NON-SUPERVISED SENSORY-MOTOR AGENTS LEARNING

Raul Sidnei Wazlawick Dr. Eng.
EDUGRAF - INE - Universidade Federal de Santa Catarina - Brazil
raul@edugraf.ufsc.br

Antônio Carlos da Rocha Costa Dr. Sc.
Instituto de Informática - Universidade Federal do Rio Grande do Sul - Brazil
rocha@inf.ufrgs.br

Abstract

This text discusses a proposal for creation and destruction of neurons based on the sensory-motor activity. This model, called *sensory-motor schema,* is used to define a sensory-motor agent as a collection of activity schemata. The activity schema permits a useful distribution of neurons in a conceptual space, creating concepts based on action and sensation. Such approach is inspired in the theory of the Swiss psychologist and epistemologist Jean Piaget, and intends to make explicit the account of the processes of continuous interaction between sensory-motor agents and their environments when agents are producing cognitive structures.

1. Introduction

The notion of an *autonomous agent* plays a central role in contemporaneous research on Artificial Intelligence [3]. *Cognitive agents* are based on symbolic processing mechanisms. *Reactive agents* are based on alternative computational mechanisms like neural networks, analogic processing, etc. The alternative approach using autonomous agents based on hybrid mechanisms has not been extensively explored, and the question of how to proceed for combining symbolic and non-symbolic mechanisms is an up-to-date research topic.

Our work [6] [1] is oriented towards Jean Piaget's Genetic Psychology and Epistemology [5]. According to that perspective, the two kinds of mechanisms are coordinated by the so called *equilibration mechanism*. The characteristic of this kind of coordination is that the symbolic mechanism results from an elaborated construction, called *reflexive abstraction* [5], based on elements given by non-symbolic mechanism.

The sensory-motor mechanism is based on Kohonen's non-supervised neural networks [2] and on genetic algorithms [4], that were combined for reproducing the main structure of the sensory-motor level, called *sensory-motor schema* [5]. This combined mechanism is used to build neural networks capable of self-adaptation in changing environments.

2. Activity Schemata

A *signal* is defined as a rational value between 0 and 1. A *generalized signal* is defined as a rational value between -1 and 1. An *alphabet of generalized signals* with a p digits mantissa is denoted by G_p. A sensorial entry e^n is defined as a signal string S_p^n. The i-th signal of e^n will be denoted by e_i^n.

A *genetic alphabet A* is defined as $A = S_p \cup \{\#\}$, where #denotes an improper element. The operation $difGene : S_p \times A \rightarrow \textbf{Nat}$, is defined by $difGene = \lambda s.\lambda g.(if\ g = \#\ then\ 1/3\ else\ \text{ABS}(s\text{-}g))$. The operation $sumGene : A \times G_p \rightarrow A$ is defined by $sumGene = \lambda g.\lambda s.(if\ g=\#\ then\ \#\ else\ g+s)$. The operation that adjusts a gene $g \in A$ to a signal $s \in S_g$, rated by a learning value α is given by: $adjustGene$: $A \times S_p \times S_p \rightarrow A$, where $adjustGene = \lambda g.\lambda s.\lambda \alpha.(if\ g=\#\ then\ \#\ else\ g + (g\text{-}s)*\alpha/2)$.

A *chromosome* is a string $<a_1, a_2, ..., a_n> \in A^n$. The *similarity* between a sensorial entry $s^n \in S_n^p$ and a chromosome $c^n \in A^n$ produces a value that corresponds to the average of the application of the function *difGene* to the genes of c^n and to the signals at the corresponding positions at s^n. The function $similarity : S_n^p \times A^n \rightarrow \textbf{Nat}$ is given by: $similarity = \lambda s^n.\lambda c^n.\ [\sum_{i=1}^{n} difGene(s_i^n, c_i^n)]/n$. The operation $crossover : A^n \times A^n \times \textbf{Nat} \rightarrow A^n \times A^n$ is defined by $crossover = \lambda x^n.\ \lambda y^n.\ \lambda r.\ (\ if\ r = n\ then\ (\ x^n,\ y^n\ )\ else$ $< x_1^n, ..., x_r^n, y_{r+1}^n, ..., y_n^n >,\ < y_1^n, ..., y_r^n, x_{r+1}^n, ..., x_n^n >)$. A *vector of mutation* is defined as a string of generalized signals from G_p. The operation $vmut : S_p \times S_p \times S_p \rightarrow \textbf{Nat}$ creates a vector of mutation respecting the following restrictions: $(i_1 * t)$ values are generated by a random function on the interval $[-i_2, +i_2]$ and $((1-i_1)*t)$ values are generated on the interval $[-i_3, +i_3]$. Therefore, $vmut = \lambda i_1.\lambda i_2.\lambda i_3.\lambda t.<randomSignal(i_1, i_2, i_3)_1, ..., randomSignal(i_1, i_2, i_3)_t >$. Mutation consists in the application of *sumGene* to the genes of a chromosome $c^n \in A^n$ with the values of a vector of mutation $v^n \in G_p^n$. Formally, $mutation : A^n \times G_p^n \rightarrow A^n$ is defined by $mutation = \lambda c^n.\lambda v^n.< sumGene(c_1^n, v_1^n),\ ...,$ $sumGene(c_2^n, v_2^n),\ sumGene(c_n^n, v_n^n)>$. The *reproduction of a pair of chromosomes* is an operation that generates a new pair of chromosomes. It's given by the function $reproduction : A^n \times A^n \rightarrow A^n \times A^n$ where $reproduction = \lambda c1^n.\lambda c2^n.(let\ i = random*n,\ let\ v1^n = vmut(0.1, 1.0, 0.2),\ let\ v2^n = vmut(0.1, 1.0, 0.2)\ in\ (let\ (p1^n, p2^n) = crossover(c1^n, c2^n, i)\ in\ (mutation(p1^n, v1^n),\ mutation(p2^n, v2^n)$)))).

A neuron is defined as a pair $<c^n, l>$, where $c^n \in A^n$ and $l \in S_p$. The element l corresponds to an activation threshold for the neuron, that is, an excitation level above that the neuron triggers. The domain of neurons of size n will be denoted by: $Neuron^n = A^n \times S_p$. The operation $application : S_p^n \times Neuron^n \rightarrow$ is defined by $application = \lambda s^n.\lambda r^n.max(0,\ similarity(s^n,\ chromosome(r^n)) - threshold(r^n))$. The *adjustment* of a neuron to a sensorial entry is given by the adjustment of each of its genes. This function is given by: $adjust : Neuron^n \times S_p^n \times S_p \rightarrow Neuron^n$, where $adjust = \lambda <c^n, l>.\lambda e^n.\lambda \alpha.<<adjustGene(c_1^n, e_1^n, \alpha),\ ...,\ adjustGene(c_n^n, e_n^n, \alpha)>, l>$.

A *schema* consists of a tuple $<<c_E^n, l_E>, <c_O^n, l_O>, a^m, \alpha, v>$, where $<c_E^n, l_E >$ is a sensorial entry neuron, $<c_O^n, l_O >$ is a goal neuron, $a^m \in A^m$ is an action chromosome, $\alpha \in S_p$ is a learning value, and $v \in S_p$ is an evaluation value. The schemata domain will be denoted by $Schema^{n,m} = Neuron^n \times Neuron^n \times A^m \times S_p \times S_p$. The *training* of a schema is given by $training = \lambda esq^{n,m}.\lambda s_E^n.\lambda s_O^n. <adjust(entry(esq^{n,m}), s_E^n, learning(esq^{n,m})),\ adjust(goal(esq^{n,m}),\ s_O^n,\ learning(esq^{n,m})),\ action(esq^{n,m}),\ learning(esq^{n,m}),\ evaluation(esq^{n,m})>$. The function $attenuation : Schema^{n,m} \times S_p \rightarrow Schema^{n,m}$ is defined by: $attenuation = \lambda esq^{n,m}.\ \lambda \varepsilon.\ <entry(esq^{n,m}),\ goal(esq^{n,m}),\ action(esq^{n,m}),\ \varepsilon \times learning(esq^{n,m}),\ evaluation(esq^{n,m})>$. The function $re\text{-}evaluation : Schema^{n,m} \times S_p^n \times S_p^n \rightarrow Schema^{n,m}$ is given by $re\text{-}evaluation = \lambda esq^{n,m}.\ \lambda s_E^n.\ \lambda s_O^n.\ <entry(esq^{n,m}),\ goal(esq^{n,m}),\ action(esq^{n,m}),\ learning(esq^{n,m}),\ (evaluation(esq^{n,m}) \times (1 - similarity(s_E^n, entry(esq^{n,m})) + similarity(s_O^n, goal(esq^{n,m})) * similarity(s_E^n, entry(esq^{n,m})))>$.

Let $e1^{n,m} = <<c1_E^n, l1_E, >, <c1_O^n, l1_O>, a1^m, \alpha1, \nu1>$ and $e2^{n,m} = <<c2_E^n, l2_E>, <c2_O^n, l2_O>, a2^m, \alpha2, \nu2>$ be schemata. The reproduction of the schemata $e1^{n,m}$ and $e2^{n,m}$ must originate a new pair of schemata $e3^{n,m} = <<c3_E^n, l3_E>, <c3_O^n, l3_O>, a3^m, \alpha3, \nu3>$ and $e4^{n,m} = <<c4_E^n, l4_E>, <c4_O^n, l4_O>, a4^m, \alpha4, \nu4>$ on the following way: (a) $(c3_E^n, c4_E^n) = reproduction(c1_E^n, c2_E^n)$; (b) $(c3_O^n, c4_O^n) = reproduction(c1_O^n, c2_O^n)$; (c) $(a3^m, a4^m) = reproduction(a1^m, a2^m)$; (d) The pairs of activation threshold $(l3_E, l4_E)$ and $(l3_O, l4_O)$ can be generated from $(l1_E, l2_E)$ and $(l1_O, l2_O)$ by the usual genetic process; (e) The values $\alpha3$ and $\alpha4$ must be fixed near to their upper bound; (f) The values of $\nu3$ and $\nu4$ must be fixed at an arbitrary value lower than 1/3.

A *population* consists of a set of schemata. The function *selection* : $Population^{n,m} \times S_p^n \rightarrow Schema^{n,m}$, is given by *selection* $= \lambda Q^{n,m}.\lambda s^n.(e^{n,m}$ such that $e^{n,m}$ is selected among the schemata of $Q^{n,m}$ with probability given by: $aplication(e^{n,m}, s^n) / \sum_{j=1}^{n} aplication(ej^{n,m}, s^n))$.

3. Non-Supervised Sensory-Motor Agent

A *non-supervised sensory-motor agent* will be defined as a set of sensors, effectors and a population $Q^{n,m}$. An agent lives in an environment $E^{n,m}$, that provides n signals to the agent's sensors, and it suffers actions composed by m signals coming from agent's effectors. An agent may be denoted by $A^{n,m} = <Q^{n,m}, e^m>$, where $Q^{n,m}$ is apopulation, and e^m are signals present in the agent's effectors. The agent's environment may be denoted by $E^{m,n} = <R^{m,n}, s^n>$, where s^n is a set of entry signals for the agent, and $R^{m,n}$ is a set of environment schemata. Let $s_{[t]}^n \in S_p^n$ and $e_{[t]}^m \in S_p^m$ be representations of signals at time t. Then, $<Q_{[t]}^{n,m}, e_{[t]}^m>$, is an *instant description of a cognitive agent*, and $<R_{[t]}^{m,n}, s_{[t]}^n>$ is an *instant description of an environment*. The working of a non-supervised sensory-motor agent $A^{n,m} = <Q^{n,m}, e^m>$ in an environment $E^{m,n} = <R^{m,n}, s^n>$ is given by the following algorithm:

Step 1: Selection. The agent observes the sensorial entry $s_{[t]}^n$ provided by the environment. Then the agent selects a winner schema for activation by $winner_{[t]}^{n,m} := selection(Q_{[t]}^{n,m}, s_{[t]}^n)$.

Step 2: Activation. The agent activates $winner_{[t]}^{n,m}$ by putting on the correspondent positions in $e_{[t+1]}^m$ the proper signals (those different of #) of the chromosome $action(winner_{[t]}^{n,m})$.

Step 3: Adjust. The agent adjusts the weights of the entry and goal neurons accordingly to the data provided by the environment: $entry(winner_{[t]}^{n,m}) := adjust(winner_{[t]}^{n,m}, s_{[t]}^n, learning(winner_{[t]}^{n,m})$; and $goal(winner_{[t]}^{n,m}) := adjust(winner_{[t]}^{n,m}, s_{[t+1]}^n, learning(winner_{[t]}^{n,m})$.

Step 4: Attenuation. The agent decrements the learning value to attenuating the adjusting rate of the schema: $winner_{[t]}^{n,m} := attenuation(winner_{[t]}^{n,m}, 0.95)$.

Step 5: New evaluation. The agent observes again the sensorial entry after the effectors activation. Let this observation be represented by $s_{[t+1]}^n$. The agent calculates the new evaluation of $winner_{[t]}^{n,m}$ by $winner_{[t]}^{n,m} := re\text{-}evaluation(winner_{[t]}^{n,m}, s_{[t]}^n, s_{[t+1]}^n)$.

Step 6: Reproduction. If the re-evaluation of $winner_{[t]}^{n,m}$ reduced the value of $evaluation(winner_{[t]}^{n,m})$, then the agent makes the reproduction of $winner_{[t]}^{n,m}$ with another schema given by $selection(Q_{[t+1]}^{n,m}, s_{[t]}^n)$. The agent adds the new pair of schemata to $Q_{[t+1]}^{n,m}$, and eliminates from $Q_{[t+1]}^{n,m}$ the two schemata with the smallest values for ν, and then returns to the step 1.

The constant adjustment of goal neurons increments gradually the evaluation of many schemata. In some moment, however, the action a^m applied to a certain entry may not product the desired effect, lowering the schema evaluation, and performing the role of a *perturbation*, as defined by Piaget [5]. Those perturbations begin schemata reproduction (Piaget has called this process *differentiation*). The reproduction doesn't destroy the schema that has suffered the perturbation, but the schemata with the lowest evaluation. Thus, the perturbed schema is maintained, and a pair of new schemata is created by reproduction. These new schemata will try to compensate the perturbation. In the case of reproduction with the perturbed schema, the weighted selection of schemata of highest value is in accord with the genetic algorithm process. This makes the agent to search some good schemata in order to try to accommodate the perturbation.

The interaction process between agent and environment may be though of as a process going on through the evolution of the system *Agent × Environment*, represented by $<A^{n,m}, E^{m,n}>$. The evolution of this pair is denoted by a sequence of cycles of the form $<< Q_{[t]}^{n,m}, e_{[t]}^m >, < R_{[t]}^{m,n}, s_{[t]}^n >>$ $\rightarrow_A << Q_{[t]}^{n,m}, e_{[t+1]}^m >, < R_{[t]}^{m,n}, s_{[t]}^n >>$ $\rightarrow_E << Q_{[t]}^m, e_{[t+1]}^m >, < R_{[t+1]}^{m,n}, s_{[t+1]}^n >> \rightarrow_A << Q_{[t+1]}^{n,m}, e_{[t+1]}^m >, < R_{[t+1]}^{m,n}, s_{[t+1]}^n >>$ where $\rightarrow_A$ represent steps of the agent's working, and $\rightarrow_E$ represent steps of the environment's working.

The sequence of those steps describes the join working of the pair agent/environment. It indicates how agent's actions modify its environment, and how these modifications affects the agent's schemata, and consequently it's sensory-motor capacity. Thus, it represents the agent's cognitive development, in interaction with its environment, and captures, in a certain sense, some processes identified by genetic epistemology.

4. Conclusions

The schema mechanism approximates a certain kind of human classification method. Concepts are associated to schemata, and schemata are built from observables, actions and goals. If the action reaches the schema's goal, then the observable belongs to the concept associated to the schema; otherwise the schema must be differentiated. The differentiation occurs when a couple of schemata is genetically reproduced. In this case, the action, observables and goals are recombined for creating new schemata, which try to better represent the activity structure related to the environmental objects. The genetic mechanism is responsible for finding the best configurations for the tuple *observable × action × goal*.

5. References

[1] COSTA, Antônio Carlos da Rocha *Machine Intelligence: Sketch of a Construtivist Approach*. Ph.D. Thesis, Porto Alegre: CPGCC-UFRGS, 1993.
[2] DAYHOFF, Judith E. *Neural Network Architectures - an introduction*. New York: Van Nostrand Reinhold, 1990.
[3] DEMAZEAU, Yves, MÜLLER, Jean Pierre *Decentralized Artificial Intelligence 1*. North-Holland, Elsevier Science Publishers, 1990.
[4] HOLLAND, John *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.
[5] PIAGET, Jean *The Equilibration of Cognitive Structures - The Central Problem of Cognitive Development*. Rio de Janeiro: Zahar, 1976.
[6] WAZLAWICK, Raul S. *An Operatory Model for Knowledge Construction*. Ph.D. Thesis, Florianópolis: PGEP-UFSC, 1993.