Neural Networks ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at SciVerse ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

2012 Special Issue

Hierarchical curiosity loops and active sensing

Goren Gordon *, Ehud Ahissar

Department of Neurobiology, Weizmann Institute of Science, Rehovot, 76100, Israel

ARTICLE INFO

Keywords:
Reinforcement learning
Intrinsic reward
Internal models
Active sensing
Whisker
Vibrissa
Touch
Object localization

ABSTRACT

A curious agent acts so as to optimize its learning about itself and its environment, without external supervision. We present a model of hierarchical curiosity loops for such an autonomous active learning agent, whereby each loop selects the optimal action that maximizes the agent's learning of sensory-motor correlations. The model is based on rewarding the learner's prediction errors in an actor-critic reinforcement learning (RL) paradigm. Hierarchy is achieved by utilizing previously learned motor-sensory mapping, which enables the learning of other mappings, thus increasing the extent and diversity of knowledge and skills. We demonstrate the relevance of this architecture to active sensing using the well-studied vibrissae (whiskers) system, where rodents acquire sensory information by virtue of repeated whisker movements. We show that hierarchical curiosity loops starting from optimally learning the internal models of whisker motion and then extending to object localization result in free-air whisking and object palpation, respectively.

1. Introduction

A curious agent, animal, human or robot, interacts with its environment in order to learn cause and effect relations. The first stage of this interaction is to learn its own body, how it moves and acts when commands are sent to its actuators. Next comes interaction with the external environment, where sensory-motor correlations are learned with increasing complexity. This hierarchical structure of actively learning the environment is ubiquitous in biological systems.

Furthermore, the first stages of this bottom-up construction is usually formed by *autonomous learning*, i.e. there is no external teacher. Only internally accessible information, such as reafferents from the sensors and efference copy of the motor commands, are used to learn the dynamical relations of interaction with the environment.

Active sensing is one way for acquiring this information, by controlling the sensor and moving it in order to better learn the environment (Ahissar & Arieli, 2001). However, in order to use the sensor effectively, the agent must first learn the internal dynamics of the sensor, composing the first stage in the hierarchy.

The question then arises: what is the optimal way to move in order to expedite the learning process in each level of the hierarchy? Previously (Gordon & Ahissar, 2011), we showed that the paradigms of active sensing and autonomous learning with reinforcement learning can be combined. Here, we present the basic curiosity loop in which the optimal actor, or action policy, is found that maximizes the learning process by rewarding a learner's prediction error in an intrinsic-reward (Oudeyer, Kaplan, & Hafner, 2007) actor-critic architecture. Hierarchical buildup of such basic loops is described, whereby each loop utilizes lower loops' learners to increase the complexity of learned agent–environment correlations. Each loop learns the optimal actions for its specific learners, thus resulting in an increase in policy repertoire.

The hierarchical curiosity loop architecture requires delicate inter-loop switching, both in the so-called developmental stage, in which each loop's optimal actor is learned, and the on-line stage, in which the currently active policy is selected from the repertoire of converged actors. The former switching is determined by the converging properties of the actor-critic architecture, while the latter switching is determined by the learners' decreasing prediction error.

We implement the suggested architecture on the well-studied vibrissae system (Ahissar & Kleinfeld, 2003; Kleinfeld, Berg, & O'Conner, 1999; Knutsen & Ahissar, 2009), whereby rodents use their whiskers to actively sense their environment (Knutsen et al., 2006; O'Connor et al., 2010; Szwed, Bagdasarian, & Ahissar, 2003; Venkatraman & Carmena, 2011). This system is relatively simple, moving in one dimension (whisker's azimuth angle (Diamond, von Heimendahl, Knutsen, Kleinfeld, & Ahissar, 2008)), yet complex enough to exhibit several stereotypical behaviors (Grant, Mitchinson, Fox, & Prescott, 2009; Towal & Hartmann, 2008), such as periodic free-air whisking and object palpation. The first level of the hierarchy is composed of the internal models of the free-air whisker dynamics, e.g. the forward model that learns to predict the angle of the whisker in the next time-step, and the

* Correspondence to: 25 Dubnov St., Rishon LeZion, 75215, Israel. Tel.: +972 525374429; fax: +972 775374424.
E-mail addresses: goren@gorengordon.com (G. Gordon), Ehud.Ahissar@weizmann.ac.il (E. Ahissar).

0893-6080/$ – see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.neunet.2012.02.024

inverse model that learns to predict the required action to bring the whisker to a desired angle. We compare the resulting behavior of each internal model curiosity loop, to that of a combined loop in which both internal models are learned concurrently.

The second level of the vibrissae hierarchy is object localization, in which touching an object results in contact information. The loop attempts to find the whisker movement policy that optimizes contact predictability or the touch forward model, i.e. given the whisker angle and performed action, whether contact will occur. We present both the isolated object localization loop, whereby contact is given by a specialized touch sensor, and the hierarchical architecture, whereby contact is given by deviation from the free-air forward model, also known as angle absorption (Szwed et al., 2003) and control is done via the whisker inverse model.

The entire process is completely autonomous, in the sense that inter-loop switching, both in the developmental stage of converging actors, and the on-line switching in selecting the active policy, is determined autonomously. The resulting behaviors of the vibrissae hierarchical curiosity loops are remarkably similar to the observed stereotypical behaviors of rats. Namely, the internal model is optimally learned by quasi-periodic whisking and object localization is optimally learned by object palpation.

The presented model suggests experimental predictions, mostly in the developmental stage, and offers a unified paradigm for a repertoire of the complex observed behaviors of rodents' whiskers. The paper is organized as follows: we first describe the basic curiosity loop in Section 2 and then analyze the whisker's internal models loop in Section 3 and object localization loop in Section 4. Section 5 describes the hierarchical loops architecture, followed by its implementation on the vibrissae system in Section 6. We end with discussion and conclusions in Section 7.

2. Basic curiosity loop

The basic curiosity loop is characterized by the correlation it is constructed to autonomously learn. Augmenting the learner with the critic-actor components of reinforcement learning results in a closed learning loop, whose goal is to find the optimal policy that maximizes the learning process. Thus, each loop's convergent dynamics result in a specific behavior which is tightly related to the objective learnable correlation. We first explain the concept of internally supervised learning with an emphasis on internal models and then describe the basic architecture of the curiosity loop, followed by a description of the explicit RL model we use, namely, the incremental natural actor critic (iNAC) (Bhatnagar, Sutton, Ghavamzadeh, & Lee, 2007). We end with a novel continuous space continuous action implementation of the actor and critic used in the rest of the paper.

2.1. Internally supervised learning and internal models

Internally supervised learning is defined by a learner that attempts to learn an input (i) output (o) transformation: $L(o|i;\chi)$ where χ are the tuned parameters. It is performed by presentation of correct input–output pairs, called the training set. We use supervised on-line learning algorithms to find the optimal parameters χ^* such that the generalization error is minimized, i.e. $L^* = \text{argmin}\left(\sum_i \left(\hat{o}^L(i) - o(i)\right)^2\right)$, where $o(i)$ is the correct output and $\hat{o}^L(i)$ is the output calculated by the learner L. In the on-line version, at each time step t, the presented output o_t is compared to the estimated output $\hat{o}^L(i_t)$, resulting in the prediction error $e_t = \hat{o}^L(i_t) - o_t$. Depending on the specific learning algorithm, the learner parameters χ are updated according to the prediction error. *Learning* is defined as the change in the learner parameters: the larger the prediction error, the bigger the change to the parameters, resulting in the proverb "you learn from your mistakes".

We focus on autonomous or internally supervised learning, which means that the training set, i.e. the correct input–output pairs, are internally accessible to the agent, without needing an external teacher. The dynamic internal models of an active sensor, e.g. the forward and inverse models, are examples of such learnable systems. The forward model (FM) predicts the future state of the system s_{t+1}, given the current state s_t and the action performed a_t: $(s_t, a_t) \rightarrow s_{t+1}$. Learning the forward model means learning the state-transition function. The inverse model (IM) predicts the required action given the current state and the future desired state: $(s_t, s_{t+1}) \rightarrow a_t$. Learning the inverse model is usually much harder than the forward model (Jordan, 1992; Nguyen-Tuong, Peters, Seeger, & Scholkopf, 2008). These models are usually used for trajectory prediction and planning for robotic arms (Behera, Gopal, & Chaudhury, 1995; Ouyang, Zhang, & Gupta, 2006) or description of internal models in the brain (Kawato, 1999; Lalazar & Vaadia, 2008; Shadmehr & Krakauer, 2008). Many learning algorithms have been developed for them (Cheah, Liu, & Slotine, 2006; Nguyen-Tuong et al., 2008; Ouyang et al., 2006; Wainscott, Donchin, & Shadmehr, 2005). However, the training sets were always composed of random presentation of input–output pairs. The basic curiosity loop attempts to find the best input–output pair presentation by selecting the appropriate actions.

2.2. Learner-critic-actor architecture

The goal of the basic curiosity loop is to autonomously and actively learn a defined correlation in the best manner. This means that the core of each loop is the learner, which attempts to map presented input–output pairs via internally supervised learning algorithm. To this end, we combine the reinforcement learning paradigm with internally supervised learning. In conventional reinforcement learning schemes, the reward function is externally set, while in the autonomous curiosity loop the reward is intrinsic. Thus one should define the correct reward function whose maximization will result in a policy that facilitates optimal learning. From the aforementioned proverb, one can heuristically reason that maximizing (corrected) mistakes should optimize learning. Explicitly, for each presented example, the learner has a prediction error, i.e. the difference between the expected output and the correct output. Sampling places of highest prediction error guarantees that in each time step, the learner adjusts maximally. To formulate this heuristics in the reinforcement learning framework, we set the reward to be the square of the learning error. In this scheme, we have incorporated the reward into the system, i.e. it is now an internal reward, both controllable and modifiable (Oudeyer et al., 2007; Simsek & Barto, 2006). The curiosity loop thus implements the reinforcement active learning (ReAL) principle, i.e. actively learning via reinforcement of prediction errors (Gordon & Ahissar, 2011).

We have chosen the actor-critic design of RL (see below), in which the critic learns to predict the value of each state and computes the temporal-difference (TD) error. The actor learns, according to the TD error, to perform the actions that will maximize accumulated future rewards. By incorporating the learner into the scheme by virtue of intrinsic reward, we have introduced a delicate interplay between the three approximators, namely the actor, critic and learner. The actor, through the selection of the appropriate action and the state-change induced by the outside world, determines which new example is presented to the learner. This, in turn, produces the prediction error which not only modifies the learner parameters, but also determines the intrinsic reward, which the critic now assimilates into its value and advantage approximators. The critic completes the curiosity loop by determining the TD error that updates both the critic and the actor.

Please cite this article in press as: Gordon, G., & Ahissar, E. Hierarchical curiosity loops and active sensing. Neural Networks (2012), doi:10.1016/j.neunet.2012.02.024

2.3. Reinforcement learning: incremental natural actor critic

As opposed to supervised learning, where a sample of the correct input–output mapping is given, reinforcement learning attempts to find the optimal action-selection policy that maximizes cumulative future rewards. Actor-critic (AC) algorithms are based on the simultaneous online estimation of the parameters of two structures, called the actor and the critic. The actor corresponds to an action selection policy, mapping states to actions in a probabilistic manner. The critic corresponds to a value function, mapping states to expected cumulative future reward. Thus, the critic addresses a problem of prediction, whereas the actor is concerned with control. These problems are separable, but are solved simultaneously to find an optimal policy, as in policy iteration. In the RL framework, the state, action and reward at each time $t \in \{0, 1, 2, \ldots\}$ are given by $s_t \in S$, $a_t \in A$ and $r_t \in \Re$ respectively. The environment's dynamics are characterized by state-transition probabilities $p(s_{t+1}|s_t, a_t)$, and single-stage expected rewards $r(s, a) = E[r_{t+1}|s_t = s; a_t = a] \forall s_t, s_{t+1} \in S; \forall a \in A$. The agent selects an action at each time t using a randomized stationary policy, designated as the *actor*:

$$\pi(a|s) = \Pr(a_t = a|s_t = s; \lambda) \tag{1}$$

where λ are the actor parameters to be tuned. The long-term average reward per step under policy π is defined as

$$J(\pi) \underset{\lim T\to\infty}{=} \frac{1}{T} E\left[\sum_{t=0}^{T-1} r_{t+1}|\pi\right]. \tag{2}$$

Our aim is to find a policy π that maximizes the average reward, i.e. $\pi^* = \text{argmax} J(\pi)$. In the average reward formulation, a policy π is assessed according to the expected differential reward associated with states s or state-action pairs (s, a). For all states and actions, the differential action-value function and the differential state-value function under policy π are defined as

$$Q^\pi(s, a) = \sum_{t=0}^{T} E[r_{t+1} - J(\pi)|s_0 = s, a_0 = a, \pi] \tag{3}$$

$$V^\pi(s) = \sum_{a\in A} \pi(a|s) Q^\pi(s, a). \tag{4}$$

The *critic* attempts to learn the value function, $\hat{V}^\pi(s; \nu)$ by tuning the parameters ν using

$$\phi_\nu(s) = \nabla_\nu \hat{V}^\pi(s; \nu). \tag{5}$$

The reinforcement learning algorithm we implement uses temporal difference (TD), here taken to be

$$\delta_t = r_{t+1} - \hat{J}_{t+1} + \hat{V}^\pi(s_t; \nu_t) - \hat{V}^\pi(s_{t+1}; \nu_t) \tag{6}$$

where $\hat{J}_{t+1}$ is the estimated average reward, which is also updated.

We implemented the incremental natural actor-critic (iNAC) algorithm, presented in Bhatnagar et al. (2007). While several different AC update algorithms were presented, three with natural-gradient and two with the Fisher-information matrix, after intensive numerical analysis, the natural-gradient AC with advantage parameters has proven to be the most efficient and will henceforth be used. The critic update state includes both the update of the value ν_t and advantage parameters w_t concurrently:

$$\nu_{t+1} = \nu_t + \alpha_t \delta_t \phi_\nu(s_t) \tag{7}$$

$$w_{t+1} = \left[I - \alpha_t \psi(s_t, a_t) \psi(s_t, a_t)^T\right] w_t + \alpha_t \delta_t \psi(s_t, a_t) \tag{8}$$

$$\psi_\lambda(s, a) = \nabla_\lambda \log \pi(a|s; \lambda) \tag{9}$$

where $\psi_\lambda(s, a)$ are compatible features derived from the actor. The actor update thus depends on the advantage parameters:

$$\lambda_{t+1} = \lambda_t + \beta_t w_{t+1}. \tag{10}$$

In Eqs. (7)–(10) the step-size schedule for the critic and actor satisfy the condition that the critic converges faster than the actor:

$$\alpha_t = \alpha_0 \left(\frac{\alpha_c}{\alpha_c + t}\right)^{0.9}, \qquad \beta_t = \beta_0 \left(\frac{\beta_c}{\beta_c + t}\right). \tag{11}$$

2.4. iNAC and learner

The curiosity loop is based on a learner that attempts to learn the input–output correlation, where the input and output can be defined in the space of states and/or actions. This learner implements supervised learning algorithm, which inherently calculates the prediction error. In the curiosity loop, this error is used twice. It updates the learner parameters in the supervised learning algorithm. It is also used as the intrinsic reward for the RL algorithm in the following manner:

$$r_{t+1} = e_t^2 = \left(\hat{o}^L(i_t) - o_t\right)^2. \tag{12}$$

Although we implemented *incremental* NAC, and not episodic NAC (Peters & Schaal, 2005), time has been segmented into episodes to signify the developmental time axis of the process in which the actor and critic has drastically altered through accumulated changes. Each episode $n = 1, \ldots, N_E$ was composed of time-steps $\tau = 0, \ldots, T - 1$ signifying the on-line time axis of the learning process in which small changes are accumulated via the incremental NAC algorithm. While the actor and critic continuously change throughout both the developmental and on-line times, i.e. each time-step in each episode is the aforementioned $t = 0, \ldots, N_E \times T$ parameter; the learner was reset at the beginning of each episode, embodying the beginning of a new map-learning sequence. For example, the object localization curiosity loop is composed of many episodes, each with a new position of the object, and hence a new learned object localization map. However, the changes in the actor and critic accumulated in an incremental manner in all episodes and all time-steps. At the end of the run, i.e. after many episodes, the assessment of the learned actor is done on an on-line time-scale, i.e. the learner is reset and a new sequence using the learned actor is performed, *without* changes to the actor or critic. The dynamics of this assessment sequence is then analyzed to evaluate the functionality of the learned actor.

2.5. Implementation

We have implemented a novel continuous state continuous action actor and critic configuration that drastically expedites the convergence of the RL process. Showing comparative results between this algorithm and other used is not the focus of the work presented here and is beyond the scope of this paper. The novelty rises from using a weighted sum of radial basis functions as an approximator for both actor and critic, with *tunable* centers and widths. The details of this novel approach are presented below.

Actor. The parameterized policy is a weighted sum of N_a radial basis functions (RBF):

$$\pi(a|s; \lambda) = \frac{\left(\lambda_{\text{amp}}^2\right)^T \phi\left(s, a; \lambda_{a,\text{cen}}, \lambda_{a,\sigma}, \lambda_{s,\text{cen}}, \lambda_{s,\sigma}\right)}{\left(\lambda_{\text{amp}}^2\right)^T \tilde{\phi}\left(s; \lambda_{a,\text{cen}}, \lambda_{a,\sigma}, \lambda_{s,\text{cen}}, \lambda_{s,\sigma}\right)} \tag{13}$$

$$\begin{aligned}\phi_i\Big(s, a; \lambda_{a,\text{cen}}^{(i)}, \lambda_{a,\sigma}^{(i)}, \lambda_{s,\text{cen}}^{(i)}, \lambda_{s,\sigma}^{(i)}\Big) &= \exp\Big(-\left(a - \lambda_{a,\text{cen}}^{(i)}\right)^T \lambda_{a,\sigma}^{(i)} \left(a - \lambda_{a,\text{cen}}^{(i)}\right) \\ &\quad - \left(s - \lambda_{s,\text{cen}}^{(i)}\right)^T \lambda_{s,\sigma}^{(i)} \left(s - \lambda_{s,\text{cen}}^{(i)}\right)\Big)\end{aligned} \tag{14}$$

$$\tilde{\phi}(s) = \int_{-\infty}^{\infty} da \phi(s, a) \tag{15}$$

Please cite this article in press as: Gordon, G., & Ahissar, E. Hierarchical curiosity loops and active sensing. Neural Networks (2012), doi:10.1016/j.neunet.2012.02.024

where $\lambda^{(i)}_{a,\sigma}$, $\lambda^{(i)}_{s,\sigma}$ denote the ith RBF width around the central action $\lambda^{(i)}_{a,\text{cen}}$ and state $\lambda^{(i)}_{s,\text{cen}}$, respectively. This policy and its derivatives can be solved analytically, hence greatly expediting the numerical evaluations.

Critic. The critic feature functions are N_c RBFs,

$$\hat{V}^{\pi}(s;\nu) = \nu^{T}_{\text{amp}}\xi\,(s;\nu_{\text{cen}},\nu_{\sigma}) \tag{16}$$

$$\xi_j\left(s;\nu^{(j)}_{\text{cen}},\nu^{(j)}_{\sigma}\right) = \exp\left[-s\left(s-\nu^{(j)}_{\text{cen}}\right)^{T}\nu^{(j)}_{\sigma}\left(s-\nu^{(j)}_{\text{cen}}\right)\right]. \tag{17}$$

The novel ability to learn the center and width of the RBFs results in a need for only a small number of RBFs, even for high dimensional actors, since most optimal actors can be represented by a small number of RBFs yet in specific action-space locations. This drastically reduces the computation time and the exponential explosion of higher dimensional policies.

Learner. We have used a feed-forward neural network with two input neurons, two hidden layers and one output neuron, with linear, hyperbolic tangent and symmetric saturated linear transfer functions, respectively. The supervised learning algorithm is a standard gradient descent learning rule with adaptive learning rate and momentum. The algorithm was implemented in C#, using parallel processing on multiple cores to reduce the running time, which was long due to the required averaging of the stochastic actors.

In the results presented below, we compare the performance of the learned actor after several developmental episodes to that of a random actor, i.e. generation of uniformly distributed random actions. Since one of our main goals is to develop an *autonomous* architecture, the fully random actor is the only a priori actor that is plausible for comparison. Any designed actor, e.g. a central pattern generator or feedback loops, is not an eligible actor for comparison since it is not learned or developed. The actors learned via the curiosity loop always start from a random-like behavior, i.e. their parameters are set such that a random action will be produced, and continue until their parameters converge.

In order to assess the performance of a given actor, we calculate the dynamics of generalization error of the learner during the assessment episode. The generalization error is given by the difference between the true mapping, calculated from the given world function, and the learned mapping of the learner, over all the phase-space (here it is a 40×40 equidistant points) of the mapping. Unless otherwise stated, the generalization error is averaged over 20 runs that start from different equidistant starting points.

3. Vibrissae internal models' curiosity loops

We implemented the curiosity loop architecture in the rodent's vibrissae system, which implements active sensing via whisker motion. In this section, we first present the numerical setup of the vibrissae system, describing the relevant state and action spaces. We then analyze three distinct basic curiosity loops of the internal models (Shadmehr & Krakauer, 2008) of the whiskers, namely the forward model loop, the inverse model loop and the combined internal model loop.

As the most basic actions, we consider the motor commands of motor-neurons that activate muscles, whose contraction move the whisker follicle and with it, the whisker (Hill, Bermejo, Zeigler, & Kleinfeld, 2008; Simony et al., 2010). The basic sensory information is taken to be the whisker angle. While proprioception muscle spindles have not been found in the internal muscle of the whisker pad, whisking cells, i.e. sensory neurons that (approximately) code angle position, have been reported (Szwed et al., 2003). Touch cells, i.e. sensory cells that are activated during whisker touch of an object, have also been found (Szwed et al., 2006), and we examine their related loop in this section.

In the whisker context, the FM is imperative since there are inherent delays in the whisker dynamics (Hill et al., 2008; Simony et al., 2010), and stable control of such a highly non-linear system requires efficient and accurate prediction. The inverse model (IM) learns to anticipate the required motor command that will achieve a desired angle. The IM, once learned, can serve as a coordinate transformation of the whisker dynamics, such that both the sensory information and the actions are given in whisker angle coordinates. One can learn the forward and inverse models concurrently, since they do not depend on one another. A random actor can accommodate learning both models (Wolpert & Kawato, 1998). Can a single curiosity loop for both learners, FM and IM, outperform a random actor? To answer this question, we have constructed a combined loop, where the intrinsic reward is the sum of the square of both forward model and inverse model learners' prediction errors.

These loops attempt to find the best policy that optimizes learning motor command and whisker angle correlations, when there are *no external objects* in the environment. The scenario in which objects are actively sensed and localized is described in the next section.

3.1. Numerical setup: the vibrissae system

The one-dimensional state-space is composed of the normalized whisker angle $s^{(1)}_t = \theta_t \in [-1,1]$, where $\theta = \pm1$ corresponds to a fully protracted (retracted) whisker. The one-dimensional action-space is the normalized motor command to the whisker follicle and controls the velocity of the whisker, $a^{(1)}_t = m_t \in [-1,1]$. The world function (also known as state-transition or sensor transformation function) is taken to be linear $\theta_{t+1} = \theta_t + Am_t$, while keeping the state inside the state-space boundaries. Hence, depending on A the forward and inverse maps may be saturated at specific state/action areas.

To make the model more realistic, we have added internal noise to both the sensed whisker angle and the sent motor command. Hence, the simulated state-transition function is $\theta_{t+1} = \theta_t + A\left(m_t+\epsilon^{(m)}\right)$ and the sensed angle is given by $\theta_t + \epsilon^{(\theta)}$, where $\epsilon^{(m)}, \epsilon^{(\theta)} \sim$ uniform $[-0.01, 0.01]$. Furthermore, the world function itself is random from episode to episode, reflecting different whisker compliances resulting from e.g. whiskers' changing length. Here we set $A = 0.25\cdot\left(1+\epsilon^{(A)}\right)$, $\epsilon^{(A)} \sim$ uniform$[-0.25, 0.25]$.

3.2. The forward model loop

Fig. 1(a) shows the architecture of the forward model curiosity loop. The algorithm was run for 10,000 episodes over which the critic and actor were learned. Fig. 1(b–d) presents the analysis of the assessment sequence, i.e. the on-line time steps after the 10,000 developmental episodes, with the converged actor. Fig. 1(b) shows the learned mapping at the end of the sequence, where the true forward model mapping, given by the world function with $A = 0.25$ is a saturated linear mapping with two non-linear sharp transitions that appear due to the bounded state-space. As can be seen, the learner approximates the true forward model very well. Fig. 1(c) shows the generalization error as a function of time steps for the converged actor (black) and a random actor (gray). As can be seen, the learned actor is significantly better than the random one during the initial stages of learning, after which both converge to a very good approximation of the true forward model. A characteristic trajectory of the converged actor (black) and a random actor (gray) is plotted in the upper panel of Fig. 1(d), and in the lower panel the accumulated reward is shown, demonstrating the fact that the learned actor indeed accumulates more reward, i.e. more learning errors, than a random actor.

Please cite this article in press as: Gordon, G., & Ahissar, E. Hierarchical curiosity loops and active sensing. Neural Networks (2012), doi:10.1016/j.neunet.2012.02.024

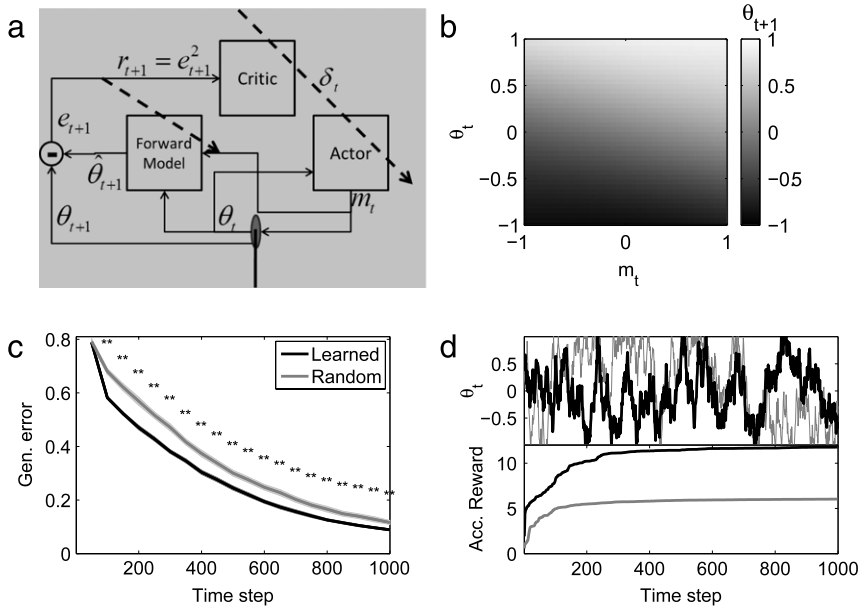

Fig. 1. Forward model curiosity loop. (a) Loop architecture. (b) Learning forward model mapping, θ_{t+1} is intensity coded. (c) Generalization error averaged over 20 actors (standard-error too small to notice) for actors after 10,000 episodes (black) and random actors (gray). (d) Upper panel: a typical trajectory of the learned (black) and random (gray) actors. Lower panel: accumulated reward of the same trajectory.

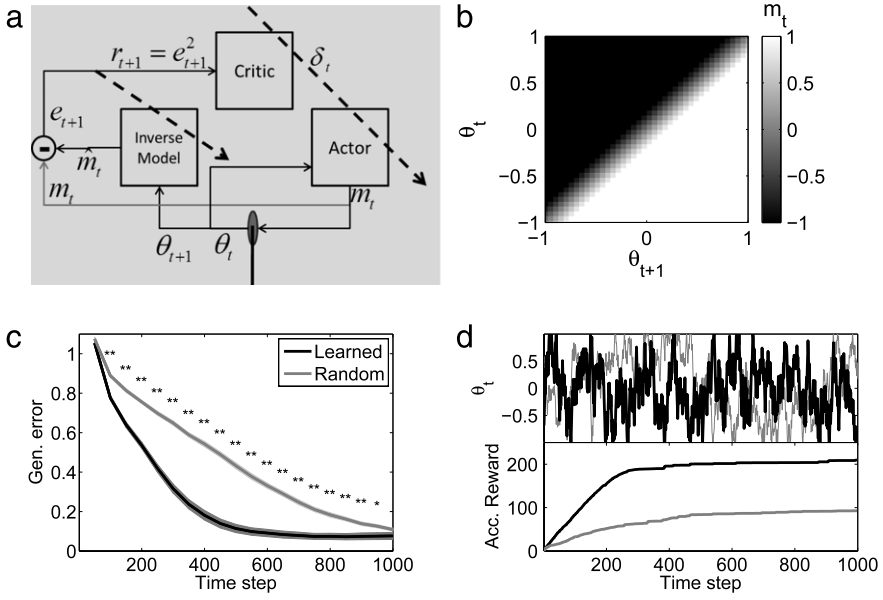

Fig. 2. Inverse model curiosity loop. (a) Loop architecture. (b) Learning inverse model mapping, m_t is intensity coded. (c–d) similar to Fig. 2.

3.3. The inverse model loop

Fig. 2(a) shows the architecture of the inverse model curiosity loop. Fig. 2(b) shows the learned mapping at the end of the sequence, where the true inverse model mapping, is also a saturated linear mapping with two non-linear sharp transitions that appear due to the bounded state-space, yet the transitions appear in the center of the mapping, opposed to the forward model, Fig. 1(b). As can be seen, the learner approximates the true inverse model very well. Fig. 2(c) shows that such the converged actor generalizes much better than the random actor. The reward accumulated, i.e. the accumulated prediction error, is higher in the initial time steps of the learning, Fig. 2(d) lower panel, thus emphasizing its fast and drastic improvement over random actions.

3.4. The combined loop

One can learn the forward and inverse models concurrently, since they are in the same level of hierarchy and do not depend on one another. A different architecture may then arise, one that receives rewards from both learners, but has a single critic and a single actor, that try to optimize the concurrent learning of both internal models. A random actor will surely accommodate learning both models, but can a single ReAL loop for both learners, FM and IM, outperform a random actor?

To answer this question, we have constructed an architecture, where the intrinsic reward is the sum of the square of both forward model and inverse model learners' prediction errors, Fig. 3(a). Also, the computed generalization error is the sum of the generalization

Please cite this article in press as: Gordon, G., & Ahissar, E. Hierarchical curiosity loops and active sensing. Neural Networks (2012), doi:10.1016/j.neunet.2012.02.024

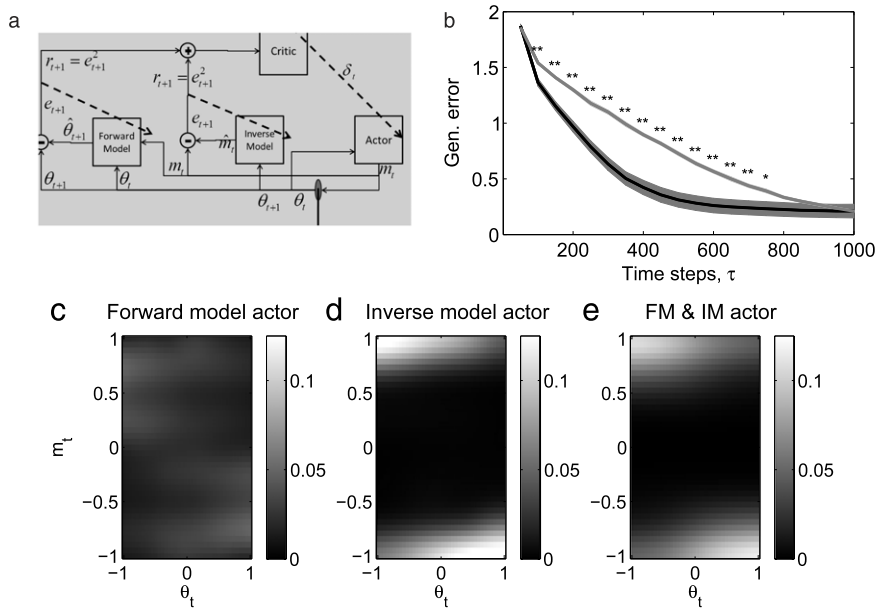

Fig. 3. Combined internal models curiosity loop. (a) Loop architecture. (b) Generalization error averaged over 20 actors (standard-error too small to notice) for actors after 10,000 episodes (black) and random actors (gray). (c–e) Learned actors for the (c) forward model loop, (d) inverse model loop and (e) combined forward and inverse model loops. Color codes probabilities of performing the actions m_t.

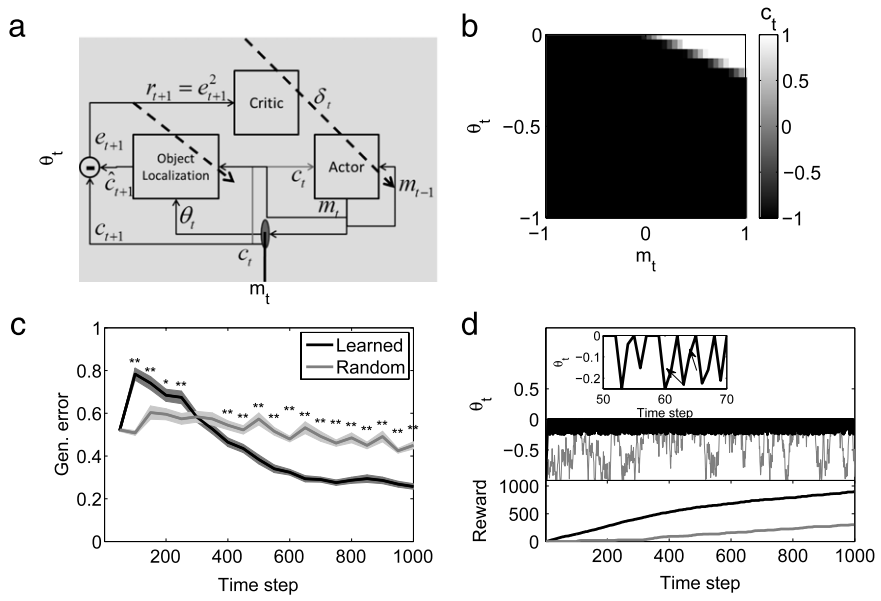

Fig. 4. Object localization curiosity loop. (a–d) similar to Fig. 2.

errors of both mappings. Fig. 3(b) shows that such an actor indeed outperforms the random actor. The combined actor, Fig. 3(e), presents the shared features of the FM, Fig. 3(c), and IM actors, Fig. 3(d), yet the IM actor features are more prominent, since the IM rewards are much larger than the FM rewards and hence have more influence on the progression.

4. Active sensing loop: object localization

To obtain object location, a touch signal is added to the state-space, $s_t^{(2)} = c_t \in \{-1, 1\}$, which may originate from a touch sensor (discussed in the current section) or from error in the learned free-air forward model (discussed below). In both schemes, the touch signal is binary, i.e. either there is touch with an object or there is not, where the strength of the touch that depends on its radial position along the whisker (Birdwell et al., 2007) and on the force of the whisker muscles (Simony et al., 2010), is neglected for simplicity. Object localization is defined here as the forward model of the touch signal, given the current whisker angle and performed action $(\theta_t, m_t) \rightarrow c_t$. We first discuss the scenario where the touch signal originates from a touch sensor, i.e. it has an external source.

Fig. 4(a) shows the architecture of the object localization curiosity loop. As can be seen, the actor receives non-Markovian information from previous time-steps. Fig. 4(b) shows the learned mapping at the end of the assessment sequence. The true mapping, i.e. object localization mapping for an object in the middle of the whisker field is confined to a small region around the object location and has a step-like shape, where far from the object there is no touch

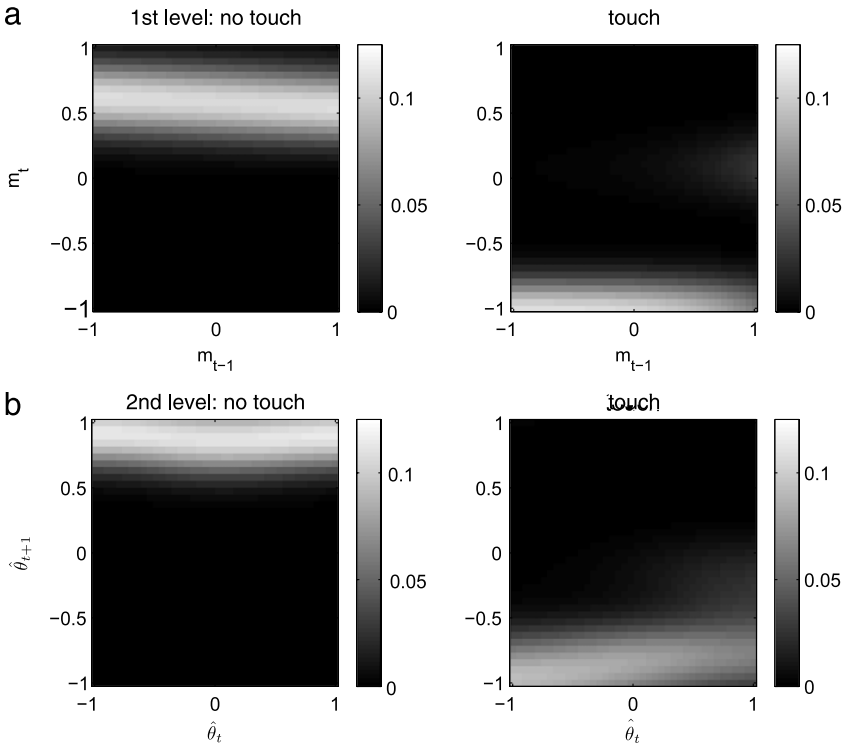

Fig. 5. Object localization actors. (a) Object localization actor for 1st level curiosity loop, i.e. when touch is supplied by external touch signal via touch cells, Fig. 4(a). (b) Object localization actor for 2nd level curiosity loop, i.e. when touch is given by error of learned free-air forward model and the output is the desired whisker angle directed to the learned free-air inverse model, Fig. 6(a–b) left panel gives the probability for an action when there is no touch, $c_t = -1$ and right panel when there is touch $c_t = 1$.

information and moving towards the object results in touch information, since the whisker motion is blocked by the object. As can be seen, the learner approximates the true mapping very well.

In this example, the whisker always started from full retraction and the object was encountered during protraction. Hence, the generalization error of the mapping is computed only on one side that is determined by the initial state and the position of the object. Fig. 4(c) shows that the learned actor significantly outperforms the random actor. The actor used in this scenario was a non-Markov actor, in which the action depends on the current touch information and the previous action $\pi\ (m_t|c_t, m_{t-1})$. Examining the trajectory of the learned actor, Fig. 4(d: upper panel), reveals that this actor *actively senses* the object localization linear separator. This palpation whisking, i.e. alternating between touching and not touching the object, drastically increases the accumulated rewards (prediction errors), Fig. 4(d): lower. The learning curve in Fig. 4(c) shows an initial worsening and then a drastic improvement in the generalization error. The former is due to the fact that the actor has learned to protract until an object is reached, avoiding an initial random exploration, which results in delayed learning of the large no-touch area. Following object-touch, which occurs after a small number of time steps, the palpation behavior learns the linear separator of the touch and no-touch areas and thus drastically reduces the generalization error. The random actor, on the other hand initially learns the large no-touch area, but due to a small number of touch events, fails to converge on the right linear separator that defines the object location.

The actor, shown in Fig. 5(a), where the left (right) panel shows the action probabilities for no touch (touch), demonstrates a negative feed-back behavior, where it protracts when not touching an object and retracts when it touches. However, the fact that the protraction is smaller than the retraction allows sampling the two sides of the linear separator, namely, after a large retraction follows two smaller protraction, the first without touch and the second with, indicated by the arrows in Fig. 4(d): upper, inset. Furthermore, the probability of slightly protracting after a large protraction that induced touch, a non-Markovian feature, allows sampling of the most non-linear feature, namely, the intersection point of the linear separator with the object position at $m_t = \theta_t = 0$. Together, these unique features enable the active learning of the object localization linear separator.

5. Hierarchical loops

After introducing the basic curiosity loop components, one can next consider the buildup of hierarchical loops.

5.1. Architecture

We wish to start from the most basic possible loop and investigate the possible formation of higher loops. The hierarchical architecture of actively learning loops implies two types of loops, depending on the learned correlations: a forward loop and an inverse loop. The forward loop, based on the predictive forward models, utilizes lower loops' information to learn new correlations, but with the same repertoire of actions. In RL jargon, this amounts to a gradual increase in the state dimensionality. These states are not in the physical world, e.g. new sensory information, but rather new learned correlations that were previously "hidden" within the motor-sensory information. For example, learning the forward model of a rat's whisker in free space, i.e. predicting the next whisker angle given a motor command, can be utilized to learn when the whisker has touched an object by comparing the learned prediction of its angle with the sensed angle. This is known as angle absorption (Szwed et al., 2003) and is a new state of the system that can be used for further learning.

Please cite this article in press as: Gordon, G., & Ahissar, E. Hierarchical curiosity loops and active sensing. Neural Networks (2012), doi:10.1016/j.neunet.2012.02.024

The inverse loop, on the other hand, utilizes lower loops correlations to increase the repertoire of available actions. In RL this amounts to a gradual increase in the action dimensionality. Again, these new actions do not require new motor plants, but rather they are new motor commands that use motor primitives based on learned "hidden" correlations between the available sensory-motor information. For example, learning the inverse model of the rat's whisker, i.e. what should the motor command be in order to reach a desired angle, can be used as a new action in higher loops. This means that higher loops' action can be given in whisker angle coordinates, instead of muscle motor commands.

The hierarchical curiosity loops paradigm results in an ever increasing repertoire of states and actions available for the agents for its use. Since each loop actively learns new correlations based on lower loops that were already actively learnt, gradual change in behavior emerges. The optimal action for learning each loop can be drastically different, making the buildup of the hierarchical structure of the network being accompanied by characteristic behaviors.

5.2. Developmental inter-loop switching

Since there are many possible loops in different levels of the hierarchy and each loop has its own actor, how is the transition between one loop and the next determined? How does the system autonomously know that the optimal actor has been reached? This developmental inter-loop switching is easily determined within the RL framework of TD learning, since the TD error indicates whether convergence has been reached or not. Once the TD error has gone below a certain low threshold and has stabilized there, there is no more change in the actor-critic component, indicating that reinforcement learning has ceased. This, in turn, can signal the next loop to start its ReAL dynamics, until it too converges. Hence, the inter-loop switching of the reinforcement learning part is achieved by ordered gating determined by each loop's TD-error.

Formally, we incorporate this switching by defining the average TD of episode n, $\langle\delta\rangle_n$, a running weighted average over episodes, Δ_n and a running weighted average of the difference, $\tilde{\Delta}_n$:

$$\langle\delta\rangle_n = \sum_{t=0}^{T-1} |\delta_t| \tag{18}$$

$$\Delta_{n+1} = \kappa\langle\delta\rangle_n + (1-\kappa)\Delta_n \tag{19}$$

$$\tilde{\Delta}_{n+1} = \tilde{\kappa}\,|\Delta_{n+1} - \Delta_n| + (1-\tilde{\kappa})\,\tilde{\Delta}_n. \tag{20}$$

Here $\kappa, \tilde{\kappa}$ are the averaging weights. The loop has converged when $\left|\tilde{\Delta}_n/\Delta_n\right| < \Delta_{\text{threshold}}$, i.e. when the average TD error in an episode has changed very little, compared to its size. Thus, for multiple-loop architecture, a developmental switching mechanism operates by sequentially going from one loop to the next whenever the former has converged. In a hierarchical configuration, the sequence of loops must follow the hierarchy, i.e. lower loops are learned before higher loops.

5.3. On-line inter-loop switching

Another type of inter-loop switching must be accounted for, namely, the on-line gating between optimal actors, after they have been learned. For example, once the optimal actor for the internal models has been reached, as well as the optimal actor of the object localization, how does the system autonomously determine if the current internal models are the correct ones and can be used for object localization? This on-line switching, between the optimal actors is done by learner error's gating, e.g. once the internal model learners' error has decreased below a certain threshold, the motor commands' control switches from the internal models' actor to the object localization actor. Formally, the actor on-line switching is implemented by defining a running reward average,

$$\rho_{\tau+1} = \gamma e_t^2 + (1-\gamma)\,\rho_\tau. \tag{21}$$

When it decreases below a certain threshold, $\rho_{\text{threshold}}$, the control is switched to the next loop's actor and the average reward is reset. This can be continued for all loops in the hierarchy.

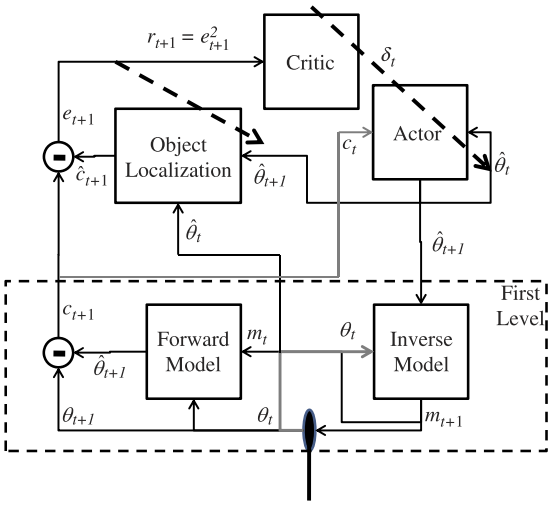

Fig. 6. Object localization hierarchical curiosity loop architecture.

6. Vibrissae's hierarchical loops

The lowest loops that autonomously and actively learn internal models can be used by higher loops, e.g. object localization (Fig. 6). Here, we show two major contributions of lower loops, namely, the forward model increases the state-space, and the inverse model increases or substitutes the action-space. (i) To demonstrate the contribution of the forward model, we consider the touch signal as the error in the already learned free-air forward model:

$$c_t = \begin{cases} 1 & \left|L^{FM}(\theta_t, m_t) - \theta_{t+1}\right| > \theta_{\text{threshold}} \\ -1 & \text{otherwise.} \end{cases} \tag{22}$$

This signal means that if the difference between the predictions of where the whisker should be and where it actually is, often dubbed angle absorption (Szwed et al., 2003), is greater than a certain threshold, there is contact with an object and a touch signal is generated. If a good forward model was learned when there were no objects, the appearance of objects in the whisker field, and the fact that the whisker movement is blocked by the object, gives the touch signal. Hence, the forward model, which was learned in a lower loop, has increased the state-space from 1 (whisker angle) to 2 (angle and touch signal). This is in contrast to the external touch signal considered above, which can be mediated by specialized touch cells. Here, the touch information is conveyed by the same whisking cells, yet augmented by the learned free-air forward model.

(ii) The inverse model can substitute higher loops action-spaces, i.e. change the actors' output from motor command m_t to the desired whisker angle $\hat{\theta}_{t+1}$. This means that higher loop actors will change from $\pi\left(m_t|\theta_t\right) \rightarrow \pi\left(\hat{\theta}_{t+1}|\theta_t\right)$ and will send their output to the inverse model $L^{IM}\left(m_t|\theta_t, \hat{\theta}_{t+1}\right)$ that will in turn generate the proper whisker motion.

One must also consider developmental inter-loop switching. The TD-error of each loop can signal the transition between one

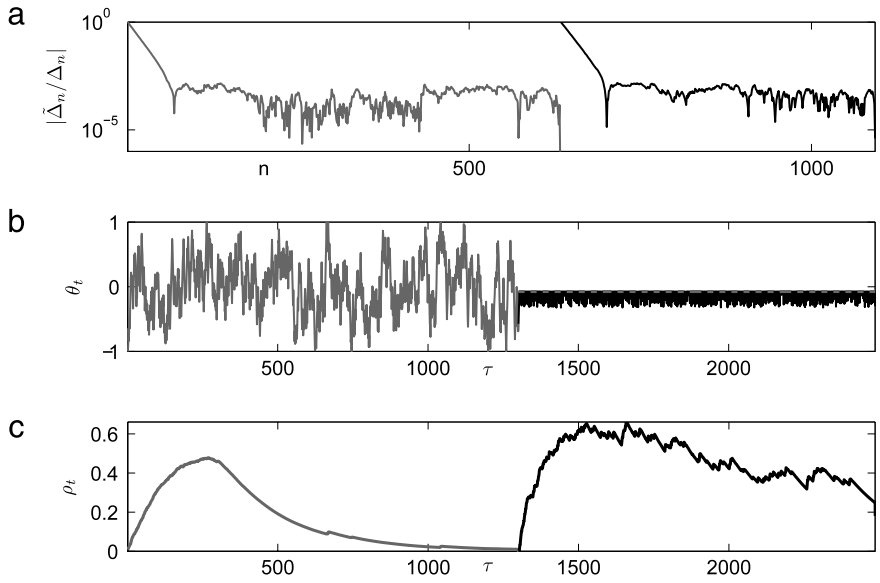

Fig. 7. Hierarchical curiosity loops. (a) Stability of TD error and developmental switching between internal model curiosity loop (gray) and object localization curiosity loop (black) as a function of episodes. (b) Execution trajectory as a function of time-steps for learning internal models (gray) and then on-line switching to object localization behavior (black). Object position is indicated by dashed gray line. (c) Running reward average as a function of time-steps for internal models (gray) and object localization (black), showing when on-line switching has occurred.

loop and another's according to Eq. (6). Furthermore, once the actors have been learned, an execution of the whole network is performed, where the actors are switched on-line, according to Eq. (21). In the example shown in Fig. 7, two loops, namely a combined forward/inverse curiosity loop and an object localization loop are sequentially switched by their respective TD-error, Fig. 7(a). During the lower loop learning, an actor that optimizes the learning of both the forward and inverse models is found, similar to Fig. 3(e). After the TD error of that loop has stabilized, the object localization curiosity loop commences and finds the optimal actor for object localization, Fig. 5(b). The actor is described by $\pi\left(\hat{\theta}_{t+1}|c_t, \hat{\theta}_t\right)$, i.e. it is a non-Markovian actor that sends the desired angle to the inverse model, based on the current (binary) contact information and the previously-sent desired angle. Comparing the object localization actors, when the touch signal is internal vs. external, and the output is motor command vs. desired angle, one can see that the overall behavior is very similar, as expected. The actor behaves as a feedback loop which protracts the whisker when there is no touch and retracts it when there is touch.

Once all the hierarchy levels actors have been learned, an execution of the hierarchical loops is performed. Initially, the internal models are learned using the lower level learned actor. Once the running average reward has reached a certain threshold, Fig. 7(c), the system automatically switches to the object localization learned actor, until it too reaches the reward threshold. The trajectory in Fig. 7(b) shows a typical on-line switching between periodic whisking that optimizes internal model learning and object palpation that optimizes object localization.

7. Discussion

We have presented a parsimonious model by which action of a curious agent develops in order to optimize its autonomous learning capabilities in a hierarchical manner. The model has an innate (i.e. fixed) architecture, developmentally learned parameters, and a hierarchical nature that enables scaffolding, i.e. mappings that are learned in low levels are used as components in higher levels of the hierarchy (Weng, 2004). Each hierarchical loop architecture is composed of learners (predictors), intrinsic rewards and RL controllers, namely, critics and actors (Schmidhuber, 2010). All the learners, critics and actors parameters are autonomously learned during development. Hence the model does not rely on external teachers, or on pre-designed behaviors, but rather on the actor-critic-learner architecture that can be applied to any autonomously learned mapping, using any internally supervised mechanism (Ahissar, Abeles, Ahissar, Haidarliu, & Vaadia, 1998; Ego-Stengel, Shulz, Haidarliu, Sosnik, & Ahissar, 2001). The implementation of the model presented here has two types of predictors, namely, state and action predictors (forward and inverse models, respectively). The intrinsic reward is the (square of the) prediction error. Finally, each hierarchical loop learns its own critic and actor, where the latter controls (either directly or via an inverse model) only external actions and not internal states (Schmidhuber, 2010).

Equating the reward to the prediction error has its limitations. For example, in an environment wherein one part is deterministic while the other is mostly noise, the agent will remain in the noisy regime, since its prediction error will always be high there (Oudeyer et al., 2007). While the change in prediction error is more commonly used (Oudeyer et al., 2007; Schmidhuber, 2010), the prediction error suffices in the implementation described here since the model world is not noisy and the predictors are capable to learn their objectives. Furthermore, the hierarchical curiosity loop architecture deals only with the *exploration* and not *exploitation*, i.e. the only goal of the agent is to learn as much as possible and not to utilize what it has learned to reach another goal. This was important in order to completely distinguish between curiosity and other drives that may promote behavior. In the vibrissae system, we believe that curiosity is the most dominant drive since it is a major active sense of the rat and hence one of its main goals is to acquire as much information as possible. We believe that other goals, such as hunger or fear, do not play a major role in whisker dynamics and hence our model captures the essence of the emergence of whisking.

The model describes the emergence of behavior and thus promotes mainly developmental predictions, i.e. behaviors and their underlying neural circuitry during the critical period of development in pups. One straightforward prediction is that pups do not

Please cite this article in press as: Gordon, G., & Ahissar, E. Hierarchical curiosity loops and active sensing. Neural Networks (2012), doi:10.1016/j.neunet.2012.02.024

whisk periodically in free-air or palpate novel objects immediately, i.e. once their whiskers are grown enough to reach objects, the model predicts that their first behavior should be quasi-random. This prediction has been recently supported in Grant, Mitchinson, and Prescott (2011). Furthermore, the converged behaviors are strongly dependent on the experience of the pup, thus changing pups' experience should produce different emergent behaviors. For example, partially paralyzing the mystacial pad muscles during development, i.e. reducing their responsiveness and contracting strength, should result in a different free-air whisking when they are adults, even if at adulthood there is no paralysis. Similarly, affecting the sensory input during development, e.g. via pharmacological manipulations along the sensory pathway, should result in markedly different behaviors in adulthood. Furthermore, preventing whisker-object touch during development, e.g. by attaching plastic cones to the snout, should result in the lack of palpation behavior during adulthood. The analysis of pup whisker behavior is in its infancy (Grant et al., 2011), but the rapid advancement of tracking techniques can expedite the performance of the proposed experiments.

The model also predicts novel neural circuitry during development. In order to facilitate rewarding prediction error, there should be a strong *input* connectivity to the rewarding system from internal model areas, e.g. cerebellum (Lalazar & Vaadia, 2008; Shadmehr & Krakauer, 2008) and sensory perception areas, e.g. primary sensory cortex. The model predicts that this connectivity should be stronger during development to allow convergence of the stereotypical whisking behaviors apparent in adult rats. Furthermore, the conveyed information in these connections should code error signals (Lalazar & Vaadia, 2008; Shadmehr & Krakauer, 2008). The anatomical and functional circuitry of the developing pup is mostly unknown, yet the underlying infrastructure for the proposed curiosity loops should be evident to corroborate the proposed model.

The novel features of the work presented here are (i) a bottom-up approach of actively learning basic correlations, such as internal models, hence requiring continuous state and continuous actions, in contrast to higher-level discrete states and actions (Barto, Singh, & Chentanez, 2004; Oudeyer et al., 2007); (ii) implementation of state-of-the-art RL algorithms with novel critic and actors with high-dimensional continuous states and actions; (iii) a hierarchical buildup of learned correlations, as opposed to segmentation of the forward model into different subspaces (Oudeyer et al., 2007); (iv) learning of both forward and inverse models that extend both the states and actions repertoire, respectively, as opposed to learning just the forward model (Barto et al., 2004; Oudeyer et al., 2007); (v) example from the biological regime of active sensing in whisking rats, showing that typical behaviors emerge from the proposed model.

To summarize, we have introduced and analyzed the curiosity hierarchical loop algorithm that combines the reinforcement and active learning paradigms into a single framework and extends it to bottom-up hierarchical loop architecture. By setting the reward as the prediction error of another learner, we have shown that ReAL results in actors that expedite learning compared to random actions. We have implemented and extended the state-of-the-art reinforcement learning algorithm iNAC and expanded it to include the learner as the new reward function. We have also presented examples of learning the forward and inverse models. In one case, namely, object localization, this trajectory could only be produced by a non-Markov actor that depends on the current state and the previous action. We have implemented the curiosity loop framework on a simplified whisker model, which is commonly used as an active-sensing model, and have shown that observed behaviors of rats, e.g. free-air whisking and active sensing by object palpation, emerge naturally from the proposed model.

Acknowledgments

This work was supported by EU grant BIOTACT (ICT-215910), the Israeli Science Foundation Grant No. 749/10, the United States-Israel Bi-national Science Foundation (BSF) Grant No. 2007121, and the Minerva Foundation funded by the Federal German Ministry for Education and Research. G.G. was supported by the Clore fellowship. E.A. holds the Helen Diller Family Professorial Chair of Neurobiology.

References

Ahissar, E., Abeles, M., Ahissar, M., Haidarliu, S., & Vaadia, E. (1998). Hebbian-like functional plasticity in the auditory cortex of the behaving monkey. *Neuropharmacology*, *37*, 633–655.

Ahissar, E., & Arieli, A. (2001). Figuring space by time. *Neuron*, *32*, 185–201.

Ahissar, E., & Kleinfeld, D. (2003). Closed-loop neuronal computations: focus on vibrissa somatosensation in rat. *Cereb Cortex*, *13*, 53–62.

Barto, A.G., Singh, S., & Chentanez, N. (2004) Intrinsically motivated learning of hierarchical collections of skills. *International conference on developmental learning (ICDL)*.

Behera, L., Gopal, M., & Chaudhury, S. (1995) Self-organizing neural networks for learning inverse dynamics of robot manipulator. *IEEE/IAS international conference on industrial automation and control (I A & C'95)* pp. 457–460.

Bhatnagar, S., Sutton, R., Ghavamzadeh, M., & Lee, M. 2007 Incremental natural actor-critic algorithms. *Twenty-first annual conference on advances in neural information processing systems* pp. 105–112.

Birdwell, J. A., Solomon, J. H., Thajchayapong, M., Taylor, M. A., Cheely, M., Towal, R. B., Conradt, J., & Hartmann, M. J. Z. (2007). Biomechanical models for radial distance determination by the rat vibrissal system. *Journal of Neurophysiology*, *98*, 2439–2455.

Cheah, C. C., Liu, C., & Slotine, J. J. (2006). Adaptive tracking control for robots with unknown kinematic and dynamic properties. *The International Journal of Robotics Research*, *25*, 283–296.

Diamond, M. E., von Heimendahl, M., Knutsen, P. M., Kleinfeld, D., & Ahissar, E. (2008). 'Where' and 'what' in the whisker sensorimotor system. *Nature Reviews Neuroscience*, *9*, 601–612.

Ego-Stengel, V., Shulz, D. E., Haidarliu, S., Sosnik, R., & Ahissar, E. (2001). Acetylcholine-dependent induction and expression of functional plasticity in the barrel cortex of the adult rat. *Journal of Neurophysiology*, *86*, 422–437.

Gordon, G., & Ahissar, E. (2011) Reinforcement active learning hierarchical loops. *International joint conference on neural networks (IJCNN)* pp. 3008–3015.

Grant, R. A., Mitchinson, B., Fox, C. W., & Prescott, T. J. (2009). Active touch sensing in the rat: anticipatory and regulatory control of whisker movements during surface exploration. *Journal of Neurophysiology*, *101*, 862–874.

Grant, R. A., Mitchinson, B., & Prescott, T. J. (2011). The development of whisker control in rats in relation to locomotion. *Developmental Psychobiology*,.

Hill, D. N., Bermejo, R., Zeigler, H. P., & Kleinfeld, D. (2008). Biomechanics of the vibrissa motor plant in rat: rhythmic whisking consists of triphasic neuromuscular activity. *Journal of Neuroscience*, *28*, 3438–3455.

Jordan, M. I. (1992). Forward models: supervised learning with a distal teacher. *Cognitive Science*, *16*, 307–354.

Kawato, M. M. (1999). Internal models for motor control and trajectory planning. *Current Opinion in Neurobiology*, *9*, 718–727.

Kleinfeld, D., Berg, R., & O'Conner, S. (1999). Anatomical loops and their electrical dynamics in relation to whisking by rat. *Somatosensory and Motor Research*, *16*, 69–88.

Knutsen, P. M., & Ahissar, E. (2009). Orthogonal coding of object location. *Trends in Neurosciences*, *32*, 101–109.

Knutsen, P. M., Pietr, M., & Ahissar, E. (2006). Haptic object localization in the vibrissal system: behavior and performance. *Journal of Neuroscience*, *26*, 8451–8464.

Lalazar, H., & Vaadia, E. (2008). Neural basis of sensorimotor learning: modifying internal models. *Current Opinion in Neurobiology*, *18*, 573–581.

Nguyen-Tuong, D., Peters, J., Seeger, M., & Scholkopf, B. (2008). Learning inverse dynamics: a comparison. *European symposium on artificial neural networks (ESANN)* pp. 13–18.

O'Connor, D. H., Clack, N. G., Huber, D., Komiyama, T., Myers, E. W., & Svoboda, K. (2010). Vibrissa-based object localization in head-fixed mice. *Journal of Neuroscience*, *30*, 1947–1967.

Oudeyer, P. Y., Kaplan, F., & Hafner, V. V. (2007). Intrinsic motivation systems for autonomous mental development. *Evolutionary Computation, IEEE Transactions on*, *11*, 265–286.

Ouyang, P. R., Zhang, W. J., & Gupta, M. M. (2006). An adaptive switching learning control method for trajectory tracking of robot manipulators. *Mechatronics*, *16*, 51–61.

Peters, J., & Schaal, S. (2005). Natural actor-critic. *Neurocomputing*, *71*, 1180–1190. 1352986.

Schmidhuber, J. (2010). Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development*, *2*, 230–247.

Shadmehr, R., & Krakauer, J. W. (2008). A computational neuroanatomy for motor control. *Experimental Brain Research*, *185*, 359–381.
Simony, E., Bagdasarian, K., Herfst, L., Brecht, M., Ahissar, E., & Golomb, D. (2010). Temporal and spatial characteristics of vibrissa responses to motor commands. *Journal of Neuroscience*, *30*, 8935–8952.
Simsek, O., & Barto, A.G. (2006) An intrinsic reward mechanism for efficient exploration. *23rd international conference on Machine learning* pp. 833–840.
Szwed, M., Bagdasarian, K., & Ahissar, E. (2003). Encoding of vibrissal active touch. *Neuron*, *40*, 621–630.
Szwed, M., Bagdasarian, K., Blumenfeld, B., Barak, O., Derdikman, D., & Ahissar, E. (2006). Responses of trigeminal ganglion neurons to the radial distance of contact during active vibrissal touch. *Journal of Neurophysiology*, *95*, 791–802.
Towal, R. B., & Hartmann, M. J. (2008). Variability in velocity profiles during free-air whisking behavior of unrestrained rats. *Journal of Neurophysiology*, *100*, 740–752.
Venkatraman, S., & Carmena, J. M. (2011). Active sensing of target location encoded by cortical microstimulation. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *19*, 317–324.
Wainscott, S. K., Donchin, O., & Shadmehr, R. (2005). Internal models and contextual cues: encoding serial order and direction of movement. *Journal of Neurophysiology*, *93*, 786–800.
Weng, J. (2004). Developmental robotics: theory and experiments. *International Journal of Humanoid Robotics*, *1*, 199–236.
Wolpert, D. M., & Kawato, M. (1998). Multiple paired forward and inverse models for motor control. *Neural Networks*, *11*, 1317–1329.

Please cite this article in press as: Gordon, G., & Ahissar, E. Hierarchical curiosity loops and active sensing. Neural Networks (2012), doi:10.1016/j.neunet.2012.02.024