where environmental (e.g., noise), and hardware constraints (e.g., camera resolution) are present.

Intrinsic Motivation Studies

Here, we review a number of self-generated reward methods from an intrinsic motivation perspective. Intrinsic motivation

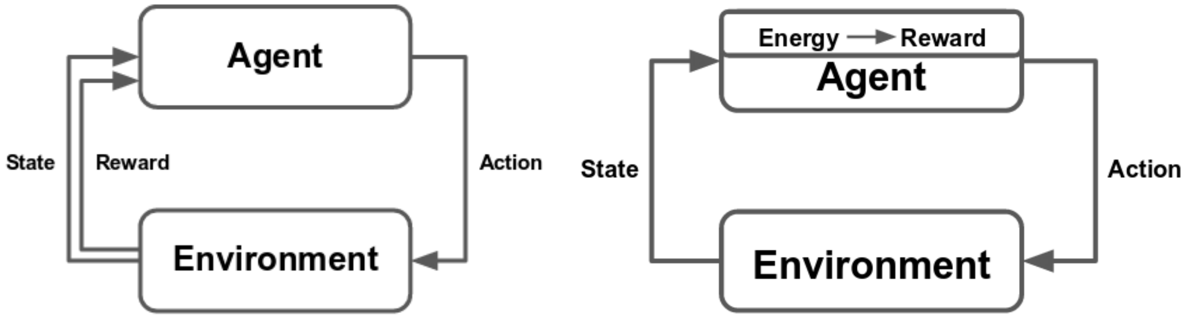


Figure 2. Comparison between standard (left) and our customized (right) reinforcement learning architectures. In standard reinforcement learning the reward is received from the environment, while in our custom architecture the reward is an outcome of the internal mechanisms of the agent.

refers to the behavior of an agent driven by internal rewards rather than external ones from the environment; the motivation to engage in the behavior arises from its being exciting and enjoyable for the agent (Ryan and Deci 2000).

In (Singh et al. 2005), the authors proposed two different reward functions (i.e., extrinsic and intrinsic) to enable the simulated agent to acquire skills in a playroom environment. The intrinsic reward function was employed in response to salient events (e.g., changes in light and sound). In our experiments, the agent only generates an internal reward by deriving the computational cost of recalling; however, this study concerns intrinsic reward in a situation where the agent is "surprised" – that is, perceiving a salient stimulus from the environment. In detail, if the agent is frequently faced with novel events, the value of reward decreases. The work in (Perula-Martinez et al. 2019) proposed a reward function for Q learning based on the well-being of a robot in an interaction scenario with a human. Although this study utilizes various psychological concepts: motivation, drives, and homeostasis, these concepts and their roles in the decision-making framework are predefined (Salichs and Malfaz 2011). For instance, if the robot interacts with a user, the "interaction" drive increases. Similarly, if the robot is acting, the "rest" drive increases; on the contrary, the drive decreases when the robot is waiting. The reward function designed in (Sequeira et al. 2011) is based on predetermined appraisal dimensions of emotion, including novelty, motivation, control, and valence, which are adapted from (Scherer 2001). The authors used the weighted linear combination of these dimensions to construct an intrinsic reward value. In detail, the agent employs an intrinsically motivated reinforcement learning framework in simulated experiments in foraging scenarios.

In this subsection, we provide some representative studies from the agent-generated reward function literature. We point out that the authors in (Moerland et al. 2018) provide a comprehensive review of emotion-related studies in reinforcement learning frameworks.

Our proposed reward generation method differs from the studies we have mentioned, in the following ways. First, our reward function merely depends on the stimulus perceived by the agent which employs an associative (visual) memories to recall the pattern. Second, unlike these studies, in our experiments, there are no predetermined rules to guide the agent to behave in specific cases. To be concrete, there is no prior information about the environment on how the recalling task should be performed and on which decisions should be made. Finally, similar to the studies that we introduced in the cognitive architecture subsection, the robotic experiment of these studies are generally not available to understand the validity of the approach in the real world. In contrast to the results presented in this study, they are mostly employed in game or simulation environments.

We emphasize that, in this study, we follow the same paradigm of our previous studies to assess both virtually and physically embodied agents' behaviors under different experimental conditions (Kirtay and Oztop 2013; Kirtay et al. 2016). The main distinctions of this study can be listed as follows: the agents operate in an environment where more state-action pairs exist and the agents allowed to visit the same state in a row. More importantly, the agents have no a priori information (i.e., the agents do not know where to stop) about the environment where to perform a given task while minimizing the computational energy consumption.

Methods

This section presents the performed methods for the components and their implementation in the reinforcement learning framework. We employed High Order Hopfield Network (HHOP) to form an associative memory for the agent. This associative memory is used for recalling visual stimuli. To derive the consumed (neural) computational energy, we count the number of changed bits (i.e., bi-polarized pixel values) between the converged pattern and the pattern that was received from the environment.

This energy value is used for extracting the reward value that enables the agent to learn the policy that is needed to select the best course of action. After taking a series of actions, the agent learns about the environment and moves from a higher energy state to a lower energy one.

Figure 2 depicts a comparison between a standard reinforcement learning framework and our proposed model. The latter is conceptually derived from the previous, with the main difference being the way we extract reward values, as the product of internal regulatory mechanisms instead of receiving the reward signal received from the environment. We note that the implementation steps described in this section consider only the robotic agent and the same procedures are also valid for the simulated agent.

Extracting reward values for perceptual processing

To process a visual stimulus from the robot camera, we employed a customized version of the Hopfield Network (Hertz et al. 1991). In this implementation, named Higher-Order Hopfield Network, the activation of each unit i (i.e., a neuron) is the sum of the products of the activations of all possible pairs of units (Chaminade et al. 2008):

$$S_i = sgn\left(\sum_{jk} W_{ijk} S_j S_k\right) \tag{1}$$

where $sgn(x)$ is defined as

$$sgn(x) = \begin{cases} -1 \;\; if \; x < 0 \\ \;\; 1 \;\; if \; x \geq 0 \end{cases} \tag{2}$$

Initially, the network was trained with five different patterns which are shown in Fig. 3. These patterns are selected to be either a digit or a letter. We perform the same procedure proposed in the study (Chaminade et al. 2008) to store these patterns for training.

(a) (b) (c) (d) (e)

Figure 3. Visual patterns stored for associative memory.

The training phase starts receiving these patterns from the robot's camera, then downsizing the patterns to 20×20 pixels by applying standard image processing algorithms, and obtaining a single bit binary representation for the pixel values. In order to construct the weight matrix for the network, the binary values obtained have been expressed with a bipolar representation (-1, 1). The weight matrix of training patterns was derived according to Eq. 3

$$W_{ijk} = \sum_{p} \xi_i^p \xi_j^p \xi_k^p \tag{3}$$

where ξ_i^p, ξ_j^p and ξ_k^p are the jth, jth and kth bipolar bits of the pth pattern ξ^p. Since we used five different images as training patterns, this weight extraction step has been performed with $p = 5$. After initializing the weights, 20 different patterns including training patterns, noisy training patterns, and completely new patterns are shuffled to construct the scene depicted in Figure 4. The robot camera is fed with patterns composing the scene as an input pattern. The same vision and image processing algorithms are applied for each input pattern, ξ, to extract their bipolar representations. Then, the asynchronous update rule is performed to lead the network to reach a steady state. Due to properties of HHOP, when it reaches a steady state, the network may have converged to one of the stored patterns, an inverse of one of the stored patterns or a combination of the stored patterns.

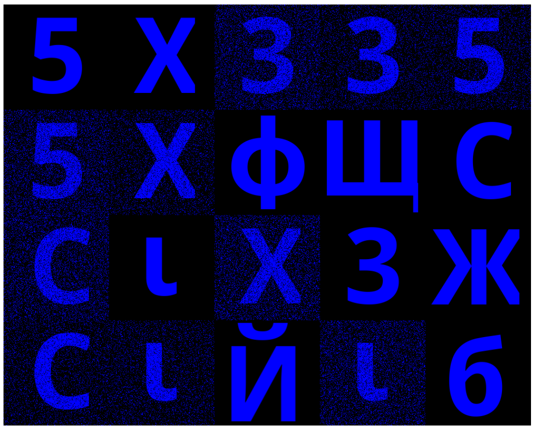


Figure 4. Constructed scene for visual perception.

The obtained pattern for the converged state is denoted by $\bar{\xi}$. The *energy* required to reach the steady state starting from a received input pattern is defined as the total number of flipped bits (change in state of the unit). We compute the energy value via Eq. 4 where N refers to the size of visual pattern.

$$E(\xi) = \sum_{i=1}^{N} \frac{|\xi_i - \bar{\xi}_i|}{2} \tag{4}$$

Note that, due to the random activations of the units in the network, the obtained value $E(\xi)$ is a lower-bound estimate of the actual number of switched activations. Therefore, it can be considered as the minimum amount of computational energy required to converge to the image stored in memory (Kirtay et al. 2016).

Recalling from Figure 2, these energy values are used to extract reward values to implement a temporal difference learning algorithm. In this way, the agent learns to shift its gaze direction towards a state (a discrete region in the scene) in order to maximize the cumulative discounted rewards. In other words, the agent sequentially learns to focus on the regions where a lesser amount of energy is required to process the given visual stimulus.

Reward value and interactions with other components

To learn the environmental dynamics with the proposed mechanism, we customized the reward function of a temporal difference (TD) learning algorithm – namely SARSA – to carry out instructions based on an adopted policy (Sutton and Barto 1998).

To formally define the adapted algorithm, we introduce the Markov Decision Process (MDP) (Ng 2003; Sutton and Barto 1998). The MDP framework can be described as a tuple (S, A, P, R). Where, S indicates a set of states, A is a set of actions that the agent can perform, P is the state transition function, and R is a reward function that evaluates the usefulness of the action, respectively. π is the policy that maps each action to a state, and the customized SARSA algorithm should learn that. It is important to notice that, in this setting, a state ($s \in S$) consists of a discrete region in the scene in Figure 4. The number of states, n_s, is designed to be 20, s_i where $i \in (0, n_s - 1)$. In each iteration, the agent locates itself in one of the available states to perceive a visual pattern via its camera and process it by using associative memory to extract a cost value for visual recalling.

An action, ($a \in A$), is defined as a coordinated movements of eyes and head, which enables the agent to move its gaze from one state to another. In our experiments, the number of action, n_a, is equal to the number of states, a_i where $i \in (0, n_a - 1)$ – that is, the agent can move from a state to all other states, including the current state where it is located. It is also worth noting that, in our implementation, for each state s and action a there is only one resulting new state s'. By iterating over a state-action pair, (s, a), the agent senses the environment through its camera and takes appropriate actions (e.g., head and eyes movements) to explore the environment while performing visual recalling.

In the MDP framework, the agent chooses an action based on a policy, π, which provides the state transition probabilities to map actions to states. However, in our case, we adopted an on-policy TD learning method which updates a policy based on an action that the agent have taken in each iteration. To navigate the visual scene by using coordinated head and eye movements, the agent performs an $\varepsilon - greedy$ strategy. This strategy lets the agent compare all the available actions' values and move its gaze direction towards the most valuable state. Moreover, with a sufficiently small value of ε the agent can exploit random states in the visual scene in order to explore the environment. In this study, ε is chosen to be 0.3. The value of a given state is calculated and updated as in Eq. 5.

$$Q(s, a) \leftarrow Q(s, a) + \mu(R(s, s') + \gamma Q(s', a') - Q(s, a)) \tag{5}$$

$Q(s, a)$ represents the current value of state-action pairs. Similarly, $Q(s', a')$ indicates the value for the action a' in the next state s'. The μ variable is the step size (i.e., learning rate) parameter, γ is an adjustment factor that discounts expected future rewards. The μ values are set to 0.5 and 0.7, with γ fixed to 0.4. Additionally, we present the experiment results for different values of μ and γ in the repository of the study. These values were initially determined performing a grid-search. We used these values to show that similar results can be achieved with different values for these parameters. To this end, we provide the outcomes of the experiments in the results section.

In most MDP frameworks, the reward function, R, is often hand-crafted. Here, we extracted the reward value of a s, s' pair, $R(s, s')$, as a function of the computational energy consumed to process a visual pattern perceived from the scene. To be more concrete, we derive the reward value of a s, s' pair based on Eq. 6. In this equation, ξ^s and $\xi^{s'}$ are the image patterns received in the states s and s' respectively, then the energy values for the execution of recalling operations, annotated by $E(\xi^s)$ and $E(\xi^{s'})$, are obtained and compared. Based on this operation, the reward value is representing whether the agent moves from a higher energy state to lower energy one or vice versa.

$$R(s, s') = \begin{cases} -1 & if\ E(\xi^s) < E(\xi^{s'}) \\ 1 & if\ E(\xi^s) \geq E(\xi^{s'}) \end{cases} \tag{6}$$

We highlight that the agent does not have any prior information about an expected reward in any state-action pair. After a sufficient number of iterations of the Q values, for each state-action pair, the agent is able to perform sequential decision making by following the extracted policy. In other words, starting from any initial state, the agent will eventually find the states where it can perform the visual recalling task with lower energy consumption.

Reproducibilty of the study

In order to reproduce the presented results and provide all the related data – including scripts, experiment report, parameters, figures and images – to other researchers, we used a public repository *. Note that, even if the repository is frequently updated, the current state of the paper with related sources can be found on the branch named *ADAPTIVE2019-submission*.

Results

The implementation of the proposed system was validated by performing experiments in simulation and by employing a robotic agent. In detail, we examined how internally-rewarded actions of the agent can populate the Q-value matrix with state-action pairs and enable the agent to decide and move from a high energy consumption state to a lower one.

Here, we note that the same implementation steps were employed on the simulated agent and the iCub robot platform in order to test whether similar results can be achieved in a real-world environment, thus in the presence of hardware limitations and environmental noise. We emphasized that the agent, either simulated or robotic, performs its actions without prior information about the environment. In particular, there are no predetermined end states to stop the exploration and exploitation in the environment.

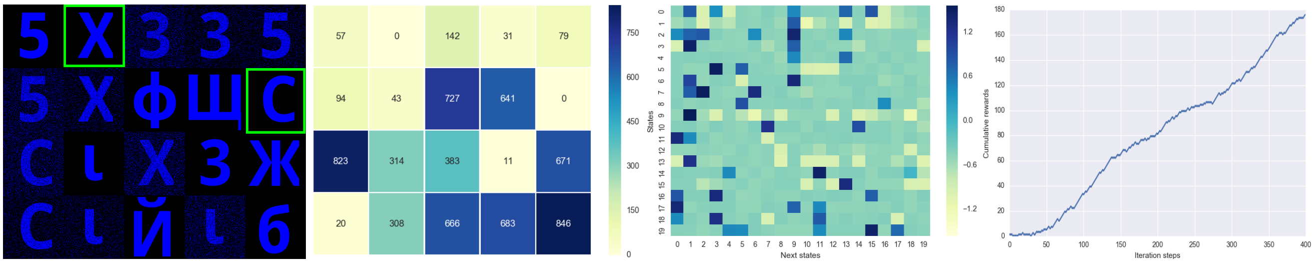
Interpretation of results

In this part, we provide an interpretation of the obtained results. First, we present the discovered states, i.e., the discrete regions in the scene, at the end of each experiment, by using Q matrix values for each state-action pairs. Second, to show that the discovered states are, in fact, the states in which a lesser amount of energy is consumed for the visual recalling task, we provide their average energy values. Then, we counted the number of actions which have led to moving from a high energy state to a lower one. In this case, we consider them as correct actions; otherwise, we deem them as wrong actions. Lastly, we provide the cumulative reward curves to illustrate the behavior of the agent during the experiment.

Simulated agent experiments

In simulation experiments, the possible actions consist of choosing one of the available states where a visual pattern is presented to the agent. The simulation trials were repeated

* www.github.com/muratkirtay/ADAPTIVE2019



(a) Discovered final states (b) Average energy value for each state (c) Q matrix heatmap (d) Cumulative reward

Figure 5. Simulated agent experiment results after 400 iteration steps.

10 times for 200, 300, 400, 500, 600, and 1000 iterations. The same procedure was also applied for different learning rate and discount factor. Although the figures in this section provided for the agent who performs TD learning with $\mu = 0.7$ and $\gamma = 0.4$, we also present results for a different run in a table format. Moreover, we shared more results for the simulated agent in the repository of the paper.

In each iteration, the agent can select an action to move towards any discrete region in the scene including the current. Since there are no predetermined final states to terminate the exploration and the exploitation, the agent should discover a final state or multiple final states by itself.

To test this, at the end of each trial, we first extracted the highest Q values for all the state-action pairs in order to obtain the final policy. If the learned policy leads the agent towards a subset of states (possibly a singleton) these are deemed final. We aim to demonstrate that these states are indeed the ones where less energy is required to process that visual stimulus: moreover, the software agent is capable of regulating its internal processes by constructing stimulus-energy-reward associations. In order to learn the environment and increase the cumulative reward, the agent has to take action to move from a high energy state to a lower one.

Figure 5 shows the obtained results at the end of 400 iteration steps for one of the randomly selected repetitions. The discovered final states are illustrated in Figure 5(a) with a green rectangle. As can be seen in Figure 5(b), the average amount of energy consumed while visiting these two state are the minima, compared with other available states. Therefore, the final states of the policy correspond to the states where less energy is needed to carry out the visual recalling task. We defined the average energy value of a state as the sum of energy values divided by the total number of visits.

The Q-value matrix for all state-action pairs is shown in Figure 5(c) as a heatmap, where the color of each cell indicates how valuable it is to move from a state to the corresponding one. For example, if we assume that the agent is placed in state 7, it will end up oscillating between the two discovered final states where the X and C visual patterns are located (states 1 and 9). To achieve this, the agent will attend the following state sequence $7 \Rightarrow 2 \Rightarrow 9 \Leftrightarrow 1$. This behavior depicted in Figure 6 as a state transition diagram. In this figure, each state is represented by a full circle with state number and the average energy value consumed in that state is reported below the circle. The dashed circles indicate the final states between which the agent oscillates. Since there is no environmental noise in the simulation experiments,

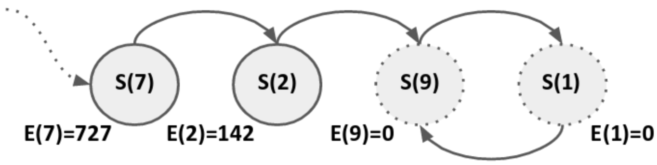


Figure 6. State transition diagram after 400 iteration steps for the simulation agent.

though some patterns had been manually contaminated, the energy values for the discovered states are zero. Based on this state-transition diagram, we conclude that the agent found, indeed, the states requiring minimum energy for visual recalling.

One important outcome of this experiment is illustrated in Figure 5(d), where it is shown that the agent steadily increases the cumulative reward by making decisions to move from high energy states to the lower ones. The cumulative reward is computed as the sum of all the rewards gathered during a trial. This behavior can also be observed for different iteration steps. Figure 7 illustrates the average cumulative reward curves of simulation trials.

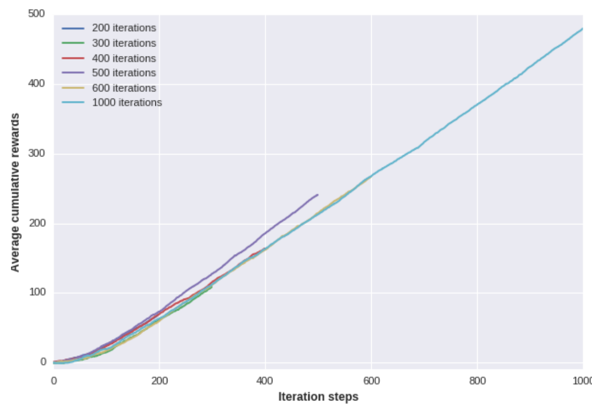


Figure 7. Cumulative reward curves for all iteration steps.

Table 1 shows the results for trials with different numbers of iteration steps. The table consists of eight columns to present iteration steps, the mean cumulative reward (normalized by iteration steps), standard deviation, a classification of state-action pair in the final policy and the discovered final states. Furthermore, the columns were grouped into two subcolumns to display numeric values for the simulated agent with different learning rates and a fixed

Steps	Reward		Std		WA		CA		X-C		X-C-3-5		Others	
	P1	P2	P1	P2	P1	P2	P1	P2	P1	P2	P1	P2	P1	P2
200	0.347	0.323	16.3	16.1	4.5%	20.5%	95.5%	79.5%	7	4	1	5	2	1
300	0.362	0.425	13.2	21.8	4.5%	18.5%	95.5%	81.5%	4	4	4	6	2	–
400	0.409	0.401	15.5	12.7	1.5%	19.5%	98.5%	80.5%	2	4	7	6	1	–
500	0.481	0.447	12.8	23.3	1.5%	17.0%	98.5%	83.0%	7	7	3	3	–	–
600	0.444	0.466	25.7	29.7	1.0%	17.5%	99.0%	82.5%	4	7	6	3	–	–
1000	0.479	0.489	35.3	41.7	–	16.0%	100%	84.0%	5	5	5	5	–	–

Table 1. Experiment results for the simulated agent for all iteration steps with ten runs. P1 columns refer to the experiment results where the μ and γ variables were set to 0.7 and 0.4. Similarly, P2 columns refer to the μ and γ variables were set to 0.5 and 0.4.

discounting factor. To be more specific, P1 and P2 indicate the experiments where the μ and γ parameters were set to 0.7, 0.4 and 0.5, 0.4, respectively.

The cumulative reward value was averaged on ten repetitions of the same trial and normalized by the number of iteration steps. As can be seen in the rows of the first column, the value of the cumulative reward increases with the number of iterations. This indicates that the more the agent performs the visual recalling task, the better the learned policy becomes. We noted that this observation is valid for both P1 and P2 subcolumns. Moreover, to quantify the average behavior of the agent, we provide in the third column, named as Std, of the table, the standard deviation of 10 runs.

The fourth and fifth columns of Table 1 show the number of wrong and correct actions – labeled as WA and CA– taken by the agent for each iteration step. This classification is derived from the Q-value matrix of the agents (see Figure 5(c)). At the end of the experiment, we examine whether the agent moves from a high energy state to the low energy one by taking action corresponding to the highest Q-value, that is the action given by the policy. We, then, compare the average computed energy for the initial state and the arrival state corresponding to the action. Recall that, if the computed energy of the initial state is higher than the energy for the arrival state, we consider this movement correct, otherwise, it is considered wrong. As can be seen in P1 and P2 subcolumns, the percentage of correct actions increases with the number of iterations, though there exist small differences in Table 1, thus indicating an improved policy.

The discovered final states, resulting from the learned policy, are shown in the last three columns of Table 1. From the obtained results, these can be grouped into there categories. In the first category, the agent discovered a single final state corresponding to either the X or C visual patterns (state 1 or 9). In the second category, we can find policies that lead to two or more final states where the training patterns (X, C, 3, and 5) located, these are the two aforementioned states (mostly 1 and 9) and the policies found in this category results in oscillating between at least two patterns or in creating two different attractors. It can also be observed that an increase in the iteration steps decreases the number of low energy states in the set of final ones. In the last category, the final states include other states with training patterns. However, the patterns in this category diminish with increasing the iteration steps for both parameter settings in P1 and P2.

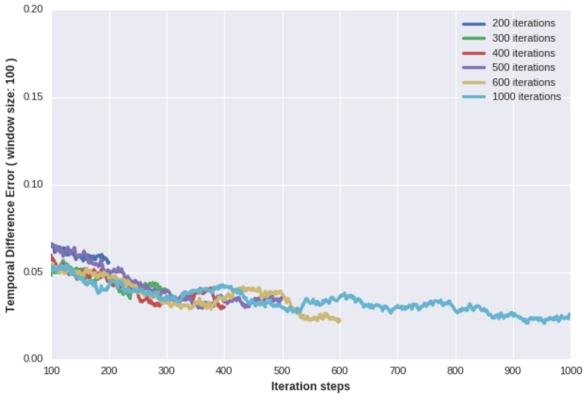


Figure 8. Running average on temporal difference error (td_{error}) over average of 10 repetitions for all iterations.

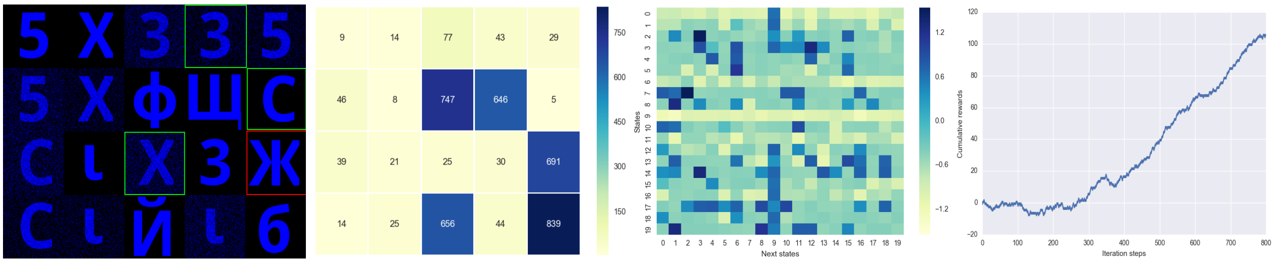
To assess the convergence of the behavior, we looked at the temporal difference error plots (td_{error}) which are derived from Eq. 7. Recalling from Eq. 5, $R(s, s')$ refers to the obtained reward from that state, s, by taking an action which leads the agent to next state, s'. The Q value of state-action pairs is shown by $Q(s, a)$ and the γ value, 0.4, is used as a discount factor for value adjustment.

$$td_{error} = \underbrace{R(s, s') + \gamma Q(s', a')}_{Learned\ value} - \underbrace{Q(s, a)}_{Old\ value} \tag{7}$$

During simulations we recorded td_{error} values and later smoothed them by performing a moving average with a window size of 100 over the average of 10 repetitions. We stopped the simulations after a fixed number of iterations for computational convenience; but, it can be seen from Figure 8 that td_{error} is decreasing towards zero, indicating a progress towards an optimal Q function. The reasons why the td_{error} graphs do not reach zero is twofold (Sutton and Barto 1998). First, the step size parameter was fixed throughout the experiment. Second, the number of iterations were not large enough to eliminate variations brought about by the initial values of the Q matrix and the random nature of the updates of the units in the associative memory. Besides, the external and internal factor that play important roles in the behavior of the agent, will be explained in detail in the discussions section.

Robotic agent experiment

To test the proposed system on a real robotic platform, we employed the iCub humanoid robot as a physically embodied agent. In this way, we aim to show that the



(a) Discovered final states (b) Average energy values (c) Q martix heatmap (d) Cumulative reward curve

Figure 9. Robotic agent experiment results for 800 iteration steps.

proposed approach is also suitable for robotic applications where environmental noise and hardware constraints hinder the processing of incoming visual patterns.

Experiment setup

As shown in Figure 10, in the experiment setup the iCub robot is placed in front of a screen that displays some visual patterns. The iCub robot has two cameras that can capture images with a resolution of 640×480. The robot is controlled using the python bindings for the YARP middleware (Paul et al. 2014). During the experiment, the

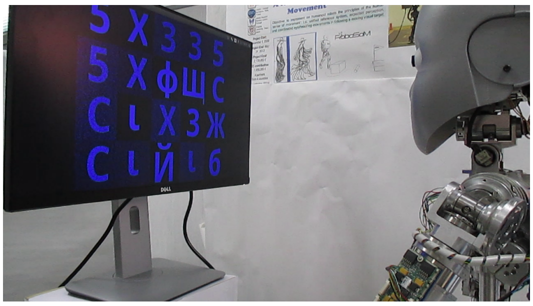


Figure 10. Experiment setup: the iCub robot and a scene for visual perception.

robot directs its gaze toward some specific regions of the scene by employing a coordinated movement of eyes and neck joints (Vannucci et al. 2014, 2015).

As in the simulated experiment, the physical one starts with showing a randomly generated scene to the robot. For comparison purposes, the same scene was used in both cases (cfr. Figure 4 and Figure 11(a)). Each pattern in the scene shall be the target gazed by the robot. We, moreover, define each of such patterns, a state. As shown in Figure 11, at the beginning of every iteration, the current state selected by the robot moving its gaze towards it, is highlighted by a green border. Then, a magnified version of the selected pattern, is shown to the robot for visual processing (Figures 11(b) and 11(c)). The scene is presented again to the agent so that an action can be chosen (Figures 11(d)). The new state is highlighted with a red border, the robot moves its gaze towards, and the corresponding visual stimulus is given(Figures 11(e) and 11(f)).

Results for robotic agent

To evaluate the proposed approach on the iCub robot platform, we conducted an experiment for 800 iteration steps, and saved the data at the 400th step to compare the agent behavior throughout the experiment. This experiment conducted with the same parameters in the first simulation experiment where μ and γ were set to 0.7, and 0.4, respectively.

Figure 9 depicts the results obtained after 800 steps. The discovered final states are shown in Figure 9(a) with green and red rectangles. The corresponding average energy value for each state are provided in Figure 9(b). From this figure, we conclude that the noise in the environment and hardware constraints prevent the associative network to generate the same (or similar) energy values for the simulation experiments. In that, the discovered states obtained are $3 \Rightarrow 12 \Rightarrow 9 \Rightarrow 14 \Rightarrow 3$. We note that this transition structure stems from the the values of the Q matrix in Figure 9(c).

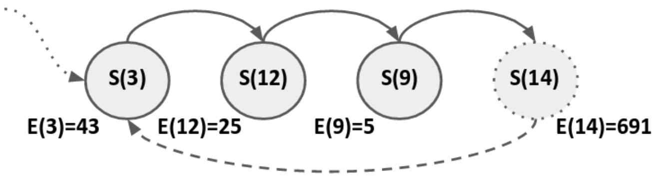
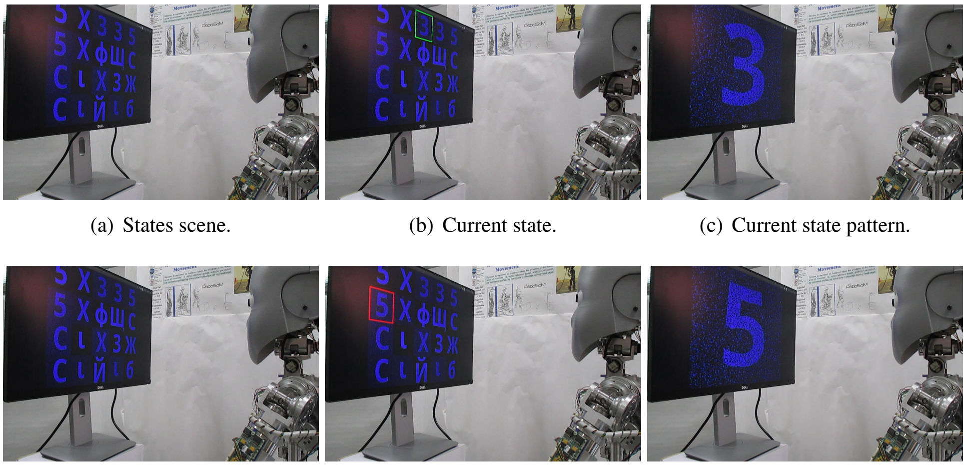


Figure 12. State transition diagram after 800 iteration steps.

To be more descriptive, we provide the state transition diagram of the discovered states in Figure 12. As we explained for Figure 6, the circle indicates the discovered states and the corresponding energy values reported just below the state circles. We underline that regardless of the initial state of the agent, at the end of the iteration the final states will be the ones in Figure 12; moreover, the sequence of gaze direction change follows the path described by the directed arrows in the figure. It can be seen that the transition between two states will lead to lower energy consumption, except for the state 14.

In principle, one might expect that the agent should continuously gaze towards state 9, but such behavior cannot be learned due to the noise in the environment, e.g., light and reflections. In particular, if the agent visits a state twice in a row, the obtained energy value and therefore the extracted reward values will be different, leading to oscillations in $Q(s, s)$ for every state, due to the alternation of positive and negative rewards. As such, it is possible that, from states with very low energy consumption, the policy could lead to a transition state with a higher level of energy consumption. This happens, for instance, in the two cycles of final states of



(a) States scene. (b) Current state. (c) Current state pattern.

(d) States scene. (e) Selected action (next state). (f) Next state pattern.

Figure 11. Robotic agent experiment snapshots. (Experiment media can be found in following link: **www.github.com/muratkirtay/ADAPTIVE2019/iCubExperiment.mp4**)

the policies under examination. This state is marked with a red border in Figure 9(a).

As an important outcome of the robotic experiment, we observe that the number of correct actions increases from 80% to 90% for 400, and 800 iterations, respectively. This indicates that the robotic experiment requires more iteration steps to populate the Q matrix. We also note that the number of transition states – a state with red rectangle– decreases with the number of iterations. To be more specific, $1 \Rightarrow 6 \Rightarrow 9 \Rightarrow 11 \Rightarrow 10 \Rightarrow 1$ were discovered after 400 steps.

The cumulative reward obtained throughout the 800 steps is shown in Figure 9(d). In this figure, by comparing the first and second half of the experiment, we note that the agent needs more iterations to increase the cumulative reward. In particular, Figure 9(d) shows that after a longer initial phase, compared to the simulated trials in Figure 7 where the agent has to sacrifice some reward in order to compensate for the cost of learning, the agent steadily increases the cumulative reward up to the end of the experiment.

Discussions

In this section, we discuss the evaluation of the simulation and the robotic experiments, based on the results presented in Figure 5, Figure 7 and Figure 9. The trends in the plots of cumulative rewards are increasing for both the robotic and the simulated agent concerning the same number of iteration steps, at least after 300 steps. Both the minimum and maximum values of the cumulative reward are higher for the simulated agent than for the robotic one, indicating that the robotic agent needs to lose more reward in the initial iterations before it starts to steadily improve cumulative values. This observation can also be noticed from the duration of the initial phase, where the cumulative reward can get below zero for each agent. The simulated agent learns more quickly to compensate for the cost of the initial learning steps, while the robotic agent needs to take more steps for compensating the learning cost. Since the robot operates in a noisy environment and it receives contaminated patterns for visual processing, the average of the computational energy consumed for discovered states is higher in the robotic experiment than in the simulation.

The reasons why the same implementation, with the same parameters, performs better in simulation trials can be described by external and internal factors of the experimental setup for the robotic agent. The external factors are mainly caused by uncontrollable environmental noise sources and limitations of the hardware, such as camera resolution, lights, and reflections. The internal factors are unique to the implemented algorithms, and they include initial values in the Q matrix for state-action pairs, a high number of random actions due to ϵ-*greedy* strategy, and the number of iterations, to mention a few.

To improve the robot experiment results, for the next phase of this study, the noise factor in the environment can numerically be derived by conducting multiple experiments before starting an actual experiment. This value can be used as a threshold to eliminate transition states. Moreover, fine-tuning some parameters such as initial values in Q-value matrix, decreasing ϵ value after certain iteration (i.e., in the last quarter of the experiment), and decaying the step-size parameter will enable the robot to achieve improved results.

The obtained results from simulation and robotic experiments highlight that the proposed system enables the agent to modulate its behavior to achieve a given visual recalling task with an energy minimization principle. Additionally, the proposed method enabled the agent, in making a series of decisions, to display a non-trivial behavior regarding the consumed energy values.

Conclusions

In this work, we have demonstrated that emotion can be considered the behavioral manifestations of a neuro-computational energy conservation mechanism of an agent that, for its survival, needs to make decisions and thus perform computations. Therefore, we implemented such a mechanism in a simple cognitive architecture and tested its functionality in simulation and real hardware (the iCub humanoid robot). The perception and memory-recall module adopted by the agent to construct stimulus-energy and energy-reward associations for the stimuli, presented in its visual field, was perceived from the environment. Then, the agent used these associations to learn how to make a sequence of actions to minimize the neuro-computational energy required by its perception-action cycle. By leveraging the proposed system, the agent displays a behavior that could be attributed to the agent's emotional affinity towards a preferred stimulus.

The results show that, in a decision-making process, adopting the neuro-computational cost for a modulatory role may be a way to create rich robot behaviors that are not based on blind mimicry of biological emotions, but rather on their emergence from computationally and evolutionary valid principles. To sum up, we presented a novel approach to extract a non-hand engineered reward function in performing non-trivial behavior regarding stimulus-energy-reward association. We noted that the reward mechanism merely relies on the agent's internal processes. The study also shows a realization of the proposed method on a robot platform in a real-world environment.

Our future studies will target a collection of cognitive agent applications with four distinct objectives. Firstly, we will integrate the energy regulation component in a state-of-the-art cognitive architecture (e.g., LIDA) to perform more complex cognitive tasks (e.g., multimodal perception for concept formation).

Secondly, we will integrate more cognitive mechanisms into the architecture such as action generation, reasoning, and learning to incorporate emotion phenomenon to propose a novel (and more complex) cognitive agent architecture. Thirdly, we would like to understand how onlookers perceive the embodied agent in executing a cognitive task driven by the emotion-like mechanism.

Lastly, the reward generation method can be further investigated by employing a different type of network, such as Restricted Boltzmann Machine, to derive energy values for not only visual recalling but also multimodal sensory representation task. Additionally, we envision that the proposed reward function can be considered as a new energy based method which can be integrated into an intrinsically motivated agent in order to compare existing reward functions in the related literature.

Acknowledgements

This research has received funding from the European Unions Horizon 2020 Framework Programme for Research and Innovation under the Specific Grant Agreement No. 785907 (Human Brain Project SGA2) and the Specific Grant Agreement No. 720270 (Human Brain Project SGA1).

We would like to thank the anonymous reviewers for their invaluable and constructive comments that contributed to improving the content of the paper.

References

Arbib MA and Fellous JM (2004) Emotions: from brain to robot. *Trends in cognitive sciences* 8(12): 554–561.

Arbib MA and Lara R (1982) A neural model of the interaction of tectal columns in prey-catching behavior. *Biological Cybernetics* 44(3): 185–196. DOI:10.1007/BF00344274. URL http://dx.doi.org/10.1007/BF00344274.

Barto AG (2013) Intrinsic motivation and reinforcement learning. In: *Intrinsically motivated learning in natural and artificial systems*. Springer, pp. 17–47.

Becker-Asano C and Wachsmuth I (2010) Affective computing with primary and secondary emotions in a virtual human. *Autonomous Agents and Multi-Agent Systems* 20(1): 32–49.

Berridge KC and Robinson TE (2003) Parsing reward. *Trends in neurosciences* 26(9): 507–513.

Chaminade T, Oztop E, Cheng G and Kawato M (2008) From self-observation to imitation: Visuomotor association on a robotic hand. *Brain research bulletin* 75(6): 775–784.

Dayan P and Balleine BW (2002) Reward, motivation, and reinforcement learning. *Neuron* 36(2): 285 – 298. DOI: https://doi.org/10.1016/S0896-6273(02)00963-7. URL http://www.scieancedirect.com/science/article/pii/S0896627302009637.

Franklin S, Madl T, DMello S and Snaider J (2014) Lida: A systems-level architecture for cognition, emotion, and learning. *IEEE Transactions on Autonomous Mental Development* 6(1): 19–41. DOI:10.1109/TAMD.2013.2277589.

Gratch J (2000) Émile: Marshalling passions in training and education. In: *Proceedings of the Fourth International Conference on Autonomous Agents*, AGENTS '00. New York, NY, USA: ACM. ISBN 1-58113-230-1, pp. 325–332. DOI: 10.1145/336595.337516. URL http://doi.acm.org/10.1145/336595.337516.

Haber SN and Knutson B (2010) The reward circuit: linking primate anatomy and human imaging. *Neuropsychopharmacology* 35(1): 4.

Hertz J, Krogh A and Palmer RG (1991) *Introduction to the theory of neural computation*, volume 1. Basic Books.

Kirtay M and Oztop E (2013) Emergent emotion via neural computational energy conservation on a humanoid robot. In: *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*. pp. 450–455. DOI:10.1109/HUMANOIDS.2013.7030013.

Kirtay M, Vannucci L, Falotico E, Oztop E and Laschi C (2016) Sequential decision making based on emergent emotion for a humanoid robot. In: *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. pp. 1101–1106. DOI:10.1109/HUMANOIDS.2016.7803408.

Langley P, Laird JE and Rogers S (2009) Cognitive architectures: Research issues and challenges. *Cognitive Systems Research* 10(2): 141–160.

Laughlin SB, de Ruyter van Steveninck RR and Anderson JC (1998) The metabolic cost of neural information. *Nat Neurosci* 1(1): 36–41. URL http://dx.doi.org/10.1038/236.

Levine DS (2009) Brain pathways for cognitive-emotional decision making in the human animal. *Neural Networks* 22(3): 286–293.

Lin J, Spraragen M, Blythe J and Zyda M (2011) Emocog: Computational integration of emotion and cognitive architecture. In: *Twenty-Fourth International FLAIRS Conference*. p. 0.

Marinier R and Laird JE (2008) Emotion-driven reinforcement learning. *Cognitive science* : 115–120.

Marinier III RP and Laird JE (2004) Toward a comprehensive computational model of emotions and feelings. In: *ICCM*. pp. 172–177.

Moerland TM, Broekens J and Jonker CM (2018) Emotion in reinforcement learning agents and robots: a survey. *Machine Learning* 107(2): 443–480.

Moren J (2002) *Emotion and Learning - A Computational Model of the Amygdala*. PhD Thesis, Lund University.

Murray EA (2007) The amygdala, reward and emotion. *Trends in cognitive sciences* 11(11): 489–497.

Ng AY (2003) *Shaping and policy search in reinforcement learning*. PhD Thesis, University of California, Berkeley.

Niven JE and Laughlin SB (2008) Energy limitation as a selective pressure on the evolution of sensory systems. *Journal of Experimental Biology* 211(11): 1792–1804.

Oudeyer PY and Kaplan F (2009) What is intrinsic motivation? a typology of computational approaches. *Frontiers in neurorobotics* 1: 6.

Paton JJ, Belova MA, Morrison SE and Salzman CD (2006) The primate amygdala represents the positive and negative value of visual stimuli during learning. *Nature* 439(7078): 865.

Paul F, Elena C, Daniele D, Ali P, Giorgio M and Lorenzo N (2014) A middle way for robotics middleware. *Journal of Software Engineering in Robotics* 5(2): 42–49.

Perula-Martinez R, lvaro Castro-Gonzalez, Malfaz M, Alonso-Martn F and Salichs MA (2019) Bioinspired decision-making for a socially interactive robot. *Cognitive Systems Research* 54: 287 – 301. DOI:https://doi.org/10.1016/j.cogsys.2018.10.028.

Pessoa L (2008) On the relationship between emotion and cognition. *Nature reviews neuroscience* 9(2): 148.

Ross DA and Martin A (2006) The origin of mind: Evolution of brain, cognition, and general intelligence. *American Journal of Psychiatry* 163(9): 1652–1653.

Ryan RM and Deci EL (2000) Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary educational psychology* 25(1): 54–67.

Salichs MA and Malfaz M (2011) A new approach to modeling emotions and their use on a decision-making system for artificial agents. *IEEE Transactions on affective computing* 3(1): 56–68.

Salzman CD and Fusi S (2010) Emotion, cognition, and mental state representation in amygdala and prefrontal cortex. *Annual review of neuroscience* 33: 173.

Scherer KR (2001) Appraisal processes in emotion.

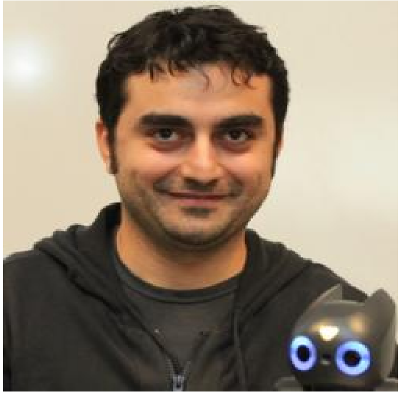
Sequeira P, Melo FS and Paiva A (2011) Emotion-based intrinsic motivation for reinforcement learning agents. In: D'Mello S, Graesser A, Schuller B and Martin JC (eds.) *Affective Computing and Intelligent Interaction*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 326–336.

Singh S, Barto AG and Chentanez N (2005) Intrinsically motivated reinforcement learning. In: *Advances in neural information processing systems*. pp. 1281–1288.

Sutton RS and Barto AG (1998) *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge.

Vannucci L, Cauli N, Falotico E, Bernardino A and Laschi C (2014) Adaptive visual pursuit involving eye-head coordination and prediction of the target motion. In: *IEEE-RAS International Conference on Humanoid Robots*. pp. 541–546.

Vannucci L, Falotico E, Di Lecce N, Dario P and Laschi C (2015) Integrating feedback and predictive control in a bio-inspired model of visual pursuit implemented on a humanoid robot. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 9222: 256–267.



Murat Kirtay is currently a postdoctoral fellow at the BioRobotics Institute. He received his PhD in BioRobotics at the BioRobotics Institute of Scuola Superiore Sant'Anna in 2019. His research interests are in the areas of Humanoid Robotics, Computational and Cognitive Neuroscience, Philosophy of Mind, Machine Learning, and Software Engineering.

Lorenzo Vannucci is a Post-Doctoral fellow at The BioRobotics Institute. He obtained his PhD in BioRobotics from Scuola Superiore Sant'Anna in 2018 (100/100 cum laude) with a thesis entitled "Brain-inspired methods for adaptive and predictive control of humanoid robots" and his M.Sc. in Computer Science at the university of Pisa in 2014 (110/110 cum laude) with a thesis entitled "An adaptive neuro-controller for head-stabilized visual pursuit in a humanoid robot". His current research activity involves the development of biologically inspired adaptive and predictive control mechanisms for humanoid robots.

Ugo Albanese is an assistant researcher at the The BioRobotics Institute. He received his M.Sc. in Computer Science from the University of Pisa in 2015 (110/110) with a thesis entitled "Data parallel pattern in Erlang/OpenCL". He partecipated in several research projects in the robotics field. He is currently involved in the development of the Neurorobotics Platfom (NRP), a simulation platform that aims at bridging the gap between computational neuroscience and robotics. The NRP is developed in the framework of the European Union H2020 FET Flagship "The Human Brain Project".

Cecilia Laschi is Full Professor of Biorobotics at the BioRobotics Institute of the Scuola Superiore Sant'Anna in Pisa, Italy. She graduated in Computer Science at the University of Pisa in 1993 and received the Ph.D. in Robotics from the University of Genoa in 1998. In 2001-2002 she was JSPS visiting researcher at Waseda University in Tokyo. Her research interests are in the field of soft robotics, humanoid robotics, and neurorobotics. She is in the Editorial Boards of several international journals. She serves as reviewer for many journals, including Nature and Science, for the European Commission, including the ERC programme, and for many national research agencies. She is Senior member of the IEEE, of the Engineering in Medicine and Biology Society, and of the Robotics & Automation Society, where she served as elected AdCom member and currently is Co- Chair of the TC on Soft Robotics.

Erhan Oztop received the Ph.D. degree from the University of Southern California, Los Angeles, CA, USA, in 2002. He researched and led several research groups at Advanced Telecommunications Research Institute International, Kyoto, Japan during 2002-2011. Then, he joined Ozyegin University in Istanbul, Turkey as a faculty member, where he is currently serving as the Chair of the Computer Science Department. His research involves computational study of intelligent behavior, human-in-the loop systems, computational neuroscience, machine learning, cognitive and developmental robotics.

Egidio Falotico received the M.S. degree in computer science from the University of Pisa, Pisa, Italy,in 2008 and the Ph.D. degree in biorobotics from Scuola Superiore Sant'Anna (SSSA), Pisa, Italy, in2013, and the Ph.D. degree in cognitive science from the University Pierre et Marie Curie, Paris, France, in March 2013. He is currently an Assistant Professor with The BioRobotics Institute, SSSA. He currently serves as the Deputy Leader and the Publications Manager in the Sub-Project 10 of the Human Brain Project. He is the author or coauthor of more than 40 international peer-reviewed papers and he regularly serves as a reviewer for more than 10 international ISI journals. He has been involved in some EU-funded projects (I-SUPPORT, SWARMs, SMART-E, RoboSoM, RobotCub), focusing on the development of brain inspired algorithms for robot control. His research interests focus on neurorobotics, i.e., the implementation of brain models from neuroscience in robots.