4-2012

Motivated learning for the development of autonomous agents

Janusz A. STARZYK

James T. GRAHAM

Pawel RAIF

Ah-hwee TAN
Singapore Management University, ahtan@smu.edu.sg

Citation

STARZYK, Janusz A.; GRAHAM, James T.; RAIF, Pawel; and TAN, Ah-hwee. Motivated learning for the development of autonomous agents. (2012). *Cognitive Systems Research*. 14, (1), 10-25.
Available at: https://ink.library.smu.edu.sg/sis_research/5195

Available online at www.sciencedirect.com

ScienceDirect

Cognitive Systems Research 14 (2012) 10–25

Cognitive Systems RESEARCH

www.elsevier.com/locate/cogsys

Motivated learning for the development of autonomous systems

Action editors: Minho Lee and Wlodzislaw Duch

Janusz A. Starzyk [a], James T. Graham [a,*], Pawel Raif [b], Ah-Hwee Tan [c]

[a] *School of Electrical Engineering and Computer Science at Ohio University, Athens, OH, USA*
[b] *Faculty of Organization and Management, Silesian University of Technology, Gliwice, Poland*
[c] *School of Computer Engineering, Nanyang Technological University, Singapore*

Received 1 April 2010; accepted 12 December 2010
Available online 12 January 2011

Abstract

A new machine learning approach known as motivated learning (ML) is presented in this work. Motivated learning drives a machine to develop abstract motivations and choose its own goals. ML also provides a self-organizing system that controls a machine's behavior based on competition between dynamically-changing pain signals. This provides an interplay of externally driven and internally generated control signals. It is demonstrated that ML not only yields a more sophisticated learning mechanism and system of values than reinforcement learning (RL), but is also more efficient in learning complex relations and delivers better performance than RL in dynamically-changing environments. In addition, this paper shows the basic neural network structures used to create abstract motivations, higher level goals, and subgoals. Finally, simulation results show comparisons between ML and RL in environments of gradually increasing sophistication and levels of difficulty.

Keywords: Autonomous systems; Intelligent agents; Motivated learning; Neural networks; Reinforcement learning

1. Introduction

Intelligent machines are expected to revolutionize the way we live, yet we still do not know how to design or build systems with "true" intelligence. The biological brain is both an inspiration and a model for the development of intelligent machines. We cannot build a brain, but we can try to design models that exhibit similar activation of perceptions, memories and motor control in a given environment. Artificial neural networks (ANN) inspired by networks of biological neurons are successfully used for classification, function approximation and control. Yet a classical ANN learns only a single task, requires extensive training effort, and close supervision.

The reinforcement learning (RL) mechanism is related to the way animals and humans learn (Bakker & Schmidhuber, 2004). Based only on occasional reward and punishment signals, RL agents must learn how to interact with their environment to maximize their expected reward. However, the learning effort and computational cost increase significantly with the environmental complexity (Barto & Mahadevan, 2003), thus, optimal decision making in a complex environment is still intractable using RL. This feature, usually called "the curse of dimensionality", is one of the main disadvantages of RL in real-world applications.

Reinforcement learning also suffers from what is called the "credit assignment problem" (Sutton, 1984; Fu & Anderson, 2006). Reinforcement learning uses a temporal difference mechanism to spread the value of the reward received to earlier stages. However, it does not have a natural mechanism to stop the spread of the reward to yet earlier stages that had nothing to do with receiving the reward. O'Reilly proposed a new primary value and

* Corresponding author.
E-mail addresses: starzykj@gmail.com (J.A. Starzyk), jg193404@ohio.edu (J.T. Graham), pawel.raif@polsl.pl (P. Raif), asahtan@ntu.edu.sg (A.-H. Tan).

1389-0417/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.cogsys.2010.12.009

learned value (PVLV) scheme that implements Pavlovian conditioning (O'Reilly, Hazy, Watz, & Frank, 2007). PVLV directly associates the stimuli and the reward and is a promising alternative to the temporal-differences (TD) used in traditional RL (O'Reilly & Frank, 2006).

One way to improve the efficiency of RL is to use subgoals to build a hierarchy of subsequent goals. The hierarchical reinforcement learning (HRL) approach tends to exploit the structure of both the environment and the agent's tasks to improve policy learning in large scale problems. Among the many approaches to hierarchical RL one can distinguish: Dayan and Hinton's research on feudal reinforcement learning (Dayan & Hinton, 1993), the study by Parr and Russell (1998) on hierarchical abstract machines (HAM) and development of MAXQ Method (Dietterich, 2000).

Bakker and Schmidhuber (2004) proposed a method for hierarchical reinforcement learning based on subgoal discovery and subpolicy specialization. Their HASSLE algorithm can outperform plain RL "by learning to create both useful subgoals and the corresponding specialized subtask solvers." In their algorithm they use HASSLE (Harmon & Baird, 1996) on both high and low levels of hierarchy. Among the limitations of this system are the large number of parameters, the lack of strict convergence guarantees and the dependence on identifying reasonable high-level observations.

Subgoals discovered in hierarchical reinforcement learning (HRL), are obtained by clustering input data (Bakker & Schmidhuber, 2004) to arrive at desired and useful results. In HRL, high-level policies are used to discover subgoals and apply them when appropriate to accomplish the goal. This yields automatic learning of the goal hierarchy minimizing the designer's effort. High-level policies optimize the subgoals and manage their real time use. Individual subgoals are managed by low-level policies that learn low-level value functions in the sensory–motor subspaces. However, identification of useful subgoals is not easy and the large number of design parameters limits the usefulness of the HRL method. While HRL with subgoal discovery does improve machine learning, it still suffers from the major limits of RL, since it is focused on maximizing total reward for externally set objectives.

However, what if we ascribe motivations to machines? An intelligent machine must be able to generate and pursue goals on its own, learning what it needs for a given set of assigned tasks, exploring for a reason, developing new motivations and setting its own goals. Existing methods have made some progress in this direction (Bakker & Schmidhuber, 2004; Barto, 2004; Huang & Weng, 2002; Oudeyer, Kaplan, & Hafner, 2007, 2010; Roa, Kruijff, & Jacobsson, 2009; Schmidhuber, 1991)

The key question is how to "motivate" a machine to act and enhance its intellectual abilities, how to improve its learning efficiency, and how to design a mechanism for structural self-organization from which higher level perceptions and skills could evolve through the machine's interaction with its environment (Pfeifer & Bongard, 2006; Steels, 2004)? What can drive an agent to explore the environment and learn the ways to effectively interact with it? Finally, how can a machine be designed that is capable of not only implementing given goals but also creating them and deciding which goals to pursue? How can this be done in a constantly changing environment, and in spite of distractions and unforeseen difficulties?

1.1. Intrinsic motivation and curiosity driven exploration

According to Pfeifer and Bongard (2006), an agent's motivation should emerge from the developmental process. This is observed in humans and has been argued that it is the result of a system that rewards the engagement of activities just above a person's current ability level. Humans seem to have an innate need to ask "Why?" and "How?" in order to understand the world.

Based on the curiosity principle, Oudeyer et al. (2007, 2010) proposed an intelligent adaptive curiosity (IAC) system, which attempted to direct a robot in continuous, noisy, inhomogeneous, environments, allowing for an autonomous self-organization of behavior toward increasingly complex behavioral patterns. It is widely believed that intrinsic motivation is integral to the way humans learn and explore their environment (Cohn, Ghahramani, & Jordan, 1996; Hasenjäger & Ritter, 2002; Schmidhuber, 1991; Schultz, 2002; White & September, 1959). Oudeyer discusses the benefits children gain by exploring their environment and some of the reasoning behind such behavior (Oudeyer et al., 2007, 2007). Development in children is considered to be autonomous and active, and while adults can provide assistance, it is only assistance. The children's decisions are (largely) their own. The fact that children like to play, and that they actively choose to play for the sake of play, rather than as a step toward solving practical problems, can be taken as proof of the existence of a kind of intrinsic motivation system.

Roa et al. (2009) explored the concept of curiosity and whether it can be emulated through a combination of active learning and RL using intrinsic and extrinsic rewards. The authors developed their intrinsic motivation system based on Oudeyer's work (Oudeyer et al., 2007), and then added an extrinsic reward system to guide the robot to its goal.

By using a learning mechanism based on intrinsic motivations, a machine can explore the environment and learn a hierarchy of skills that it will need to work in this environment (Barto, 2004). Intrinsic motivation can be based on surprise, novelty (Huang & Weng, 2002), or a learning progression as discussed by Kaplan and Oudeyer (2004).

Intrinsic motivation as used in curiosity based learning is similar to exploration in reinforcement learning. In RL a machine does not always respond in an optimum way but occasionally tries a random search in state-action space. However, without proper oversight of curiosity based learning, a machine may not progress efficiently or may even behave in a destructive way (Oudeyer et al., 2007). Intrinsic motivations lead to the selection of actions that yield the maximum rate of reduction of prediction error. However, switching between tasks may slow down learning in spite of the maximum rate of error reduction.

Weng (2004) argues that machine learning methods that focus on the performance of a system in specific tasks are not suited for developmental robots and systems with intrinsic motivations are better suited. While this is a true statement, it is insufficient to consider curiosity based learning as the sole intrinsic motivation. The strength of curiosity based learning may become its weakness once a machine needs to perform specific tasks and needs to specialize. In complex systems with multi-goal operation, machine motivations and the selection of goals need to be internally managed. Thus, there is a need to combine curiosity based learning with motivated learning.

The question is what kind of intrinsic motivations can be used to provide lifetime, task-independent learning to stimulate actions and development? There is a dilemma between developing task-independent cumulative knowledge, and task driven motivations and exploration.

We need a mechanism that will motivate the machine to improve its understanding of the environment while exploring it and still working to accomplish externally set objectives. This mechanism must be different from aimless curiosity based learning, and different from relentless pursuit and exclusive focus on externally set objectives (typical for RL where the machine is rewarded only for such actions).

We propose the motivated learning (ML) approach as an extension of reinforcement learning to include different kinds of intrinsic motivations. Such motivations may be driven by curiosity, as well as external or internal pain signals. The ML agent receives reinforcement from the environment for its most primitive objectives and uses them to develop internal motivations and a complex system of goals and related values. Once established, internal motivations are responsible for more advanced stages of development, learning new concepts, new skills and providing internal rewards attached to successful actions that satisfy them. ML produces value systems related to many abstract concepts in the environment and relates them to its objectives, without receiving an explicit reward for this learning. It resembles human learning, with internal, goal-guided motivations that may lead to the creation of advanced knowledge and intelligent behavior.

In a survey of artificial cognitive systems,Vernon, Metta, and Sandini (2007) addressed various paradigms of cognition, addressing cognitive (symbol based) approaches, emergent system approaches (consisting of connectionist, dynamical, and enactive), and their hybrid combinations. The emergent approach is the one by which a system becomes gradually viable and effective in its environment. Our approach belongs to this category.

1.2. Paper organization

In this paper we describe a motivated learning scheme, which derives a machine's motivations and creates its goals from external pains. We treat the term "pain" as a synonym for all discomforts, fear, panic, anger and pressures.

First, in Section 2 we discuss the role of pain in motivating a machine to learn and to develop its abilities. We indicate how this brings about advanced perception and results in the emergence of desired skills. Then, we discuss how a machine can define higher level goals and then learn to build representations for sensory–motor interactions to support these goals. We illustrate the development process which is responsible for increasing the complexity of the machine's actions as well as managing its goal selection and continuous operation.

In Section 3 we present an example structural organization of neural networks that implement the ML ideas through self-organization. We discuss learning of desired response to selected motivation, discuss curiosity based exploration, and implementation of subgoals. We also describe major differences between ML and RL methods.

Finally, in Section 4 the efficiency of motivated learning is compared to reinforcement learning. This paper concludes with a summary of the proposed approach to revamp the idea of motivated learning, stressing the necessary extensions to the existing approaches.

2. Motivated learning method

An important question is how to create a mechanism that will implement many functions related to sensing, reward processing to trigger learning, managing competing motivations and creating new ones, as well as controlling a machine's actions? An intelligent machine must be able to learn ways to reduce external pain signals received from the environment, and extend this learning to internally generated abstract pains. The external pain signals will be predefined and connected to pain detection centers that trigger the learning mechanism when such pains are either increasing (bad) or decreasing (good). Thus, the machine's motivation comes from its response to external pain signals. The machine will learn how to minimize these pains, and by doing so, will learn the rules and laws of the environment.

Perpetual hostility of the environment may become a foundation for motivations for learning and acting, goal creation, planning, thinking, and problem solving. Knowledge is a by-product of learning. Thus, it is not necessary to include a pre-existing knowledge base in the machine memory. However, some pre-existing knowledge (such as breathing or the sucking reflex of humans or the twitching action babies go through to help develop motor control) is likely to accelerate learning in an intelligent machine.

Intelligence cannot develop without embodiment or interaction with the environment (Pfeifer & Scheier, 1999). Through embodiment, intelligent agents carry out motor actions and affect the environment. The response of the environment (including the pain signals) is registered through sensors implanted in the embodiment. At the same time, the embodiment is a part of the environment that can be perceived, modeled and learned by the intelligent machine, leading to self-determination.

2.1. Role of pain in learning

Although we can use both reward and punishment signals to learn, avoiding punishment may be sufficient for an agent's development (at least in simpler systems) and unlike reward maximization will lead to stable systems. There are many examples of instability in reward based systems. For instance, it was observed that rats would electrically stimulate their reward centers, in preference to eating, until they die (Baars & Gage, 2007). It is also well known that drug abuse in humans (of those drugs that stimulate pleasure centers) may lead to death. While we can always interpret the reduction of pain as a reward, maximization of total reward leads to different solutions than minimization of the dominant pain (negative signal). The first leads to a classical maximization problem and may produce unstable systems (with infinite reward); the second one will be terminated once the negative pain signal is reduced below a specified threshold. In addition, in multi-objective systems, the pain reduction mechanism provides a natural way to manage motivations and goal selection. Mathematically it corresponds to solving a minimax problem, where the optimization effort is concentrated on the strongest pain signal and automatically switches to another objective once the dominant pain is reduced below other pain values. This approach is capable of managing many goals simultaneously.

Pain, as a synonym for all discomforts, fear, panic, anger and pressures, is a common experience of all people. Neurobiological study confirms that there are multiple regions of the brain involved in the pain system also called the "pain matrix" (Derbyshire et al., 1997; Hsieh et al., 2001; Melzack, 1990; Peyron, Laurent, & Garcia-Larrea, 2000; Porro et al., 2002; Tölle et al., 1999). It has been widely accepted that pain has sensory-discriminative, affective, motivational, and evaluative components (Melzack & Casey, 1968; Mesulam, 1990). In this paper, a pain network that is responsible for the goal creation process and affects motivation, attention and sensory perception, is proposed.

2.2. Motivated learning – overall model

In the proposed motivated learning approach, the machine uses neuronal structures to self-organize the motivation and goal creation (GC) system. The system stimulates motivation and the creation of goals not only on the level of externally set motivations (to avoid external pain) but also on various abstraction levels developed by the machine in the learning process. GC is responsible for evaluating actions in relation to setting goals, stimulating learning of useful associations and representations for sensory inputs and motor outputs.

In the proposed motivated learning approach, internal reinforcement signals are used by the machine to make the learning of goals more efficient. Since internal rewards depend on accomplishing goals set internally by the machine, learning is organized without reinforcement input from the teacher. Once the machine learns how to accomplish lower level goals, it develops a need for the sensory inputs required to perform a beneficial action, and this need is used to define higher level motivations and goals. Thus, the agent uses an integrated system of motivations and goals, derived from the primitive motivations (pains) and external rewards, to choose and evaluate its actions. We define motivated learning as follows:

Definition 1. Motivated learning (ML) is pain based motivation, abstract goal creation and learning in an embodied agent.

- ML uses explicitly defined primitive pain signals.
- The machine is rewarded for minimizing the primitive pain signals.
- ML creates abstract motivations and chooses goals based on the primitive pain signals.
- The machine receives internal rewards for satisfying its goals (both primitive and abstract).
- ML applies to embodied agents working in a hostile environment.

Motivated learning needs a mechanism for the creation of abstract motivations and related goals that satisfy these motivations, as well as a mechanism to select goals and supervise their execution. A ML machine is in a continuous process of building new motivations and responding to established ones. Competing signals that represent abstract pains direct the machine to choose a goal to act on and to follow this goal. These signals vary as the machine acts and the environment around it changes. In searching for new solutions, the machine may use RL and curiosity based learning, thus benefitting from their strengths.

The mechanism to build motivations and choose goals enhances perception by triggering the learning of new concepts that were useful for the machine's operation and helps to build internal representations of these objects. In addition, it establishes associations between sensory perceptions and appropriate motor actions. This learning defines new categories useful for the machine's operation. Such useful object categories are learned better and faster than other observed objects discovered through a curiosity based search or frequent observations of these objects in the environment. Effective use of this type of concept learning requires a mechanism for episodic memory. Detailed discussion of the computational model for episodic memory is beyond the scope of this work.

The motivated learning mechanism yields various abstract pain centers responsible for pain evaluation and learning, based on the changing pain signals. Pain signals compete for machine's attention, and the winning signal motivates the machine to act. The machine's goal is to reduce the dominating pain signal.

2.3. Pain-based goal creation

Primitive pain signals are externally defined and generated. In sophisticated environments there are rules that govern relationships between various objects that affect the machine's perception and, in particular, its pain signals. By discovering these rules and learning how to use them to its advantage the machine develops complex knowledge about the environment. A motivated learning system relates its goals to desire for creating conditions in the environment under which solutions of its goals are possible. Thus, the machine learns how to actively change the environment to its own advantage, rather than just responding to an existing state of the environment.

The machine's development is driven by a simple built-in pain based mechanism. The primitive pain (equivalent to a negative reward signal) comes from the hostile environment, and forces the machine to respond. A primitive pain leads to a primitive goal and its satisfaction through proper action triggers the development of higher level pain centers and creates higher level motivations.

The proposed motivated learning mechanism uses basic pain detection and learning units shown in Fig. 1. In this figure sensory and motor neuron activities, for simplicity, are symbolically represented by single neurons (S and M), although, distributed representations of sensory objects or motor actions are more effective and can be used in this method. We use a similar simplification to describe the neural network organization in Section 3.

The pain detection center responds to the input pain signal and represents the negative stimulus that the machine needs to minimize. If the pain exists due to the absence of a certain resource that the machine may need, then proper action that results in finding such a resource in the environment will reduce this pain signal. In Fig. 1 the actual pain level, P, is controlled by a bias, B, (which is linked to the resource's level) times w_{BP} weight. A pain memory center stores the delayed pain level. The newly measured pain signal is compared with the previous pain signal in the second group of neurons responsible for learning control.

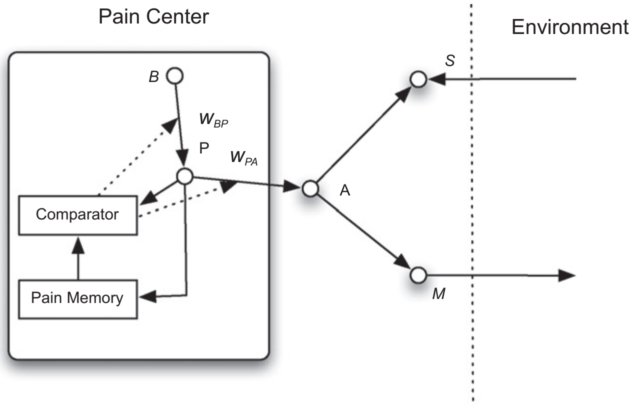


Fig. 1. Basic pain detection and learning unit.

Increasing pain signals force the machine to explore various motor actions by stimulating the action neurons, A, through initially random connection weights (as represented by w_{PA}). The machine searches for the proper action starting from the one with the strongest activation (strongest weights connecting to the pain stimuli). All action neurons and pain neurons compete with each other using Winner-Take-All (WTA) competition (see Fig. 3).

2.4. Representation building

The winning pain signal forces the machine to explore its environment to reduce the pain. A solution can be found through exploration or observation of another entity performing a desired task (Rizzolatti, Fadiga, Gallese, & Fogassi, 1996). In doing so, the machine discovers relationships between objects observed through its sensory inputs and actions it performs. Observed concepts are not predefined but emerge as a result of successful operations. Thus, a concept of an object is related to useful and predictable properties the object may have with respect to the machine's objectives and its ability to fulfill them with the proper actions(s). In connectionist networks, objects are recognized mostly through correlation and self-organization of similar features, while feature invariance building is accomplished through continuous observations and correlation through time. Reliable perception and invariant representation building are active research topics and their full discussion is beyond the scope of this paper. Thus, in our description of the ML mechanism and in simulation experiments we use predefined concepts and motor operations for simplicity. This however does not constrain ML's ability to learn new concepts and skills.

For the optimum development of concepts and related skills, the machine operates in a protective environment that gradually increases its complexity. Thus, the developmental process must be monitored and the learning environment structured in such a way as to facilitate the machine's learning.

An important observation is that representation building, (which results from the association of observed actions with the internal or external reward), comes from the motivation of the machine to act, whereas motivations to act come from representation building. New representations may yield new motivations to protect or acquire desired resources while new motivations force the machine to discover new ways of solving its problems and learning new concepts.

2.5. Creation of abstract pains and motivations

As soon as the machine discovers a valid action, any inability to perform this action in the future (lack of resources or deprivation of motor actions) will result in an abstract pain. For instance, if a machine needs a certain resource to satisfy its primitive pain, and the resource is not available, this creates an abstract pain signal. This abstract pain motivates the machine to explore how to obtain the missing resource. An abstract pain center uses a similar organization to trigger this motivation, as shown in Fig. 1. However, an abstract pain center is not stimulated from a physical pain sensor; it only symbolizes an internal pain from not having sufficient resources to lower its primitive or abstract pain.

At any given time, the machine may experience a number of different pains, each one triggering different goals. Changing pains change the machine's motivation for action, concentrating its efforts on reducing the winning pain. The same mechanism that created the response to a lower level pain will govern learning how to respond to abstract higher level pains. It will result in the emergence of a complex system of drives, values, and concepts about the observed environment. In addition, this motivating mechanism stimulates the machine to interact with its environment and to develop its skills.

For instance, an agent may suffer from a primitive pain when it is hungry. When "food" is available and the agent "eats", the primitive pain is relieved. An abstract pain center responding to lack of food is created. An inhibitory link is developed between the sensory signal representing the presence of "food" and the abstract pain center, and detection of "food" can inhibit the abstract pain. When "food" is not available, the agent tries to find a solution to reduce the "abstract pain" lack of food. Thus, the agent may feel the abstract pain (no food) without feeling the lower level pain (not hungry).

Motivated by the dominant abstract pain, the agent is forced to explore to reduce this abstract pain. Eventually, the reduction in the abstract pain of no food may result from the action "open" combined with the sensory object "refrigerator". This indicates that the abstract pain triggered by the absence of "food" will be associated with the sensory–motor pair "refrigerator"–"open". In the case of the machine opening the refrigerator and seeing food, the abstract pain "no food" is suppressed. This strengthens the interconnection weight between the abstract pain "no food" and the action that alleviates this pain "refrigerator"–"open," reinforcing the performed action. In addition, an expectation link from the action "open"–"refrigerator" to the sensory neuron "food" is built; thus "food" will be expected as the result of the action "refrigerator"–"open". This expectation link will be used for planning future actions with expected response from the environment. This process is illustrated using Fig. 2.

This abstract pain and related goal hierarchy can be further expanded. If the agent "opens" the "refrigerator", but the "food" is not found, the machine needs to try other options to suppress this abstract pain. It may explore the environment or use instruction. At this stage either RL or random search may be used. Once the machine "spends" some "money" (in a store) to buy food, food becomes available and the level 1 abstract pain ("no food") is reduced. Such an action is rewarded by an internal reward signal that depends on the effectiveness of the pain reduction. So, in the future, the action will be strongly stimulated by the abstract pain center "no food in refrigerator". However, when "money" is not available, an abstract pain center on Level 3 (not shown) is activated with an inhibitory link from "money". Subsequently, the machine needs to learn how to solve the abstract pain on Level 3 related to lack of "money", etc.

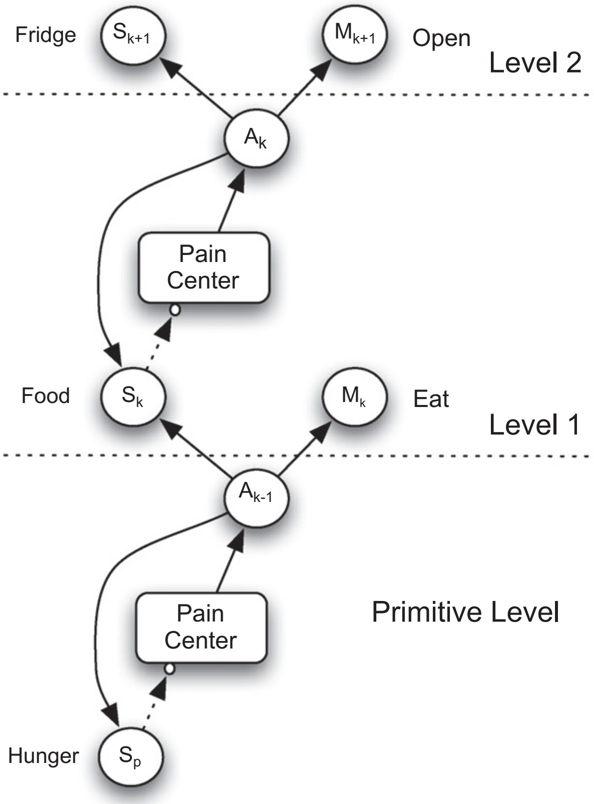


Fig. 2. Creation of abstract pain signals.

In motivated learning, at every step, the machine finds an action that satisfies its goals, and this action and the involved representations may result in creating further motivations and abstract goals. Therefore, via this simple mechanism, the machine simultaneously learns to match the goals with deliberate actions, the expected results of actions, the means to represent and obtain objects, and relations among objects. It learns which objects are related to its motivations. The machine governs the execution of actions to satisfy its goals and manages the goal priorities at any given time.

3. Neural network organization of learning motivations, goals, and subgoals

In this section we describe a neural network structure capable of creating motivations and abstract goals for a machine to act, learn, and develop. It uses externally defined pain signals that are associated with primitive pains. The presented neuronal structures are by no means the only possible organization of ML and are used here as an illustration of how one can implement ML. The following description is concerned primarily with the basic mechanism behind creation of abstract motivations and goals and does not necessarily explain the more complex motivational and planning structures.

3.1. Network organization

The goal creation system network, in addition to sensory, S, and motor, M, neurons, contains pain neurons, P, which register the pain signals, and action neurons, A, responsible for pain reduction. Each pain neuron is associated with its corresponding pain detection and learning unit, as illustrated in Fig. 1 and motivates the machine to act. Fig. 3 shows symbolically the structure of interconnections, between S, P, B, A and M neurons.

Selected pain center neurons, P_p, are connected to the external reward/punishment signals, S_p. In RL these neurons will receive a reward or punishment signal according to the training algorithm and in ML they will receive primitive pain signals. These pain signals directly increase or decrease activation of selected pain neurons. All pain neurons and action neurons are activated based on the result of Winner-Take-All (WTA) competition between them. Thus, a winning pain establishes a current goal and the winning action establishes an intended means to satisfy this goal. In this description, we will associate abstract pains with various sensory inputs. The number of action neurons is equal to the number of sensory–motor pairs.

There is one-to-one correspondence between sensory and pain centers, but there are no direct links between S and P neurons. There are feed-forward connections between the pain and the action neurons, between the action and the motor neurons, and feedback connections from the action to the sensory neurons. All pain neurons P_k (except those stimulated directly by the primitive pain signal denoted as P_p) have their own bias input B_k. In Fig. 3 an abstract pain neuron P_k connects to its bias, and action neurons are shown.

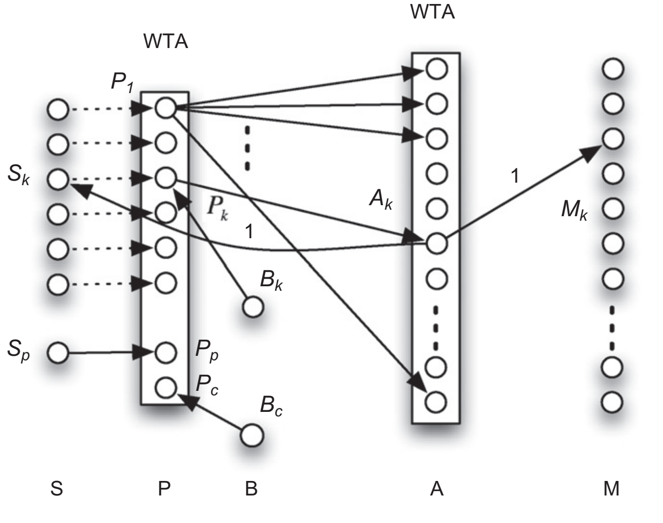


Fig. 3. Connections between sensory, motor, bias, pain and action neurons.

Action neurons, A, are connected to corresponding S and M neurons via weights equal to 1; P and A neurons are fully connected with trainable weights w_{PA}. There is no direct connection from the pain center neurons, P, to the motor neurons, M.

3.2. Goal related learning

The exploration of the environment starts from activation of a winning action neuron based on initially random values of activation links. As the machine learns, links are changed to reflect the acquired knowledge about the environment. The machine may also operate using links that were initially (genetically) set to handle the primitive pains. Such "genetically" set links facilitate learning of higher level skills and correspond to built-in skills. Genetic setting of lower level skills may be useful in designing machines that need to develop complex skills without repeating the learning cycle for lower lever skills.

Each time activation of a selected sensory–motor pair (S_k–M_k) results in a decrease of a dominant pain, P, there is an increase in the connection weights (w_{PA} on Fig. 4.) between this pain neuron and the action neuron, A_k, that corresponds to this (S_k–M_k) pair. The only exception to this rule is in a curiosity-based action signifying a decrease in curiosity due to gradual learning. Furthermore, the connection weights for the other non-curiosity pains are all increased or decreased depending on the effect of the curiosity-based action on the associated pain centers.

In addition, when pain decreases, the bias link strength of the abstract pain neuron P_k associated witch the selected sensory input S_k increases. The weights of other links to activated neurons are slightly decreased. The bias signal is associated with the probability of the corresponding sensory input activation indicating availability of the resource:

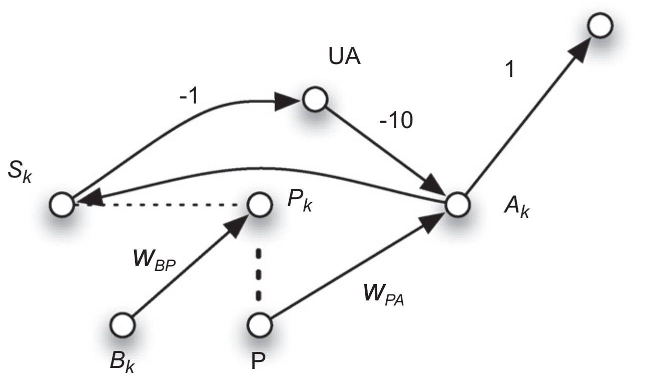


Fig. 4. Trainable connections between pain, bias, and action neurons.

$$B = -\log_2(\text{estimated probability}) \tag{1}$$

Reduction of the bias signal reduces the associated abstract pain P_k and triggers learning. However, if the dominant pain increases as result of the selected action, then the interconnection weight between corresponding P and A_k neurons is reduced. All pain and action related weights might be subject to a small reduction. Fig. 4 shows trainable connections between pain, bias, and action neurons.

Initial weights between P–A neurons are randomly selected in a 0–α_g interval (a good setting for initial weights is between $0.49\alpha_g$ and $0.51\alpha_g$ for faster learning). Assume that the weights are adjusted upwards or downwards by a maximum amount μ_g. In order to keep the interconnection weights within pre-specified limits ($0 < w_{PA} < \alpha_g$), the value of the actual weight adjustment applied can be less than μ_g and is computed as

$$\Delta \mathrm{w}_{PA} = \mu_g \min(|\alpha_g - \mathrm{w}_{PA}|, \mathrm{w}_{PA}) \tag{2}$$

This weight adjustment produces weights that slowly saturate towards 0 or α_g. (For a quick learning set $\mu_g = 0.5$). No other weights from other pain centers to this specific action are changed, so the sum of weights incoming to the node A is not constant. However, all w_{PA} weights from the selected pain center P to all actions A are adjusted to have a constant sum.

At the start all B_i–P_i weights are set to 0. The machine initially responds only to the primitive pain signal P_p directly stimulated by the environment. Each time a specific pain P is reduced the weight w_{BP} of B_k–P_k bias link increases. However, if the action activated by the pain center P is completed and does not result in a reduction of pain P, then the weights w_{BP} are reduced.

Since the bias weight B_k–P_k indicates how useful it is to have a desired S_k, bias weight adjustment parameter Δ_b must be properly selected to reflect the rate of stimuli to a higher order pain center. This rate reflects how often a given abstract pain center P_k was used to reduce the lower order pain P.

Bias links w_{BP} are adjusted to indicate a significance of each abstract pain. Each time an abstract pain is reduced as a result of an action its bias weight is automatically reduced according to $w_{BP1} = w_{BP1}(1 - \Delta_{b-})$ and the bias of the associated abstract pain is increased as $w_{BP2} = w_{BP2} + \Delta_{b+}\ (\alpha_b - w_{BP2})$. This adjustment takes place in two different goal centers as illustrated in Fig. 5.

Suppose that as a result of an action A involving sensory–motor pair (S_2–M_2) the resource on sensory input S_1 is obtained and reduces the pain P_1, then w_{BP1} is reduced and w_{BP2} is increased.

When, at the end of the training session, all the inputs to an action A are less than μ_g, the corresponding action is removed together with its entire set of incident links. This case typifies a useless action (like eating money) that did not reduce any pain. Since each pain neuron can be connected to S × M action neurons learning to remove an action neuron A may take on the order of S × M steps.

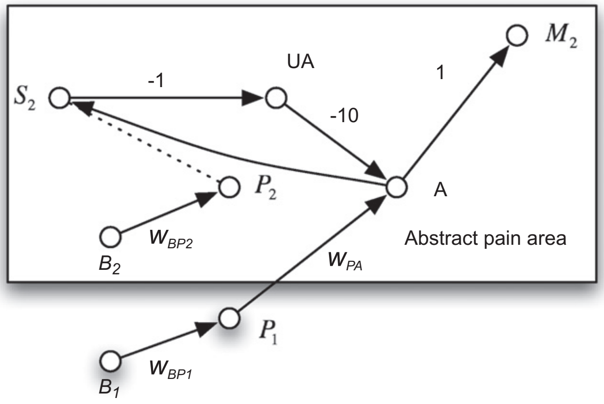


Fig. 5. Bias weights adjusted after action.

If a specific action is not invoked for a long period of time its importance in satisfying a lower level pain is gradually reduced. A similar reduction of B_k–P_k links indicates a gradual decline in importance of an abstract pain P_k. This mechanism of lowering the weights to an abstract pain center prevents the machine from overestimating its abstract pain importance by adjusting the relevance of this abstract pain to the lover level pain that was responsible for its creation. Otherwise, the machine can generate higher level goals even if they are no longer required to support its lower level goals. For instance, if making money is necessary to support living, an internal stimulus may force the machine to make more money even though the machine no longer needs it (or has a sufficient amount to cover its needs for a long period of time).

We understand that evaluation of one's goals may require a more complex mechanism than a constant rate of diminishing importance of goals that are not activated. However, for now we use this simplifying approach.

Sensory–motor sub-networks of the goal creation mechanism include unavailable action neurons (UA) whose role is to inhibit neuron A from firing if a sensory input required for this action is not present (see Fig. 4) or, in general, if the selected action cannot be performed. Each UA neuron fires automatically unless it is inhibited by the sensory neuron activation. In addition, the network has an abstract pain center neuron associated with each sensory input. Finally, a fully connected network of P–A links completes the network configuration.

The machine uses its goal creation approach to learn what to do and to adjust to changing environment conditions. It is doing so by adjusting pain biases and weights between the pain signals and actions.

3.3. Curiosity learning

How is curiosity learning organized in this neural network implementation of ML? It operates similarly to regular pain-based action with a few significant differences. A curiosity-based action will occur whenever none of the other abstract or primitive pains is above threshold and the machine still feels curious. It was mentioned, that curiosity based w_{PA} weights decrease in value to indicate that something was learned. The decrease in w_{PA} weights also indicates an overall decrease in curiosity. When all curiosity-based action weights, w_{PA}, have fallen below a predefined threshold, the machine will no longer perform curiosity-based actions, unless new concepts are identified and need to be explored.

In this network, curiosity is implemented as a constant low-level pain just above the pain threshold. This allows the machine to explore the environment when not performing any other pain-based actions such as eating food, or working. That's not to say that the machine might not try to eat food out of curiosity.

Let us consider such a situation, if based on its curiosity, the agent would observe that eating food reduces its primitive hunger and depletes its food supply. It would then adjust the appropriate pain-action weights, but would not learn anything pain related (via the bias-pain weights), since it did not perform the action based on a pain.

In addition to the continually decreasing curiosity w_{PA} weights, there is another factor in establishing a winning curiosity action that we refer to as "certainty". Certainty is a measure indicating how certain we are about a particular action. For example if any of the w_{PA} weights associated with a specific goal approaches 1, we can say that the certainty approaches 1. Then, when calculating the curiosity action value we multiply the w_{PA} value with one minus the certainty, because if we are certain about a particular action, there is no reason to be curious about it. Conversely, if all the weight values w_{PA} for a particular action A approach zero, we can say that the certainty for that action also approaches 1. In summary, certainty, C, is determined as:

$$C = 1 - \min(w_{PA}, 1 - w_{PA}) \tag{3}$$

This means actions that have been determined to do something (or nothing) useful will be assigned a higher certainty, while actions with indeterminate capabilities will continue to be examined.

3.4. Extensions toward subgoals

Some goals need a sequence of steps to implement them. These steps can be treated as subgoals and each subgoal may require specific conditions to implement. In this section we discuss how a series of goals, each with its own prerequisites, may be implemented. The concept of subgoals is well understood in RL. Thus, this discussion is intended to show the difference between abstract goals in motivated learning and subgoals in reinforcement learning, as abstract goals may in some cases be viewed as subgoals needed to implement a complex goal.

The neural network architecture to manage sequential goals slightly extends the preceding structures shown in Figs. 3–5 to allow for the creation of sequence of subgoals. Fig. 6 depicts this modified structure.

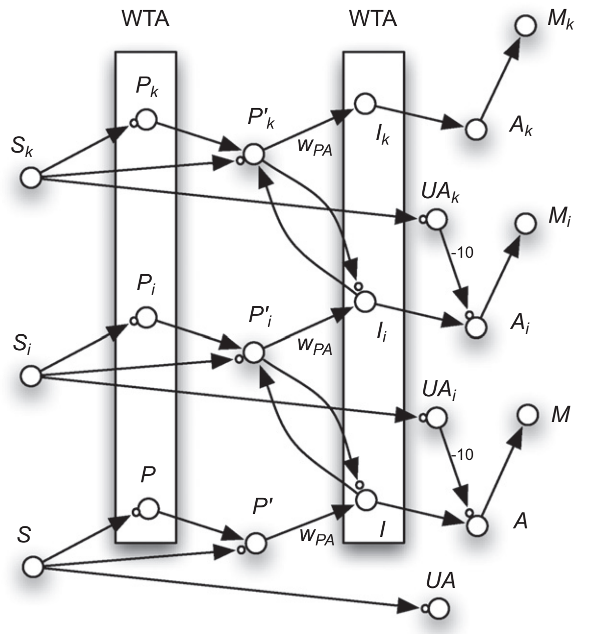


Fig. 6. Subgoal capable network.

The main difference between this structure and the earlier examples is the inclusion of the intention (I) and subgoal motivation (P′) neurons. The I neuron acts as a gate for the A neuron, such that there is one I neuron for every A neuron, while subgoal motivation motivates machine to implement a subgoal. Unlike the competition between action neurons A in Fig. 3, I neurons compete within the second Winner-Take-All (WTA) block to select the current intention. To illustrate the situation, let us assume a network that has already learned all of the necessary associations and that we have a winning pain P. This P will subsequently activate the associated P′ and I neurons (where the active I neuron is determined by a WTA event among all the available I neurons as determined by the weights w_{PA} from P′ to the set of I neurons). Let us assume that the resource required to implement this winning intention is not available. In this scenario, the I neuron will attempt to activate the associated action A, only to find that it cannot because a UA neuron, indicating that the needed resource or event is unavailable, blocks it. This will cause the I neuron to activate the P'_i associated with the UA neuron, which will in turn inhibit the current I neuron. The now active P'_i neuron will then attempt to activate its own I neuron, which we'll refer to as I_i. However, what if the A_i associated with I_i is unavailable as well? The process simply repeats with I_i activating a P'_k neuron, which in turn will cause an attempt to resolve the lack of resources needed for I_i.

Notice that several subgoal motivation neurons P'_i can be simultaneously active. They are deactivated on the completion of a corresponding subgoal.

Once A_k successfully completes, the resource associated with the sensory input S_k will become available, which in turn, will deactivate the UA_k and P'_k neurons, allowing for I_i and A_i to reactivate (notice that P'_i remained active). A_i will then be able to execute, which will allow the original I and A neurons to become active once again. Finally, the original action may be completed. This example considered only a simple situation with two subgoals; however, it can be easily applied to a significantly larger series of subgoals.

However, this also leads to questions, such as what happens in a circular system? For example, what happens if you need to patch a hole in a bucket, and through a series of subgoals you end up back where you started, in need of a bucket? Fortunately, this is a non-problem for the proposed network. Because the I neuron associated with using the bucket will have been inhibited earlier in the sequence, it will not be available later on for selection, meaning the network will be forced to find some other solution.

Another potential situation to be examined, is what happens when an action, A, requires multiple resources? The answer is to have multiple UA neurons attached to, and multiple links from, the neuron I back to the associated P'_i neurons (see Fig. 7). Only these P'_i neurons that correspond to tasks not yet completed will be activated. These activated P'_i neurons will trigger the corresponding goals in the activation order that results from their corresponding w_{PA} weights. Once all the required resources are obtained, then the I neuron will turn on the action neuron A to complete its task.

3.5. Summary of motivated learning approach

In contrast to classical reinforcement learning (RL), where the reinforcement signals come from the outside environment, the motivated learning mechanism presented in this paper generates internal reward signals associated with abstract motivations and goals developed by the machine. The machine's actions are followed by internal assessments of how well the internally set objectives were satisfied, and based on these assessments; an internal system of motivations, goals and values is built. At the same time, internal motivations are for accomplishing specific goals. Yet, when at any given time, an agent does not have specific goals; it uses artificial curiosity to explore the environment. This exploration helps the agent to learn its goal driven actions more efficiently.

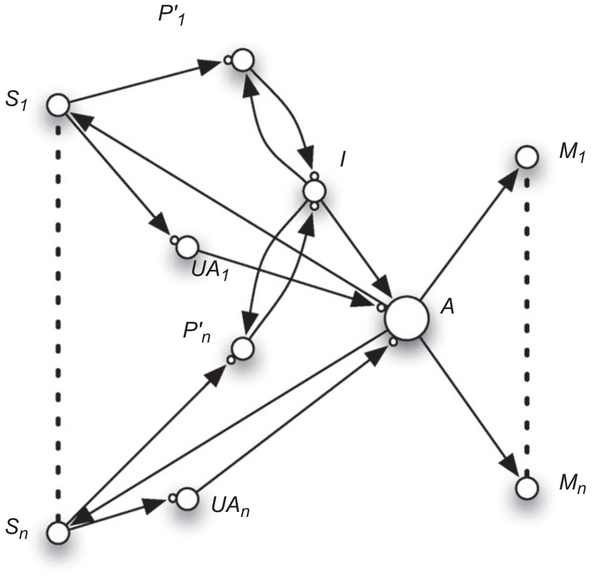


Fig. 7. Multiple resource requirements.

In our approach the agent uses not only the external reward signals (as the RL mechanism does), but is also motivated by internal abstract pains. The machine is motivated to reduce all pains. Internal motivations are created by the machine based on their relationship to externally specified objectives. Thus, the machine learns causal relations between its internal goals and externally reinforced ones. By learning how to satisfy the external goals, the machine learns to anticipate an outcome of its actions. The machine can also change a planned set of actions if the conditions in the environment indicate that the chosen actions cannot be successfully completed. This can be done, for instance, by blocking the selected action if the machine observes that a resource needed to complete the action cannot be found in the environment or if the resource is not of a sufficient quantity at a given time.

Learning complex tasks requires the implementation of several subgoals. In implementing a sequence of actions, the positive effect of these actions may be observed only at the end of the sequence. This is different than learning how to implement a higher level goal. For instance, earning money is a higher level goal that is created when a machine needs to buy food, and on the way to discover the means to accomplish this goal, the machine also learns a concept of money. However, driving to the store, selecting fresh produce, and spending money are subgoals for the goal of buying the food from the store. This requires a mechanism that monitors implementation of a sequence of subgoals. Another mechanism needed for complex goals, is failure detection. The machine needs to detect a failure and take corrective action by returning to earlier stages of goal implementation and trying alternative subgoals. A separate issue discussed in this paper is detection and avoidance of self-reference goals. This may happen when, in order to implement a goal, the machine needs to implement a sequence of goals, and in this sequence the original goal is used, leading to self-referenced goals. A mechanism was discussed to detect and avoid such self-referencing in the Extensions Toward Subgoals section.

Motivated learning is well equipped to deal with complex dynamically changing environments. It yields a machine that is motivated mostly by its internally generated abstract pains and related goals. In the presented Motivated Learning system, we model both artificial curiosity and goal creation to create intelligent systems. We model the first one to explore, and the second one to learn efficiently with a purpose. They complement one another in motivated systems as exploration and exploitation complement each other in reinforcement learning. However, unlike in RL, this allows the development of a complex structure of internal motivations, goals and rewards that makes learning more efficient. A machine equipped with a ML mechanism is allowed to pursue goals that are different than those set by the designer (those controlled by the primitive pain signals). In some situations, the machine may neglect its primitive pain, if an abstract pain dominates. This is not what a RL machine will do, as it is always in pursuit of its goals. A RL machine can identify and perform subgoals only if they serve to accomplish a goal, as was illustrated in the hierarchical reinforcement algorithms (Bakker & Schmidhuber, 2004).

These abstract motivations and goals in ML should not be confused with executing subgoals in RL. Once RL learns its subgoals' hierarchy, it will implement them to accomplish its goals. On the other hand, a ML machine may be motivated to perform a search for a solution to an abstract goal, even though it knows how and has means to accomplish its primitive (designer specified) goals. But by doing so, it may discover more efficient (intelligent) ways of dealing with changes in the environment, even when it was not instructed to do so. For instance, when environmental resources are depleted, the machine may already know how to deal with the new situation.

Pure curiosity based learning could give similar knowledge about the environment as ML, however, because it is not guided by any specific goal, its probability of discovering useful relations in changing environments is low compared to ML. It is similar to the difference in discovery of rules in the environment by playing (where everything new may be interesting to learn) vs. learning the rules related to set objectives and higher motivations to act with a specific purpose. In this duo, ML takes precedence over curiosity, since curiosity based learning is triggered only when all the pains are reduced below a specified threshold. This typically happens in early stages of development, when the environment is simple, and the agent has not yet developed many abstract motivations. Table 1 summarizes major differences between reinforcement and motivated learning approaches.

Table 1
Major differences between RL and ML.

Reinforcement learning	Motivated learning
Learns single value function – for an external goal	Learns multiple value functions – one for each internal goal
Measurable rewards – can be optimized	Internal rewards – cannot be optimized
Predictable	Unpredictable
All objectives are set by designer	Sets its own objectives
Maximizes the reward – potentially unstable	Solves a minimax problem – always stable
Focuses on most rewarding goal	Switches attention once a goal was accomplished
Learning effort increases greatly with complexity	Learns better in complex environment than RL
Always active	Acts when needed

4. Comparison between ML and RL

In this section we will show experimental results using the motivated learning approach in a complex environment with hierarchical dependencies. We have conducted several computational experiments to compare the effectiveness of the proposed motivated learning and reinforcement learning methods in a virtual environment. The RL algorithm was implemented through TD-Falcon (Tan, Ning, & Dan, 2008). TD-Falcon (Temporal Difference – Fusion Architecture for Learning, Cognition, and Navigation) is a generalization of Adaptive Resonance Theory – a class of self-organizing neural networks – that incorporates temporal difference methods (TD) for real time RL. This algorithm learns the value functions of the state-action space using temporal difference methods, and then uses them to determine the optimal action selection policy. Finally, it enables an autonomous agent to adapt and act in a dynamic environment with both immediate and delayed reinforcement signals. We have chosen this algorithm because of its superior performance (in terms of learning efficiency measured by the number of trials) in comparison to other implementations of RL.

The experimental setup consists of two main components: the environment and the agent. The agent is located in a hostile environment. By the term "hostile environment" we mean that the amount of available resources is limited but can be renewed by the learning agent through a deliberate action. Below we describe three experiments that tested various aspects of learning in such environments.

4.1. Setting the environment

The machine's actions change the environment, effecting the machine's perception of the environment and its strategy. In the simulated environment, limited resources that may exist in the environment are represented by a gradual decline in probability that a specific resource will be available.

For instance, if the machine spends money to buy food, the money supply goes down, regardless of whether the machine uses all the food or not. This simulates a case when food may rot if it is not consumed. Thus, the machine needs to learn to use resources wisely.

4.2. Simple mild environment

In the base benchmark task there are six different categories of resources that the machine can use in the environment. Five of them have limited availability (sugar, food supplies, money at hand, spending limits, and job opportunities). There are five categories of objects on which the machine can operate: Food, Grocery, Bank, Office, and School, listed in the order from the least abstract to most

Table 2
Meaningful sensory–motor pairs and their effect on the environment.

Sensory	Motor	Increases	Decreases
Food	Eat	Sugar level	Food supplies
Food at grocery	Buy	Food supplies	Money at hand
Money from bank	Withdraw	Money at hand	Spending limits
In the office	Work	Spending limits	Job opportunities
At school	Study	Job opportunities	–

abstract. There is one additional and most "abstract" resource that is inexhaustible. Thus, as the machine learns how to use all these resources, it can successfully operate in the environment. The agent has five sensory inputs and five motor outputs. Sensory inputs are sensitive to availability of a specific resource. Motor outputs represent actions that can be executed by the machine.

Table 2, indicates a basic, linearly hierarchical, arrangement of sensor–motor/pain interactions. We use a single primitive pain (low sugar level) and a simple linear hierarchy of abstract goals. This is done to simplify description of experiments and it is not a limitation of the ML approach. In general, many primitive pains can be used and will, together with abstract pains, compete for the machine's attention using a single WTA mechanism. In addition, complex relations between abstract pains (and related motivations) can be represented.

The "Increases" column describes the result of the motor action on the state of the environment or internal state of the machine. It indicates a positive improvement in the state of one of the available resources and corresponding decrease in the pain associated with that resource. The "Decreases" column describes the result of the motor action on the reservoir of goods related to that motor action. Thus, when the supply of a particular item in the "Sensory" field is low, the machine will attempt to take the appropriate "Motor" action to increase the supply. To elaborate, as time passes, the machine's sugar level decreases (meaning the hunger pain grows), leading the machine to take action to reduce this primitive pain. It will eventually do this by eating food. However, while eating food alleviates the pain, it decreases the available food supply, leading to a corresponding increase in the "lack of food" abstract pain.

In this paper resources are represented by the probability of finding the resource in the environment. As a resource is used the probability of finding it decreases. However, it can be restored with the correct action.

Fig. 8 shows how the machine handles the various pains. Notice how the maximum "Primitive Hunger" pain is initially high relative to the other pains, and how over the first hundred iterations its peak is lower than 0.7. Additionally, after the "Lack of Food" pain peaks (the solid black line), it takes the machine fewer iterations to pass the pain threshold (here set at 0.2) as the algorithm progresses. This is a result of the machine successively increasing the "importance" of the "Food" resource as it learns that "Food" is useful to reduce the "hunger" pain.

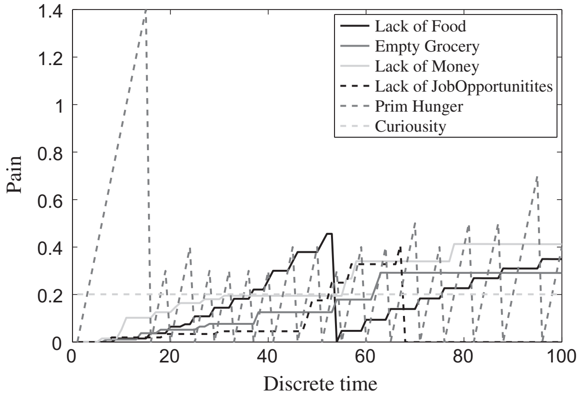


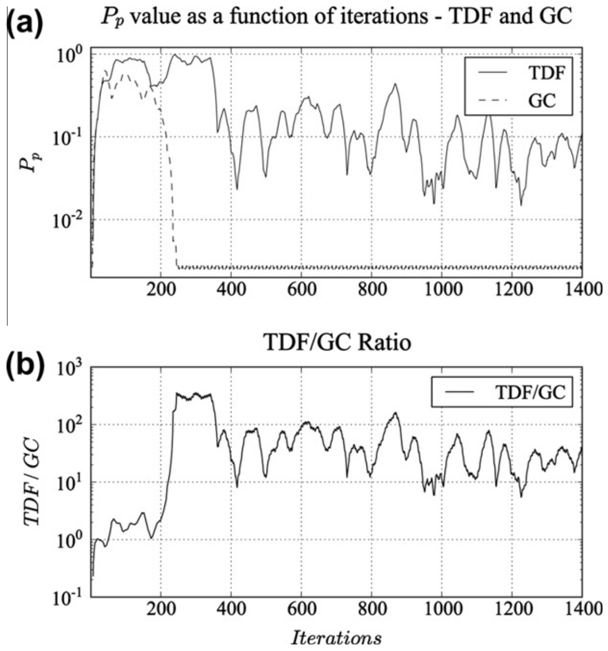
Fig. 8. Pain signal values in the first 100 iterations.

During the course of operation, the system will learn and adjust itself until it reaches equilibrium. In Fig. 8, only the first hundred iterations are shown. However, on the longer time scale, a point would be observed where the system decides to stop exploring its environment via "curiosity". At this point, its behavior becomes more regular due to the absence of semi-random resource consumption resulting from curiosity based exploration.

In these tests, we use the probability of finding a specified resource on the sensory input. The function which describes the probability of finding resources in the base experiment setup is as follows:

$$f_{ci}(k_{ci}) = \frac{1}{1 + \frac{k_c}{\tau_c}} \tag{4}$$

where: τ_c – scaling factor that describes a resource declining rate k_c – number of times a resource was used


(a) P_p value as a function of iterations - TDF and GC

(b) TDF/GC Ratio

Fig. 9. (a) Moving average of P_p value as a function of number of iterations and (b) TDF/GC P_p ratio.

Results of simulated actions in such simple environment are shown in Fig. 9. Fig. 9a shows the moving average value of the primitive pain signal P$_p$ for TD-Falcon (TDF – the solid line) and motivated learning based on goal creation method (GC – depicted via dashed line). The first observation is that GC yields much lower average pain than TDF and stabilizes sooner than TDF. The second graph shows TDF/GC P$_p$ ratio.

The GC based agent was able to learn how to use resources in order to minimize its pain in about half the number of iterations than the agent based on TDF could. Agents based on GC yielded from 10 to 100 times smaller average internal pain than TDF. This means that agents using the motivated learning method based on the goal creation system were able to control their environment better than those using TDF.

4.3. Complex mild environment

In the second experiment, instead of using an environment with only five hierarchy levels (each of which represents different resources) we have prepared several environments with deeper levels of hierarchy. The results obtained are illustrated in Fig. 10a and b for 8 and 18 levels of hierarchy, respectively, and show the average primitive pain levels in both methods.

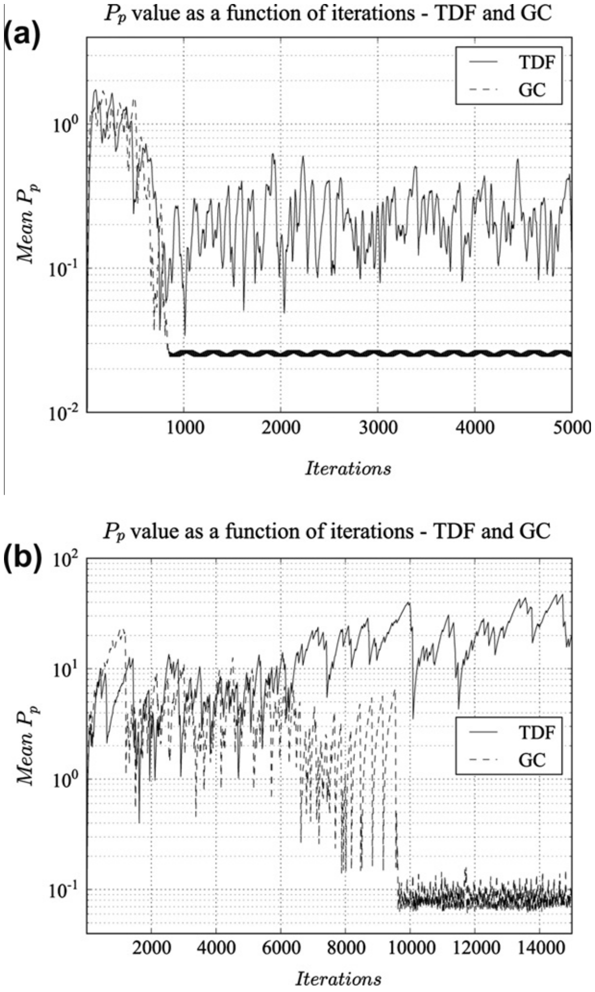


Fig. 10. Moving average of P$_p$ value as a function of number of iterations: (a) 8-levels of hierarchy and (b) 18-levels of hierarchy.

From these experiments we conclude that in some cases an agent using a RL algorithm can control its environment quite efficiently in the early stages of simulation (it behaves in a similar way to the ML agent based on GC). However, in later stages TDF is usually less effective at controlling its "internal pain". The reason for this initial success of RL is that this environment was not "hostile" enough. It means that even after extensive use of resources there were still enough resources in the environment and the RL agent could find them through random actions.

We can get mroe information about the efforts of both agents by observing their use of all types of resources available in the environment. Fig. 11 shows changes in the primitive pain signals and resource utilization on three levels of abstract hierarchy using both methods. The higher the signal value the larger the pain or resource utilization. Good resource management requires resource restoration, thus, the higher the resource utilization, the more difficult it is to find it in the environment and the more difficult it is to lower the primitive pain. As we can observe, the ML agent is able to manage all of the needed resources restoring them as they are used up, while the RL agent learned to manage resources only at the two lowest levels (levels 1 and 2 on Fig. 11). The RL agent uses resources from higher levels without learning how to restore them. This can be observed in the higher pain levels in Fig. 11a, versus those in Fig. 11b. By 2500 iterations, the ML agent learned how to manage all of the resources, causing its primitive pain level to be very low; however, this is not the case for the RL agent.

The only reason that the primitive pain of the RL agent was still modest is that the probability of finding the needed resource in this environment was still relatively high. The RL agent had no reason to learn (and explore) its environment because it was able to survive there without more significant learning effort. It had no motivation to improve its "skill level."

To expose this weakness of RL we designed another experiment for both the RL and the ML agents in which the environment was not only complex and dynamic but also very hostile.

4.4. Harsh environment

Higher hostility of the environment was achieved by changing the function that describes the probability of finding resources to the following:

$$f_{ci}(k_{ci}) = \exp^{-\frac{k_c}{\tau_c}} \tag{5}$$

where: τ_c – scaling factor that describes a resource declining rate k_c – number of times a resource was used.

After simulations in this more hostile environment we observed that the agent based on the RL algorithm (TDF) was not able to learn the higher dependencies

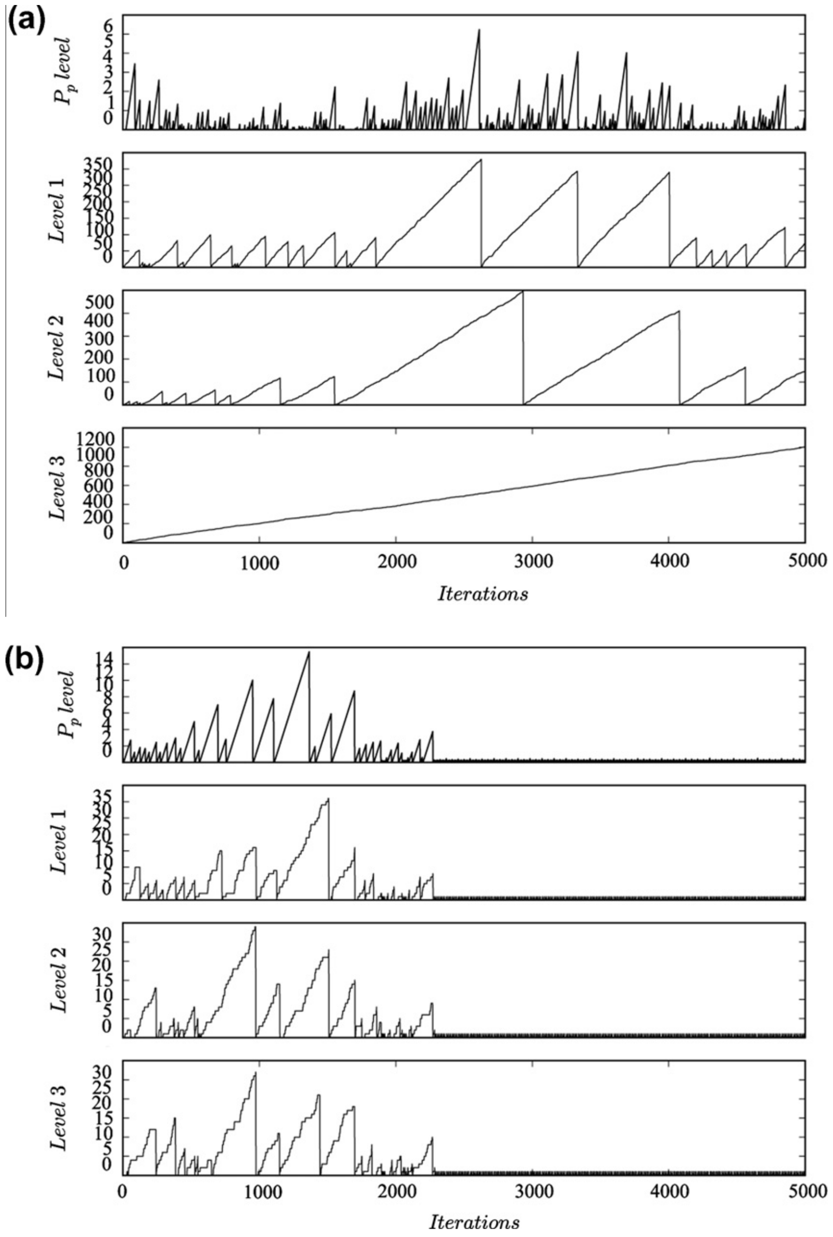


Fig. 11. Pain levels and resource utilization in: (a) RL and (b) ML. RL is not able to replenish resources on the Level 3.

between different available resources. After about 1100 iterations the RL agent had exhausted all the base resources and was not able to replenish them. After that time its "internal pain" started to grow almost linearly (Solid line in Fig. 12). We can also observe that the motivated learning agent (GC) (Dashed line in Fig. 12) was still able to learn all the dependencies between the environment's resources and use this knowledge to control its internal and external pains.

We observe that between 190 and 350 iterations the pain signal of the RL agent was much lower than that of the GC system. However, during this time, the GC system continued to learn the complex environment neglecting its primitive pain since its abstract pains dominated. At the same time, the RL agent used up all available resources trying to minimize its primitive pain, while the environment conditions were worsening. This example indicated a clear failure of the RL agent to learn behavior appropriate for this harsh environment.

4.5. Summary of experiments

In these experiments we demonstrated some disadvantages of the reinforcement learning method compared to motivated learning. ML outperformed RL in the learning task quickly converging to a stable solution, while RL after initial success was unable to accommodate changes in the environment and converge to a stable solution. This was particularly obvious in the more hostile environment.

Motivated learning can be combined with reinforcement learning to search for a solution to a well established goal (or a subgoal). The motivated learning mechanism will not

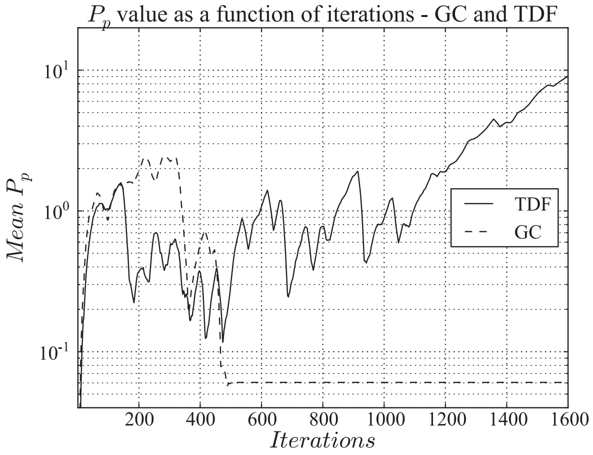


Fig. 12. Results of experiments in a more hostile environment.

only create internal abstract goals but it will also manage them. It will switch between these created goals (using internal motivations) when needed for the optimum performance. Its internal reward system will provide the "RL component" with reward information to learn an appropriate action.

We think that this kind of hybrid system will be able to take advantage of sophisticated methods developed for RL in order to efficiently solve problems where the environment is complex and has complex relations between its factors. One of the next most promising and relatively inexpensive ways to implement advanced ML methods in situations similar to real-world applications is to use them in simulations performed in virtual environments.

Reduction of resources was used as an easy to understand and implement example of changes in the environment. The agent is also exploring its environment as it learns. If the environment changes in other significant way (e.g. new technologies can be used to its advantage or the old one became obsolete) the agent will simply ignore older goals as "impossible" or less competitive and attempt to learn new solutions. Pain-action associations change by changing weights between them, thus new associations might be introduced and old ones may become less important.

5. Conclusion

This paper presents a new approach to machine learning and compares it with reinforcement learning. We discussed a need to extend machine learning methods in the direction of natural, goal oriented motivations useful for machine development. This motivated learning method can be combined with artificial curiosity and reinforcement learning. It enhances their versatility and learning efficiency, particularly in changing environments with complex dependencies between environment parameters.

ML is better equipped to deal with complex dynamical environments than RL, and can perform more effectively in a hostile environment with complex rules.

ML provides a much needed mechanism for switching a machines attention to new motivations and implementation of internal goals. A motivated learning machine develops and manages its own motivations and selects goals using continuous competition between various levels of pain signals (and possible attention switching signals). This form of distributed goal management and competing motivations is a core of "central executive" control that may govern the cognitive operation of intelligent machines as discussed in (Starzyk & Prasad, 2011).

We must emphasize that although the described ML approach is intended for autonomous agents, no robotic system was built that uses the proposed approach. Although the paper presents a preliminary study of ML, and there are numerous additional simulations that can be identified to further test the methodology, we believe that this methodology will prove to be a significant advancement for autonomous systems that must make their own decisions to successfully operate in harsh environments.

Acknowledgement

This work was supported in part by the Singapore National Research Foundation Interactive Digital Media R&D Program, under research Grant NRF2008IDM-IDM004-037.

References

Baars, B. J., & Gage, N. M. (2007). Cognition, brain, and consciousness (1st ed.). Academic Press.

Bakker, B., & Schmidhuber, J. (2004). Hierarchical reinforcement learning with subpolicies specializing for learned subgoals. In *Proc. 2nd IASTED int. conf. neural networks and computational intelligence* (pp. 125–130).

Barto, A. G. (2004). Intrinsically motivated learning of hierarchical collections of skills. In *Proc. 3rd int. conf. development learn, San Diego, CA* (pp. 112–119).

Barto, A. G., & Mahadevan, S. (2003). Recent advances in hierarchical reinforcement learning. *Special issue of Discrete Event Systems, 13*, 41–77.

Cohn, D. A., Ghahramani, Z., & Jordan, M. I. (1996). Active learning with statistical models. *Journal of Artificial Intelligence Research, 4*, 129–145.

Dayan, P., & Hinton, G. E. (1993). Feudal reinforcement learning. *Advances in neural information processing systems* (Vol. 5, pp. 271–278). Morgan Kaufman.

Derbyshire, S. W., Jones, A. K., Gyulai, F., Clark, S., Townsend, D., & Firestone, L. L. (1997). Pain processing during three levels of noxious stimulation produces differential patterns of central activity. *Pain, 73*(3), 431–445.

Dietterich, T. G. (2000). Hierarchical reinforcement learning with the maxq value function decomposition. *Journal of Artificial Intelligence Research, 13*, 227–303.

Fu, W-T., & Anderson, J. R. (2006). Solving the credit assignment problem: Explicit and implicit learning with internal and external state information. In *28th Annual conference of the cognitive science society*.

Harmon, M. E., & Baird, L. C. (1996). Multi-player residual advantage learning with general function approximation. Tech. rep., Wright Laboratory.

Hasenjäger, M., & Ritter, H. (2002). Active learning in neural networks (pp. 137 169).

Hsieh, J-C., Tu, C-H., Chen, F-P., Chen, M-C., Yeh, T-C., Cheng, H-C., et al. (2001). Activation of the hypothalamus characterizes the acupuncture stimulation at the analgesic point in human: A positron emission tomography study. *Neuroscience Letters, 307*(2), 105–108.

Huang, X., & Weng, J. (2002). Novelty and reinforcement learning in the value system of developmental robots. In *Proc. 2nd int. workshop on epigenetic robotics: Modeling cognitive development in robotic systems* (pp. 47–55).

Kaplan, F., & Oudeyer, P-Y. (2004). Maximizing learning progress: An internal reward system for development. In *Embodied artificial intelligence* (pp. 259–270). Springer-Verlag.

Kaplan, F., & Oudeyer, P-Y. (2007). In search of the neural circuits of intrinsic motivation. *Frontiers in Neuroscience, 1*, 225–236.

Melzack, R. (1990). Phantom limbs and the concept of a neuromatrix. *Trends in Neurosciences, 13*(3), 88–92.

Melzack, R., & Casey, K. L. (1968). Sensory, motivational, and central control determinants of pain. In Kenshalo, D. R. (Ed.), *The skin senses* (pp. 423–439).

Mesulam, M. (1990). Large-scale neurocognitive networks and distributed processing for attention, language, and memory. *Annals of Neurology, 28*(1), 597–613.

O'Reilly, R. C., & Frank, M. J. (2006). Making working memory work: A computational model of learning in the prefrontal cortex and basal ganglia. *Neural Computation, 18*(2), 283–328.

O'Reilly, R. C., Hazy, T. E., Watz, O., & Frank, M. J. (2007). Pvlv: The primary value and learned value pavlovian learning algorithm. *Behavioral Neuroscience, 121*, 31–49.

Oudeyer, P-Y., Baranes, A., & Kaplan, F. (2010). Intrinsically motivated exploration for developmental and active sensorimotor learning. In O. Sigaud & J. Peters (Eds.), *From motor learning to interaction learning in robots* (pp. 107–146). Berlin: Springer.

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

Parr, R., & Russell, S. (1998). Reinforcement learning with hierarchies of machines. *Advances in neural information processing systems* (Vol. 10, pp. 1043–1049). MIT Press.

Peyron, R., Laurent, B., & Garcia-Larrea, L. (2000). Functional imaging of brain responses to pain a review and meta-analysis. *Clinical Neurophychology, 30*(5), 263–288.

Pfeifer, R., & Bongard, J. C. (2006). How the body shapes the way we think: A new view of intelligence (bradford books). The MIT Press.

Pfeifer, R., & Scheier, C. (1999). *Understanding Intelligence*. The MIT Press.

Porro, C. A., Baraldi, P., Pagnoni, G., Serafini, M., Facchin, P., Maieron, M., et al. (2002). Does anticipation of pain affect cortical nociceptive systems? *Journal of Neuroscience, 22*(8), 3206–3214.

Rizzolatti, G., Fadiga, L., Gallese, V., & Fogassi, L. (1996). Premotor cortex and the recognition of motor actions. *Brain Research, 3*(2), 131–141.

Roa, S., Kruijff, G-J. M., & Jacobsson, H. (2009). Curiosity-driven acquisition of sensorimotor concepts using memory-based active learning. In *ROBIO '09: Proceedings of the 2008 IEEE international conference on robotics and biomimetics* (pp. 665–670). Washington, DC, USA: IEEE Computer Society.

Schmidhuber, J. (1991). Curious model-building control systems. In *Proc. international joint conference on neural networks, Singapore* (pp. 1458–1463). IEEE.

Schultz, W. (2002). Getting formal with dopamine and reward. *Neuron, 36*(2), 241–263.

Starzyk, J. A., & Prasad, D. K. (2011). A computational model of machine consciousness. *International Journal of Machine Consciousness*, *3*(2), in press.

Steels, L. (2004). The autotelic principle, in embodied artificial intelligence. *Lecture notes in AI* (Vol. 3139, pp. 231–242). Springer-Verlag.

Sutton, R. S. (1984). Temporal credit assignment in reinforcement learning. Ph.D. thesis, University of Massachusetts, Amherst, MA.

Tan, A-H., Ning, L., & Dan, X. (2008). Integrating temporal difference methods and self-organizing neural networks for reinforcement learning with delayed evaluative feedback. *IEEE Transactions on Neural Networks, 9*(2), 230–244.

Tölle, T. R., Kaufmann, T., Siessmeier, T., Lautenbacher, S., Berthele, A., Munz, F., et al. (1999). Region-specific encoding of sensory and affective components of pain in the human brain: A positron emission tomography correlation analysis. *Annals of Neurology, 45*(1), 40–47.

Vernon, D., Metta, G., & Sandini, G. (2007). A survey of artificial cognitive systems: Implications for the autonomous development of mental capabilities in computational agents. *IEEE Transactions on Evolutionary Computation, 11*(2), 151–180.

Weng, J. (2004). Developmental robotics: Theory and experiments. *International Journal of Humanoid Robotics, 1*(2), 199–236.

White, R. W. (1959). Motivation reconsidered: The concept of competence. *Psychological Review, 66*(September), 297–333.