

Neuromorphic Motivated Systems

James Daly, Jacob Brown, and Juyang Weng *Fellow, IEEE*

Abstract—Although reinforcement learning has been extensively modeled, few agent models that incorporate values use biologically plausible neural networks as a uniform computational architecture. We call biologically plausible neural network architecture *neuromorphic*. This paper discusses some theoretical constraints on neuromorphic intrinsic value systems [3]. By *intrinsic*, we mean a value system that is likely programmed by the genes, whose value bias has already taken a shape at the birth time. Such an intrinsic value system plays an important role in developing *extrinsic* values through the agent’s own experience during its life span. Based on our theoretical constraints, we model two types of neurotransmitters, serotonin and dopamine, to construct a neuromorphic intrinsic value system based on a uniform neural network architecture. Serotonin represents punishment and stress, while dopamine represents reward and pleasure. Experimentally, this model allows our simulated robot to develop an attachment to one entity and fear another.

I. INTRODUCTION

A major function of the brain is to develop circuits for processing sensory signals and generating motor actions. The signals in the brain are largely transmitted through neurotransmitters, endogenous chemicals that are sent from a neuron to a target cell across a synapse.

A. Neurotransmitters

Facing a tremendous challenge of developing circuits fully autonomously, the brain seems to use different cell types each of which is sensitive to a particular type (or multiple types) of neurotransmitters [5], [9], [23]. For example, glutamate is a type of neurotransmitter. Nerve impulses trigger release of glutamate from the pre-synaptic cell. Once released, the glutamate neurotransmitter binds a glutamate receptor, such as the NMDA receptor, in the post-synaptic neuron. A sufficient number of bindings by neurotransmitters results in the firing of the post-synaptic neuron. Each cell typically has many receptors, of many different kinds.

While glutamate and GABA are neurotransmitters whose values are largely neutral in terms of preference at the time of birth, some other neurotransmitters appear to have been used by the brain to represent certain signals with intrinsic values. For example, serotonin (5-HT) seems to be involved with punishment, stress and threats; while dopamine (DA) appears to be related to reward, pleasure, and wanting.

Therefore, 5-HT and DA, along with many other neurotransmitters that have inherent values, seem to be useful for modeling the intrinsic value system of the central nervous system and artificial neural networks.

James Daly, Jacob Brown, and Juyang Weng are with Michigan State University, East Lansing, MI, USA (email {dalyjame, brown291, weng}@cse.msu.edu). Juyang Weng is also with the MSU Cognitive Science Program and the MSU Neuroscience Program.

B. Models of value systems

In machine learning, models of reinforcement learning go beyond supervised learning by allowing an agent to learn through trial and error and then prefer actions that tend to result in the chance of reward [7], [24]. Such systems can act by exploring actions that have higher expected rewards.

A major focus of modeling in reinforcement machine learning has been the problem of delay of rewards. Q-learning [26] has been often used in computer simulations as it is *model free* — it does not require explicit estimation of probability distributions which are expensive. It is an online algorithm since it learns while the agent accumulates experience. Q-learning uses a time discount model to address the problem with delayed rewards — the system prefers recent rewards as future rewards are recursively discounted.

Psychological studies have provided rich evidence about the existence of the motivational system [11], [12], [20], [21], [18]. It is known that motivational systems are important to autonomous learning in the brain.

It is beneficial to model only the intrinsic components of the motivational system for three major reasons. First, intrinsic components are the brain’s driving forces of many other complex, higher motivational behaviors. Second, a motivational system that models only intrinsic values tends to be computationally more efficient. Third, such a system has a superior generality as it has a potential to generalized to other complex, higher motivational behaviors. In other words, many higher motivational behaviors (e.g., preferences for particular foods) emerge from learning. A rigidly modeled higher motivational behavior tends to have a limited applicability.

However, it is unclear which components of the motivational system are intrinsic and which are extrinsic. Sutton & Barto 1981 [25] modeled rewards as positive values that the system learns to predict. Ogmen’s work [15] was based on Adaptive Resonance Theory (ART), which took into account not only punishments and rewards, but also the novelty in expected punishments and rewards, where punishments, rewards, and novelty are all based on a single value. Kakade & Dayan [8] proposed a dopamine model, which uses novelty and shaping to drive exploration in reinforcement learning, although they did not provide sources of information for novelty nor a computational model to measure the novelty. Oudeyer et al. 2007 [16] proposed that the objective function a robot uses as a criterion to choose an action fall into three categories, (1) error maximization, (2) progress maximization, and (3) similarity-based progress maximization. Huang & Weng 2007 [6] proposed an intrinsic motivation system that prioritizes three types of information with decreasing urgency: (1) pun-

ishment, (2) reward, and (3) novelty. As punishment and rewards are typically sparse in time, novelty can provide temporally dense motivation even during early life. Krichmar 2008 [10] provided a survey that includes five types of neural transmitters. Singh et al, 2010 [19] adopted an evolutionary perspective and define a new reward framework that captures evolutionary success across environments. Niekum et al, 2010 [14] presented a genetic programming algorithm to search for alternate reward functions to improve agent learning performance.

As neuromorphic architecture is more restrictive than one that does not have any restriction, the above models of intrinsic value systems are not neuromorphic.

C. Autonomous development: task non-specificity

A neuromorphic system requires that the computational unit is neuron like. It is well accepted that a neural network has an emergent representation — adaptive synaptic weights and distributed firing patterns. It is also well recognized that local learning is a desirable property for neural networks. Weng et al. 2008 [29] further argued that the *genomic equivalence principle* [17] implies that development and computation are both cell-centered. Each cell is autonomous during development in general and during learning in particular. A consequence of this cell autonomy is that each cell does not have dedicated learner for its own learning — it is fully responsible for the learning all by itself in its environment. In other words, each neuron must use its intrinsic properties and the environmental conditions to accomplish its learning. In a larger scope, Weng et al. 2001 [30] proposed that autonomous mental development should be task-nonspecific. If the task is not known, progress maximization and similarity-based progress maximization seem ill defined.

The above conditions for a neuromorphic system does not mean to make learning less powerful or more difficult, but rather they enable the autonomous developmental system to learn higher values that are not restricted to a small domain.

Table I conceptually compares agents along two conceptual axes: motivated and neuromorphic. The above models belong to the category of symbolic motivated agents.

TABLE I
ARCHITECTURAL CONCEPTS

	Non-neuromorphic	Neuromorphic
Not motivated	Symbolic agents	Neural networks
Motivated	Symbolic motivated agents	Neuromorphic motivated agents

D. Neuromorphic value systems: Challenges

Almassy et al. 1998 [1], further refined in Sporns et al. 2000 [22], proposed a neuromorphic architecture for learning primary and secondary conditioning that tend to avoid actions that lead to punishments and adopt actions that lead to reward. Cox & Krichmar 2007 [2] experimented with a neuromorphic

architecture that integrates three types of neurotransmitters, 5-HT, DA and Ach with Ach for increased attention efforts.

Dealing with time in frame precision, that is at the rate that the simulation updates, is an unsolved problem. For example, in their Darwin V simulation [1], [22], behaviors are hand-crafted modes that span many frames, such as obstacle avoidance, approaching and avoiding. There have been no neuromorphic systems that deal with both value and time with the same precision as the network update rate. For example, a behavior cannot be terminated until the behavior execution is finished. The recent model of DN made this possible. The novelty of this work lies in a new architecture for an intrinsic value system with a neuromorphic system so that both deal with time at the frame precision. In this way, only the primitive actions are defined innately, each spanning a single time frame only. Longer-term behaviors are all emergent, numerous, and unbounded in the total number even with a limited network memory. This is because the complexity of the longer-term behaviors also depend on the complexity of the environment. For example, an approaching action can be altered at any time frame if an aversive agent came into the way.

In a neuromorphic system, how the neuromorphic value system interacts with the neuromorphic sensorimotor system is also unknown. The Darwin V simulation [1], [22] uses appetitive and aversive stimuli to directly link the corresponding appetitive and aversive behaviors, respectively. Many symbolic methods associate each symbolic long-term behavior with a value, so that a value-based selection mechanism arbitrates which symbolic long-term behavior is executed [26]. Therefore, the value system is like an approval system. Such an approval idea ran into problems with neuromorphic systems. For example, Merrick 2011 [13] proposed an network architecture in which the value system acts like an approval system for sensory inputs, like symbolic systems. Thus, her architecture requires that each neuron in the motivation layer to pass the synaptic weights (not a single response value) to the succeeding neurons. There seems no evidence yet that a neuron can transmit its synaptic weights. Therefore, we do not classify Merrick’s model as neuromorphic.

E. A new neuromorphic value architecture

Our motivated neuromorphic architecture is based on a recent network called the Developmental Network (DN), which deals with both space and time in an integrated way — temporal context is recursively “folded” into the spatial area of a finite automaton (FA) so that current motor state (response pattern) represents all the temporal context attended at the current time and they are all treated equivalent. Furthermore, all future processing is based on such an equivalence.

Based on the DN framework, we propose a new neuromorphic architecture with an neuromorphic intrinsic value system. There is no need to directly link an aversive stimuli directly to an avoidance behavior since our primitive actions are all short (single-frame time), shared by aversive and appetitive behaviors. The architecture does not require synaptic weights to be transmitted to succeeding neurons. In the new architecture,

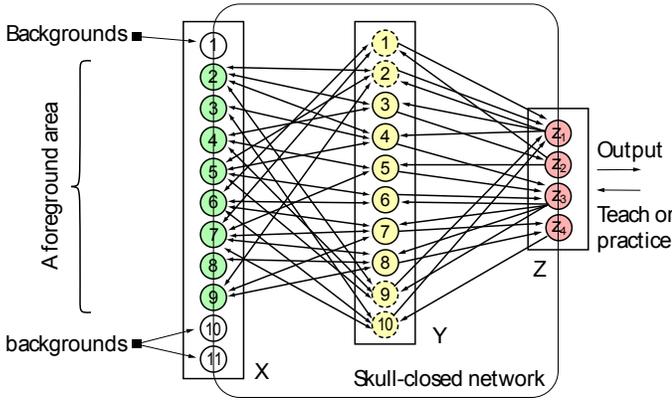


Fig. 1. The architecture of DA. It contains top-down connections from Z to Y for context represented by the motor area. It contains top-down connections from Y to X for sensory prediction (but this part is not used in the work here). Pink areas are human designed or human taught. Yellow areas are autonomously generated (emergent and developed).

the neuromorphic motivation system develops in parallel with the basic neuromorphic sensorimotor system. The architecture enables the two systems to interact in the motor area, via a simple *collateral triplets* which is supposed to be hardwired for each primary motor neuron. Each collateral triplet is a combination of the unbiased, appetitive, and aversive states. In this way, each primary motor neuron can be excited by an appetitive stimulus, inhibited by an aversive stimulus, or both concurrently.

Experimentally, we show how adding serotonin and dopamine-like punishment and reward sensations can be used to guide a robot into learning to make the right decisions. Our robot is initially placed within a simulated room with two other robots, an “attractor” robot which proximity to triggers the release of dopamine in our robot, and a “repulsor” robot with whom close proximity triggers the release of serotonin. Our robot is rewarded for moving towards the attractor and away from the repulsor and punished for doing the reverse.

The rest of the paper is organized as follows. In Section II, we review the rest of the theory behind our model. In Section III, we show the results of our experiments. Finally, in Section IV we present our conclusions and closing remarks.

II. THEORY

We will first introduce the basic framework of the network, before we discuss the theory about value-based networks.

A. Developmental Network

Developmental Networks (DNs) are a new class of recurrent neural network for spatiotemporal processing. The most basic version of DN has three areas (layers), the sensory area X , the internal area Y and the motor area Z , with an example in Fig. 1. The internal neurons in Y have connections with both the sensory end X and the motor end Z .

The Developmental Program (DP) for DN is not task-specific as suggested for the brain in [30] (e.g., not concept-specific) at the birth time. The environmental concepts are

learned incrementally through interactions with the environments. In principle, the X area can model any sensory modality (e.g., vision, auditory and touch). The motor area Z serves as both input and output ports. When the environment supervises Z , Z is the input to the network. Otherwise, Z gives an output vector to drive effectors (muscles) to act on the real world. The order from low to high is: X, Y, Z . The developmental learning of DN, regulated by its DP, is desirably very mechanical:

Algorithm of DN:

- 1) At time $t = 0$, for each area A in $\{X, Y, Z\}$, initialize its adaptive part $N = (V, G)$ and the response vector \mathbf{r} , where V contains all the synaptic weight vectors and G stores all the neuron ages.
- 2) At time $t = 1, 2, \dots$, for each area A in $\{X, Y, Z\}$, do the following two steps repeatedly forever:
 - a) Every area A computes using the response function f .

$$(\mathbf{r}', N') = f(\mathbf{b}, \mathbf{t}, N) \quad (1)$$

where f is the unified response function described below; \mathbf{b} and \mathbf{t} are area’s bottom-up and top-down inputs, respectively; and \mathbf{r}' is its response vector.

- b) For each area A in $\{X, Y, Z\}$, A replaces: $N \leftarrow N'$ and $\mathbf{r} \leftarrow \mathbf{r}'$.

If X is a sensory area, $\mathbf{x} \in X$ is always supervised and then it does not need a synaptic vector. The $\mathbf{z} \in Z$ is supervised only when the teacher chooses to. Otherwise, \mathbf{z} gives motor output.

Next, we describe the response function f . Each neuron in area A has a weight vector $\mathbf{v} = (\mathbf{v}_b, \mathbf{v}_t)$. Its pre-action potential is the sum of two normalized inner products:

$$r(\mathbf{v}_b, \mathbf{b}, \mathbf{v}_t, \mathbf{t}) = \frac{\mathbf{v}_b \cdot \mathbf{b}}{\|\mathbf{v}_b\| \cdot \|\mathbf{b}\|} + \frac{\mathbf{v}_t \cdot \mathbf{t}}{\|\mathbf{v}_t\| \cdot \|\mathbf{t}\|} = \dot{\mathbf{v}} \cdot \dot{\mathbf{p}} \quad (2)$$

which measures the degree of similarity between the directions of $\dot{\mathbf{v}} = (\mathbf{v}_b/\|\mathbf{v}_b\|, \mathbf{v}_t/\|\mathbf{v}_t\|)$ and $\dot{\mathbf{p}} = (\mathbf{b}/\|\mathbf{b}\|, \mathbf{t}/\|\mathbf{t}\|)$.

To simulate lateral inhibitions (winner-take-all behavior) within each area A , only the top k winners fire. Considering $k = 1$, the winner neuron j is identified by:

$$j = \arg \max_{1 \leq i \leq c} r(\mathbf{v}_{bi}, \mathbf{b}, \mathbf{v}_{ti}, \mathbf{t}). \quad (3)$$

The area dynamically scales the top- k winners so that the top- k responses with values in $[0, 1]$. For $k = 1$, only the single winner fires with response value $y_j = 1$ and all other neurons in A do not fire. The response value y_j approximates the probability for $\dot{\mathbf{p}}$ to fall into the Voronoi region of its $\dot{\mathbf{v}}_j$ where the “nearness” is $r(\mathbf{v}_b, \mathbf{b}, \mathbf{v}_t, \mathbf{t})$.

All the connections in a DN are learned incrementally based on Hebbian learning — cofiring of the pre-synaptic activity $\dot{\mathbf{p}}$ and the post-synaptic activity y of the firing neuron.

The focus of this paper is about how to enable a network to have motivation. It is beyond the scope of this paper to discuss a series of properties of the DN. The reader is referred to Weng 2009 [27] for a series of DN properties and references to a series of experimental studies of DN.

B. Neural modulation for intrinsic motivation

A motivational system goes beyond information processing and sensorimotor behaviors. It provides mechanisms to a developmental system so that it develops its likes and dislikes. Without a motivational system, it is difficult to enable a system to autonomously learn and perform desirable tasks.

Neural modulation addresses how a few particular types of neural transmitters are used by the central nervous system to regulate the development and operations of its circuits in general, and intrinsic motivation in particular. The material in the previous section deals with signal processing in high spatial and temporal resolutions. Such processing is characterized by direct synaptic transmission — the presynaptic neuron directly influences the post-synaptic neuron. In contrast, in the subject of neural modulation, a small group of neurons specialized for a particular type of neural modulatory transmitters secrete such transmitters which diffuse through large areas of the nervous system, producing an effect on multiple neurons.

Functionally, neural modulation is needed for non-associative learning (e.g., sensitization and habituation), classical conditioning, instrumental conditioning (also called reinforcement learning), and many other types of autonomous learning. Furthermore, neural modulation is also necessary for the agent to have an effective rejection option (do not know). However, neural modulation must rely on the developed circuits as it only modulates the working of such circuits.

In this work, we focus on two types of neuronal transmitter systems, Serotonin and Dopamine. To model 5-HT and DA systems, our model needs two more types of neurons:

Dopaminergic neurons are those neurons that are sensitive to dopamine. Firing of these neurons indicates pleasure. The substantia nigra and the ventral tegmental area (VTA) release dopamine (DA).[10]

Serotonergic neurons are those neurons that are sensitive to serotonin. Firing of these neurons indicate stress. Serotonin in the central nervous system originates in the raphe nuclei of the brainstem.[10]

The serotonin and dopamine systems appear to act on motor neurons. When a motor neuron receives dopamine, it is more likely to fire. When a motor neuron receives serotonin, it is less likely to fire.

Therefore, during training, motor neurons that correspond to doing some dangerous action or leaving some pleasurable activity should receive serotonin; motor neurons that correspond to performing a pleasurable activity or evading a dangerous one should receive dopamine. Together, these two systems encourage the agent to prefer actions that experience has shown to bring pleasure over those that tend to cause pain. Fig. 2 presents the architecture of DN after it is augmented with a motivational system.

C. Models with 5-HT and DA

We define the concept of (congenitally) unbiased and biased for a cortical area. By unbiased, we mean that the area is not affected by the motivational system at the birth time. Its bias is acquired largely through postnatal experience. This concept

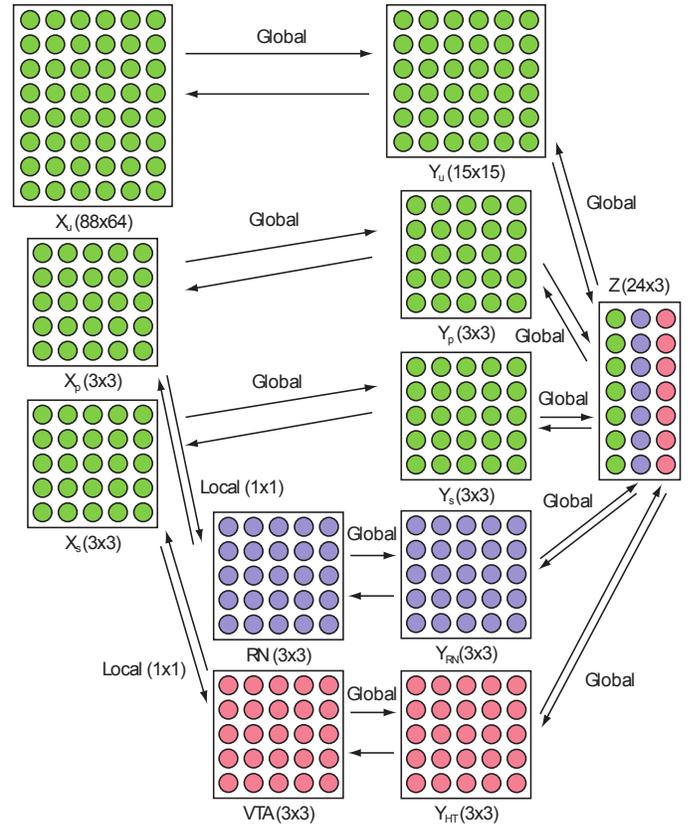


Fig. 2. A DN with a motivational system represented by 5-HT and DA. The blue color denotes serotonergic neurons. The pink color denotes dopaminergic neurons. In the motor area, each row of neurons corresponds to a collateral triplet.

is only roughly true in biology as hardly any cortical area is not affected by various neural modulatory systems at birth time. By biased, we mean that the area is affected by the motivational system at the birth time.

As illustrated in Fig. 2, the architecture links all pain receptor with raphe nuclei (RN) located in the brain stem — represented as an area which has the same number of neurons as the number of pain sensors. Every neuron in RN releases serotonin.

The architecture also links all sweet receptors, which are sensitive to dopamine, with VTA— represented as an area, which has the same number of neurons as the number of sweet receptors. Every neuron in the VTA releases dopamine.

Therefore, the sensory area $X = (X_u, X_p, X_s)$ consisting of an unbiased array X_u , a pain array X_p , a sweet array X_s .

$Y = (Y_u, Y_p, Y_s, Y_{RN}, Y_{HT})$ connects with $X = (X_u, X_p, X_s)$, RN and HT as bottom-up inputs and Z as top-down input.

In a motivated neural network, the motor area is denoted as $Z = (\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_m)$, where m is the number of muxels. Each \mathbf{z}_i has three neurons $\mathbf{z}_i = (z_{iu}, z_{ip}, z_{is})$, where z_{iu}, z_{ip}, z_{is} are unbiased, pain, and sweet, respectively, $i = 1, 2, \dots, m$. z_{ip} and z_{is} are serotonin and dopamine collaterals, associated with z_{iu} , as illustrated by the Z area in Fig. 2.

Whether the action i is released depends on not only the response of z_{iu} but also those of z_{ip} and z_{is} . z_{ip} and z_{is} report how much negative value and positive value are associated with the i -th action. We use the following collateral rule:

Definition 1 (Collateral Rule): Each motivated action is a vector $\mathbf{z}_i = (z_{iu}, z_{ip}, z_{is})$ in $Z = (\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_m)$, $i = 1, 2, \dots, m$. The response of the action neuron is determined by

$$z_{iu} \leftarrow \max\{z_{iu}(1 + z_{is} - \alpha z_{ip}), 0\} \quad (4)$$

with a very large constant α .

In other words, if $z_{ip} > 0$, the corresponding action neuron mostly likely does not fire, as pain is the most dominant factor to avoid. Otherwise, z_{is} boosts the pre-action potential for the i -th action to be released.

D. Operation

As an example of a motivational system, let us discuss how such a motivated network realizes *instrumental conditioning*, a well known animal model of reinforcement learning discussed in psychology [4].

We only first consider a one-step delay in the following model here. We use an arrow (\rightarrow) to denote causal change over time.

Suppose that an action \mathbf{z}_a leads to pain but another action \mathbf{z}_b leads to sweet. Using our notation, we have

$$\begin{aligned} (\mathbf{x}(t_1), \mathbf{z}(t_1)) &= (\mathbf{x}_u(t_1), \mathbf{o}(t_1), \mathbf{o}(t_1), \mathbf{z}_a(t_1)) \\ \rightarrow &((\mathbf{x}_u(t_1 + 1), \mathbf{x}_p(t_1 + 1), \mathbf{o}(t_1 + 1), \mathbf{z}_{a,p}(t_1 + 1))) \end{aligned}$$

and

$$\begin{aligned} (\mathbf{x}(t_2), \mathbf{z}(t_2)) &= (\mathbf{x}_u(t_2), \mathbf{o}(t_2), \mathbf{o}(t_2), \mathbf{z}_b(t_2)) \\ \rightarrow &((\mathbf{x}_u(t_2 + 1), \mathbf{o}(t_2 + 1), \mathbf{x}_s(t_2 + 1), \mathbf{z}_{b,s}(t_2 + 1))) \end{aligned}$$

where p and s indicates pain and sweet, respectively and \mathbf{o} denotes a zero vector. The vectors \mathbf{z}_a and \mathbf{z}_b represent two different actions with different i 's for Z_i . In this work, we use a different neuron in Z to represent a different action, although this is only for simplicity.

In our example above, let $\mathbf{z}_a = (1, 0, 0)$ and $\mathbf{z}_b = (1, 0, 0)$, but $\mathbf{z}_{a,p} = (1, 1, 0)$ and $\mathbf{z}_{b,s} = (1, 0, 1)$.

Next, the agent runs into a similar scenario $\mathbf{x}'_u \approx \mathbf{x}_u$. $\mathbf{x}' = (\mathbf{x}'_u, \mathbf{o}, \mathbf{o})$ is matched by the same vector \mathbf{y} as $\mathbf{x} = (\mathbf{x}_u, \mathbf{o}, \mathbf{o})$. Through the \mathbf{y} vector response in Y , the motor area comes up with two actions $\mathbf{z}_{a,p} = (1, 1, 0)$ and $\mathbf{z}_{b,s} = (1, 0, 1)$. Using our collateral rule, \mathbf{z}_a is suppressed and \mathbf{z}_b is executed.

We note that the above discussion only spans one unit time of the network update. However, the network can continue to predict:

$$(\mathbf{x}(t), \mathbf{z}(t)) \rightarrow (\mathbf{x}(t+1), \mathbf{z}(t+1)) \rightarrow (\mathbf{x}(t+2), \mathbf{z}(t+2)) \quad (5)$$

and so on. This seems a biologically plausible way of dealing with delayed reward. This way is different from Q-learning [26] which is symbolic in the sense that every node in the Q-learning graph is atomic, not a pattern of response vector.

E. Experimental procedures

A protocol for learning and testing is as follows:

- 1) Learn pains: $(\mathbf{x}_u, \mathbf{o}, \mathbf{o}, \mathbf{z}_a) \rightarrow (\mathbf{x}_u, \mathbf{x}_p, \mathbf{o}, \mathbf{z}_{u,p})$
- 2) Learn sweets: $(\mathbf{x}_u, \mathbf{o}, \mathbf{o}, \mathbf{z}_b) \rightarrow (\mathbf{x}_u, \mathbf{o}, \mathbf{x}_s, \mathbf{z}_{u,s})$
- 3) Test pain avoidance and pleasure seeking:

$$(\mathbf{x}_u, \mathbf{o}, \mathbf{o}, \mathbf{o}) \rightarrow (\mathbf{x}_u, \mathbf{o}, \mathbf{o}, \mathbf{z}_{a \cup b}) \rightarrow (\mathbf{x}_u, \mathbf{x}_s, \mathbf{o}, \mathbf{z}_{b,s}) \quad (6)$$

where $\mathbf{z}_{a \cup b} \in Z$ denotes a response vector where both \mathbf{z}_a and \mathbf{z}_b are certain but with different collaterals.

III. EXPERIMENTS

Here, we describe the experiments we have conducted that implement and test the above theory and algorithms.

A. Experimental design

In our experiments, there are three robots in a simulation. Two of them are denoted as an attractor and a repulsor while a third is the agent that is able to think and act. If the agent approaches the attractor, it is rewarded with dopamine, but if it approaches the repulsor it is punished with serotonin. In this way, the agent will learn to approach the attractor and evade the repulsor. However, it learns this through its own trial-and-error; the agent must take the actions of its own volition rather than having some imposed on it. It learns which ones lead to which results only through its own experience.

The agent ‘‘brain’’ is a DN with three areas, X , Y , and Z , where X is the lower sensor area, Z is the upper motor area, and Y is the middle internal area, as illustrated in Fig. 2. The X area has three sub-areas, X_u , the unbiased area taken from its sensors, X_p , the pain area, and X_r , the sweet area. At each time step, each area produces a response vector based on the physical state of the world. The \mathbf{x}_u vector is created directly from the physical state of the world. The \mathbf{x}_p vector identifies in which ways the robot is being punished and represents the release of serotonin in RN. The last vector, \mathbf{x}_r , identifies in which ways the robot is being rewarded and represents the release of dopamine from VTA.

All of the Y and Z sub-areas compute their response vectors in the same way. The input into the area is the array \mathbf{u} of vectors. Each neuron in the area maintains its current state \mathbf{p} which has the same dimensions as \mathbf{u} . The Z area then recombines the results from its collaterals to compute a single response vector.

At the end of each time step, the neurons in the Y and Z areas that fired update themselves. This is done according to the following series of equations, based on the neurons current state, \mathbf{p}_i , age, a_i , the response vector \mathbf{r} , and the input vector \mathbf{u}

$$\beta = 1/a_i \quad (7)$$

$$\alpha = 1 - \beta \quad (8)$$

$$\mathbf{p}_i = \alpha \mathbf{p}_i + \beta r_i \mathbf{u} \quad (9)$$

$$a_i = a_i + r_i \quad (10)$$

For the Y area, all of the top k neurons update, but none of the others do, simulating the Hebian learning of the area. See Weng & Luciw 2009 [28] for the optimal theory behind such updates. Z is updated the same way, except that its response is affected by the collaterals.

There is no special “training state”. All areas update and learn after every timestep.

B. Input and output

The Y and Z_u areas are initialized to contain small random data in their state vectors. The Z_p and Z_r areas are initialized to zero vectors since the robot initially has no idea which actions will cause it weal or woe. The ages of all neurons are initialized to 1.

The above representation is independent of the task at hand. The number of neurons c in the Y area and the number of them that may fire, k , can be selected based on the resources available.

The size of the Z area is equal to the number of actions that can be taken by the robot. In our implementation, there are nine possible actions; it can move in each of the cardinal or intercardinal directions or it can maintain its current position.

The size of each of the vectors in the X area is determined by the transformation function through which the robot senses the world. Our robot can sense the location of the other two entities. It’s transformation function works as follows, given the three entities, S (self), A (attractor), and R (repulsor).

$$\theta_A = \arctan(S_x - A_x, S_y - A_y) \quad (11)$$

$$d_A = \sqrt{(S_x - A_x)^2 + (S_y - A_y)^2} \quad (12)$$

$$\theta_R = \arctan(S_x - R_x, S_y - R_y) \quad (13)$$

$$d_R = \sqrt{(S_x - R_x)^2 + (S_y - R_y)^2} \quad (14)$$

$$\mathbf{x}_u = \left\{ \cos \theta_A, \sin \theta_A, \cos \theta_R, \sin \theta_R, \frac{d_A}{d_A + d_R}, \frac{d_R}{d_A + d_R} \right\} \quad (15)$$

These component designs avoid the problem with the angle representation, which is discontinuous at 2π .

The pain sensor input has just two values representing loneliness and fear. The loneliness value is 1 if $d_A > d_{Lonely}$ and 0 otherwise while the fear value is 1 if $d_R < d_{Fear}$. The sweet sensor has only one value which represents desire. It is set to 1 if $d_A < d_{Desire}$.

C. Experimental setup

We created a simulation setting to demonstrate how such a motivated agent would respond in the presence of a friendly entity and an adversarial entity. The world is represented by a coordinate plane of pre-determined size. The friend and adversary move in semi-random directions within this plane, while having a tendency to continue. This results in an ant-like wandering action. Our motivated agent, which we will call the self, is controlled by its motivated “brain”. The “brain” releases dopamine and serotonin for the friend and adversary

based on specified circumstances within its world. These circumstances are a result of user set parameters and random occurrences. Throughout our simulation the self will learn by reinforcement, deciding which one to avoid and which one to embrace based on the release of dopamine and/or serotonin.

Our world view that we feed into the self’s brain consists of an objective view as mentioned above. This unbiased view contains the actual state of the world from the perspective of our self entity. We also include a pain and pleasure view. These views are used to represent the level of danger or pleasure that our self entity is currently engaged in. These views are affected by the values for d_{Desire} , d_{Fear} and $d_{Loneliness}$ as used above and are input parameters for our simulation.

For our experiments we control several parameters to display the experiential learning that takes place in our self entity. We set the starting coordinates of our self entity to the origin. The adversary is placed at (0, 100) and the friend is placed at (100,0). We control the size of the plane, setting it at 500 by 500. Most importantly, we set the fear and loneliness thresholds. In our simulations we set the desire threshold value to the fear threshold value. These thresholds designate how the self entity will feel pain and pleasure within the world. The world simulation lasts 1000 time steps.

At each time step, the horizontal and vertical coordinates are collected for each entity within the world. Using this data, we calculate the Euclidean distance between the entities. By observing the proximity of our self entity to these other entities, we can measure the progress of the self entity’s learning. Part of our simulation uses a GUI to display the movements of our entities around the field in the form of an animation. This allows us to visualize our data more effectively as we analyze the actions of the entities.

The agent starts with a behavior pattern determined by its initial neural network configuration. The unbiased regions are initialized with small random data while the biased regions are initialized to zero. This gives the initial appearance of a random behavior pattern. Eventually, it performs an action that causes it to be rewarded or punished, causing it to either favor or avoid that action when placed in similar situations in the future.

D. Analysis

The following are different instances of the problem by changing the values of the fear, desire, and loneliness thresholds. The results are summarized in Figs. 3, 4, and 5.

1) *No Brainer (Control)*: In this instance our self entity receives no pain or pleasure stimulus. Without any feedback, the entity tends to repeat the same action as incrementally very little change happens to the state relative to its point of view. As such, it tends to wander off into a wall. In this example, the entity touched and wandered past the adversary.

2) *Love Or War (50, 50)*: The loneliness threshold was set to 50 and the fear threshold to 50. The starting state of the self entity feels pain from loneliness; however the self entity has no indication that the adversary can also cause pain. It also does not know that the friend causes pleasure or that the friend will

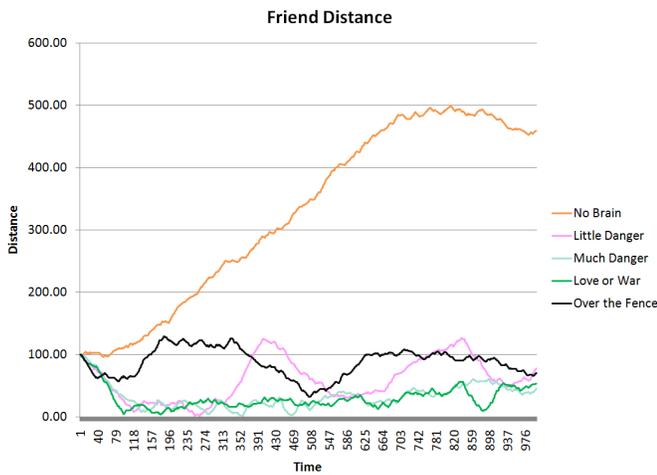


Fig. 3. The distances between the self and friend entities in each simulation.

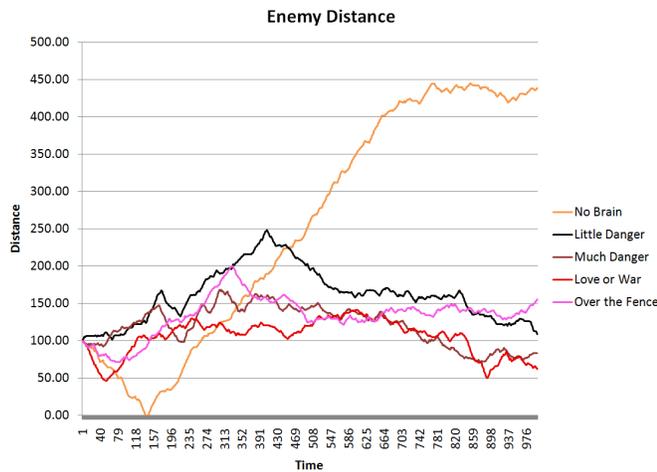


Fig. 4. The distances between the self and the adversary entities in each simulation.

make the loneliness pain go away. When the entities began to move around the world, things change. At around time step 50, our self entity encountered the radius of the adversary and the friend. After this point notice that the self entity gets closer to the friend and farther from the enemy. Throughout its lifetime, the self entity attempts to keep the enemy distance above 50 and the friend distance below 50.

3) *Little Danger (25, 125)*: This scenario initializes our self entity in no danger. Because the self entity is within the loneliness threshold and outside of the fear threshold, our self entity starts out in a neutral state, not feeling pain nor pleasure. But when our self entity moves toward the friend randomly, it realizes that it can receive pleasure as seen between time steps 100 and 300. During this time range, the self entity stays within the desire range. However, the self entity moves away from the friend until it reaches the loneliness threshold, where it returns to the friend because of the pain it feels. Near the end of the self entity’s life, the friend and adversary approach

each other, but the self entity tries to stay close to the friend, while avoiding the adversary.

4) *Much Danger (125, 25)*: The self entity is initialized into a state of peril, where the loneliness threshold is breached and the fear threshold is upon the self entity. Originally, our self entity does not know that it must be near the friend to relieve some of its pain. As a result, it simply runs away from the adversary. Fortunately, the self entity runs into the loneliness radius of the friend. Once here, the entity stays next to the friend, until it is approached by the adversary. Because it wants to stay out of the fear threshold, the entity moves away from the friend. During this entire scenario, the entity is feeling pleasure because it remains within the friend threshold. However, we see that the pain forces dictate its movements.

5) *Looking Over The Fence (125, 125)*: In this interesting case, the self entity is within the loneliness threshold and also within the fear and desire thresholds. Not knowing where the pain or pleasure is coming from, the self entity hangs around both the friend and the adversary. However, when the self entity leaves the loneliness threshold and fear threshold, the self entity realizes that the adversary is causing pain and going away from the friend is causing more pain. From this point on the self entity successfully avoids the adversary and keeps close to the friend. Notice how the friend stays under 125 units away and adversary over 125 units away. In this specific instance, the friend and adversary stay very close to the threshold lines. This can be viewed as a child viewing a tiger in a cage, hence it’s like “Looking over the fence.”

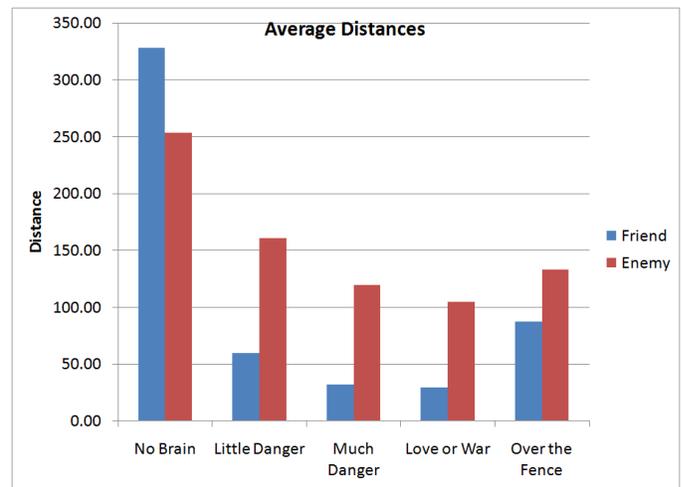


Fig. 5. The average distance between the self and the other entities for each simulation.

As shown in Figure 5, the control sample tends to be a large distance away from the other entities, roughly half the length of the field. This is because the entity receives no reward or punishment feedback and so never learns to move towards or away from the other entities. In the other situations, the distance to the friend entity tends to be much smaller than that to the enemy. In all of these cases, the average distance to the enemy is greater than the fear threshold and the average

distance to the friend is smaller than the loneliness threshold. Additionally, for the “Love or War” and “Looking over the Fence” sets, the average distance is below the desire threshold. This suggests that the reinforcement learning was useful for teaching the robot to stay close to its friend and away from its enemy.

IV. CONCLUSIONS

The new neuromorphic motivated system has the following architecture novelties: (a) Both sensorimotor system and the motivational systems develop in parallel but not hardwired in terms of the value bias. (b) The collateral triplets in the motor area mediate the dynamic excitation and inhibition of every motor primitive, so that each motor primitive can be shared by longer term appetitive behaviors and aversive behaviors. (c) The temporal precision of motivated behaviors is at the sensory frame precision, so that each long-term behavior can be terminated at the frame precision when the environment calls for.

We conclude that the proposed method of simulated dopamine and serotonin is successful. Our robots were able to figure out for themselves how to react in a given situation rather than having to be explicitly taught where to move. The control set verifies that it is indeed the pain and pleasure actions that cause the seen behavior.

In future experiments, the robot could be placed in a more complicated setting with multiple attractors and repulsors. It may also be given a more limited world view than the one used here; our robot was omniscient in that it knew where everything was at all times. Finally, it could learn how to do things like tricks by having a human operator feeding its pain or pleasure sensors when it does something interesting. In this way, it would be able to perform tasks not thought of at design time.

REFERENCES

- [1] N. Almassy, G. M. Edelman, and O. Sporns. Behavioral constraints in the development of neural properties: A cortical model embedded in a real-world device. *Cerebral Cortex*, 8(4):346–361, 1998.
- [2] B. Cox and J. Krichmar. Neuromodulation as a robot controller. *IEEE Robotics and Automations Magazine*, 16(3):72 – 80, 2009.
- [3] E. Deci and R. Ryan. *Intrinsic motivation and self-determination in human behaviour*. Plenum Press, New York, 1985.
- [4] M. Domjan. *The Principles of Learning and Behavior*. Brooks/Cole, Belmont, California, fourth edition, 1998.
- [5] S. F. Gilbert. *Developmental Biology*. Sinauer, Sunderland, Massachusetts, 8 edition, 2006.
- [6] X. Huang and J. Weng. Inherent value systems for autonomous mental development. *International Journal of Humanoid Robotics*, 4(2):407–433, 2007.
- [7] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.
- [8] S. Kakade and P. Dayan. Dopamine: generalization and bonuses. *Neural Network*, 15:549559, 2002.
- [9] E. R. Kandel, J. H. Schwartz, and T. M. Jessell, editors. *Principles of Neural Science*. McGraw-Hill, New York, 4th edition, 2000.
- [10] J. L. Krichmar. The neuromodulatory system: A framework for survival and adaptive behavior in a challenging world. *Adaptive Behavior*, 16(6):385–399, 2008.
- [11] A. H. Maslow. A theory of human motivation. *Psychological Review*, 50(4):370–396, 1943.
- [12] A. H. Maslow. *Motivation and Personality*. Harper and Row, New York, 1 edition, 1954.
- [13] K. E. Merrick. A comparative study of value systems for self-motivated exploration and learning by robots. *IEEE Trans. Autonomous Mental Development*, 2(2):119–131, 2010.
- [14] S. Niekum, A. G. Barto, and L. Spector. Genetic programming for reward function search. *IEEE Trans. Autonomous Mental Development*, 2(2):83–90, 2010.
- [15] H. Ogmen. A developmental perspective to neural models of intelligence and learning. In D. Levine and R. Elsberty, editors, *Optimality in Biological and Artificial Networks*, page 363395. Lawrence Erlbaum, Hillsdale, NJ, 1997.
- [16] P.-Y. Oudeyer, F. Kaplan, and V. Hafner. Intrinsic motivation for autonomous mental development. *IEEE Transactions on Evolutionary Computation*, 11(2):265286, 2007.
- [17] W. K. Purves, D. Sadava, G. H. Orians, and H. C. Heller. *Life: The Science of Biology*. Sinauer, Sunderland, MA, 7 edition, 2004.
- [18] T. W. Robbins and B. J. Everitt. Neurobehavioural mechanisms of reward and motivation. *Current Opinion in Neurobiology*, 6(2):228–236, April 1996.
- [19] S. Singh, R. L. Lewis, A. G. Barto, and J. Sorg. Intrinsically motivated reinforcement learning: An evolutionary perspective. *IEEE Trans. Autonomous Mental Development*, 2(2):70–82, 2010.
- [20] R. L. Solomon and J. D. Corbit. An opponent-process theory of motivation: II. cigarette addiction. *Journal of Abnormal Psychology*, 81:158–171, 1973.
- [21] R. L. Solomon and J. D. Corbit. An opponent-process theory of motivation: I. the temporal dynamics of affect. *Psychological Review*, 81:119–145, 1974.
- [22] O. Sporns, N. Almassy, and G.M. Edelman. Plasticity in value systems and its role in adaptive behavior. *Adaptive Behavior*, 7(3), 1999.
- [23] M. Sur and J. L. R. Rubenstein. Patterning and plasticity of the cerebral cortex. *Science*, 310:805–810, 2005.
- [24] R. S. Sutton and A. Barto. *Reinforcement Learning*. MIT Press, Cambridge, Massachusetts, 1998.
- [25] R.S. Sutton and A.G. Barto. Toward a modern theory of adaptive networks: Expectation and prediction. *Psychological Review*, 88:135–170, 1981.
- [26] C. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8:279–292, 1992.
- [27] J. Weng. A 5-chunk developmental brain-mind network model for multiple events in complex backgrounds. In *Proc. Int’l Joint Conf. Neural Networks*, pages 1–8, Barcelona, Spain, July 18-23 2010.
- [28] J. Weng and M. Luciw. Dually optimal neuronal layers: Lobe component analysis. *IEEE Trans. Autonomous Mental Development*, 1(1):68–85, 2009.
- [29] J. Weng, T. Luwang, H. Lu, and X. Xue. Multilayer in-place learning networks for modeling functional layers in the laminar cortex. *Neural Networks*, 21:150–159, 2008.
- [30] J. Weng, J. McClelland, A. Pentland, O. Sporns, I. Stockman, M. Sur, and E. Thelen. Autonomous mental development by robots and animals. *Science*, 291(5504):599–600, 2001.