A PSYCHOGENETIC ALGORITHM FOR BEHAVIORAL SEQUENCE LEARNING

Vittorio Maniezzo, Matteo Roffilli

Dept. Computer Science, University of Bologna,
Contrada Sacchi 3, 47023 Cesena, Italy
e-mail: *vittorio.maniezzo@unibo.it, roffilli@csr.unibo.it*

Abstract: This work presents an original algorithmic model of some essential features of psychogenetic theory, as was proposed by J.Piaget. Specifically, we modeled some elements of cognitive structure learning in children from 0 to 4 months of life. We are in fact convinced that the study of well-established cognitive models of human learning can suggest new, interesting approaches to problem so far not satisfactorily solved in the field of machine learning. Further, we discussed the possible parallels between our model and subsymbolic machine learning and neuroscience. The model was implemented and tested in some simple experimental settings, with reference to the task of learning sensorimotor sequences.

1. Introduction

Artificial Intelligence (AI), ever since its foundation, has been characterized by its multidisciplinary perspective, where two essential components are computer science on one side and cognitive science, and notably cognitive psychology, on the other.

Along its over 50 years of history, the connections between AI and cognitive psychology have been intense and bidirectional. AI contributed to cognitive researches mainly by providing techniques and methodologies for analysis and validation, whereas cognitive psychology provided several effective models to serve as basis for algorithmic design: just to mention some, we recall here production rules (Quillian, 1968), semantic networks (Minsky, 1975), multilayer neural networks (Rumelhart, McClelland, 1986), syntactic trees (Chomsky, 1959), GPA (Simon, 1972), etc. These contributions have been so basic that there has hardly been an AI sector which did not declare cognitive foundations, at least in its beginnings.

However, while several among these sectors got actual benefits from this multidisciplinarity, some of them were not able to identify any true cognitive reference (for example qualitative physics, Hayes, 1978), or keep only loose linkages, essentially nominal ones. This is the case of symbolic Machine Learning (ML), which is a well-established, intrinsically multidisciplinary sector, since it includes contribution ranging from combinatorial optimization to non-standard logics, from control theory to neurobiology, but which has few cognitive references, usually derived from general AI topics such as knowledge representation or processing.

Cognitive psychology contributions to machine learning so far have in fact had few direct application linkages (Schank, 1977; Nosofsky, 1992), and this so more so for the topic of sensorimotor sequence learning or, more in general, of Reinforced Learning (RL). This is true despite the possibility of an algorithmic transposition of different cognitive theories.

RL has produced two main algorithmic approaches: one is rooted in dynamic programming (Q-learning, Watkins, 1989,1992; Temporal differences, Sutton, 1988) and one in global optimization and production systems (LCS, Holland, 1975). These algorithms proved effective in solving simple tasks, but show decreasingly acceptable performance when the complexity of the problem to solve increases (McDonald, 1997; Maltoni, 1994). This is partly due to the difficulties, intrinsic to these approaches, in finding long decision sequences, therefore in managing situations far from the stimulus / response (S/R) framework. In particular, some well-known problems derive from the difficulty to identify and maintain long decision sequences and from the need to explicitly reinforce sub-objectives, i.e., from the need to a priori decompose the problem to solve.

This work reports about the results of a first phase of a research aimed at validating the possibility to utilize the main theoretical model so far proposed in the area of cognitive psychology for sensorimotor sequence learning, i.e., psychogenetic theory and specifically J.Piaget's contributions. It is fact apparent that children, even very young ones, can sequence actions to achieve non-immediate objectives. We find therefore surprising that such an influential theoretic corpus as the works of Piaget and of his school have so far had so little impact on machine learning (we are in fact only aware of one single work, Drescher 1991, dealing with this topic).

We have therefore defined an original algorithmic transposition of the fundamental elements of psychogenetics, in order to obtain some preliminary computational results. These results are to be considered as a feasibility study for a more general project aimed at developing a cognitively well-founded approach and at the definition of new algorithms for unsupervised learning.

The paper is structured as follows. In Section 2 we briefly recall the objectives of unsupervised learning, in Section 3 we introduce the elements of psychogenetic theory that we included in our model. In Section 4 we describe the essential components of our model and in Section 5 we present the computational results that we obtained by means of the implemented system. Some discussion on the possible framing of our model in subsymbolic machine learning is contained in Section 6, while Section 7 further extends this discussion to neuroscience. Our current conclusions are discussed in Section 8.

2. Elements of reinforced learning

The name "reinforced learning" was introduced by Minsky (1954) with reference to behavioral psychology studies on animal learning. The essential idea of this learning paradigm is that, when an action is followed by a satisfactory situation or anyway by an improvement of the current one, then the probability to choose that action in the future increases, that is, the choice is reinforced (Barto, 1992). As we will detail in the following, the meaning of "being followed by a satisfactory situation" is to be considered wide enough to include long term outcomes of current choices. Fig. 2.1 shows a schema of reinforcement learning.
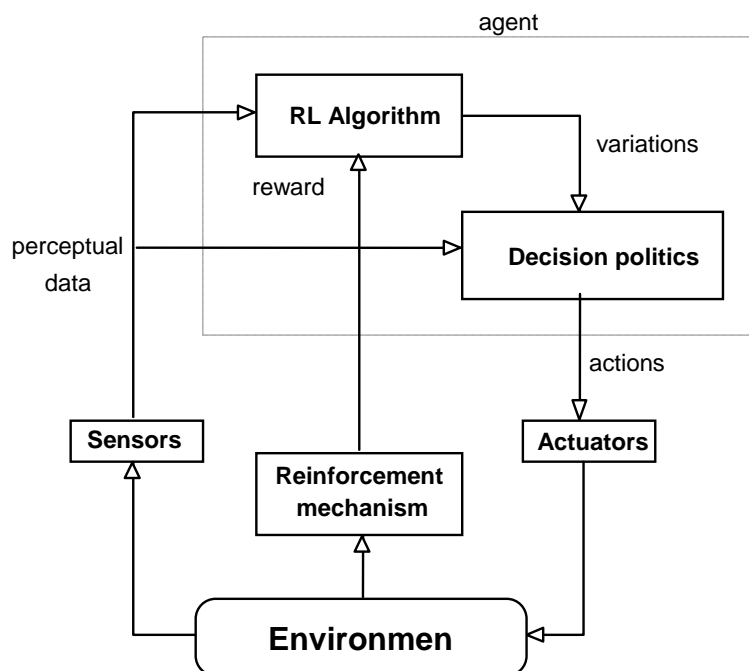

fig. 2.1- Reinforcement learning paradigm

The agent gets data on the state of the environment by means of its sensors and performs actions in the environment by means of its actuators. At each step, the agent observes the current state of the environment and generates an action according to its decision policy. The agent gets a reward, also called a payoff, after one or more steps, and goes through a state transition. The reward is used to modify the agent's decision policy, according to the specific RL paradigm implemented. In general, both rewards and state transitions can be stochastic.

The actions performed by the agent do not only affect the immediate rewards, but also the successive states, thus future rewards. The objective of learning is to generate a decision policy (a correspondence between states and actions) which maximizes the reinforcements received by the agent on a long term horizon.

Everything the agent knows about the external world is maintained as state variables. It is not guaranteed that what the agent does in a particular situation implements the best possible course of actions. The agent does not know whether (and how much) the past actions determined current rewards.

Rewards are generated as a function of the agent actions and of the states in which the actions were performed. It is generally assumed that the reinforcement mechanism is a part of the environment, that is, the agent is rewarded or punished by the environment it lives in. This setting is analogous to animal learning.

In order to implement a agent based on reinforcement learning, so that it can learn to perform a task, it is necessary to:

- design an effective representation of actions and agent inputs (i.e., effective data structures),
- design a reinforcement mechanism which contains the agent objective (algorithm).

RL is incremental in nature: the agent learns continuously while it works on its task. Agents in RL are adaptive and self-improving, thus they could potentially fit to complex domains.

3. Elements of psychogenetic theory of cognitive development

We present here some elements of Piaget theory (1936, 1964, 1975) of cognitive development that are relevant for our model. The essential points of psychogenetic theory that we included in our algorithm are the following ones.

Piaget defines four fundamental stages for the development of mental structures:

1. Stage of sensorimotor intelligence (children up to two years of age).

2. Stage of symbolical or pre-operational intelligence (from 2 to 7-8 years).

3. Stage of concrete operational intelligence (from 7-8 to 11-12 years).

4. Stage of operational formal intelligence (from 12 years on).

The first stage is further partitioned into six substages, during which the child completes its motorial development. We consider here only the first three ones:

- *First substage*: practice of reflexes (0-1 month of life).
- *Second substage*: first acquired adaptations and primary circular reaction (1-3 months).
- *Third substage*: secondary circular reaction and procedures aimed at maintaining interesting events (4-8 months).

Piaget's theory describes the mental development process according to a constructivist approach, by focusing on the capacity of the baby to build mental structures through a continuous interaction with the environment, and to progressively acquire consciousness of surrounding reality. Consciousness comes from the interaction between the progressively constructed mental structures and the physical or social environment, and it is the result of actions of the baby on physical items.

Experiences are stored as *schemata*, which are structures organized according to logical relations (spatial, temporal relations, etc.) that are mandatory to perceive and conceptualize the surrounding world.

The schema is the main mental structure in each stage of child mental development.

Piaget describes schemata as "something" which incorporates both child actions and the corresponding environmental reactions; "*a schema is what can be generalized of a given action ...*", it is a mental structure which allows to internalize the effects of actions in the world.

At birth the schemata present in the newborn (*innate schemata*) contain only primitive actions. Further on in the child mental development these schemata evolve. Schemata in fact change through two adaptation processes: *assimilation* and *accommodation*. Assimilation involves the interpretation of events in terms of existing schemata, whereas accommodation refers to changing the schemata to make sense of the environment. Piaget considers in fact intelligence rising from mental adaptation, where the adaptation is the equilibration of the action of an organism on the environment (assimilation) and of the action of the environment on the organism (accommodation). The subject assimilates the external world into existing structures (initially consisting in only inherited mechanisms) and updates them according to the experienced transformations, thus it accommodates them to the external objects. "*From this point of view, all mental life tends to*

progressively assimilate the surrounding environment, implementing this incorporation by means of structures, or psychic organs, whose range of action becomes ever wider [...]. In this way, by assimilating the objects, action and thought are forced to adjust to them, that is, to refine as a consequence of external variations" (Piaget, 1964. Translation from French by the authors, here and in the following).

An essential element of cognitive development is thus that cognitive structures are progressively constructed on the basis of innate functions, the processes of adaptation and organization, which work in response to external stimuli and to predefined internal organizations. The mechanism which maintains the equilibrium of the structures, both among themselves and with the physical and social environment, is named *equilibration,* and it works by error self-correction. It is affected by the degree of development of cognitive structures, by the amount of exercise and by the properties of interacting physical entities. Equilibration is the basis of the construction of internal structures, because it coordinates the effects of cognitive structures with those produced by external stimuli.

Without disequilibria knowledge would be static, it would stop at birth, since they play the role of activators of cognitive processes. When trying to re-equilibrate, the subject is forced to overcome his current non compensated state and to search in new directions, thereby acquiring new schemata useful for overcoming the adaptation difficulties.

We worked on this basis, designing an algorithm which constructs a basic knowledge by interacting with a surrounding environment using only some innate functionalities, in our case visual perception, grasping and motion.

4. A possible computational model

The symbolic ML algorithm that we designed and implemented simulates the development of sensorimotor intelligence in the first months of life of a newborn.

Before describing our model for the first three substages, we introduce the main representation structure of the model: the schema. We used schemata composed by three main parts (fig. 4.1), which is also in accordance with the model of Drescher (1991):

- context: the set of conditions (called *items*) that should be fulfilled in order to apply the schema.
- action: the activity performed by the agent.
- result: the state that could be reached after the execution of the action.
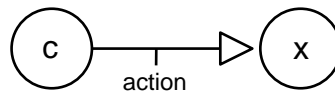
fig. 4.1 – A typical schema

A schema provides a prediction on what could happen if the given action is performed (in this sense it is thus a declarative representational unit) and thereby directs the agent toward the achievement of a given goal (in this sense it is thus a procedural representation unit). Moreover, a schema can compare what happens with or without a given action, or with or without the satisfaction of a given condition.

In the following, we will refer to a computational agent that must learn sensorimotor sequences in order to achieve a goal in a given environment. The goal is to grasp an object, the interaction with the environment is based on sensorial (complete view of the environment, tactile and proprioceptive sensations relative to the agent's 'hand') and motorial primitives (movements of the hand and of the foveal region of the sight).

The agent's state and its perception of the world are represented internally by suitable values of state variables called *items*. Typical items are hand_open, touch_object etc. They are therefore variables which represent a situation of the world and of the agent itself. The context and the result of a schema are represented by a conjunction of one or more items.

Each item can take three possible values:
- ON: when the associated sensorial input is satisfied.
- OFF: when the associated sensorial input is not satisfied.
- UNKNOWN: when no information is available about the associated sensorial input.

The context of a schema is s*atisfied* when all positive items included are ON. When the context of a schema is satisfied, the schema is *applicable*.

When an item is not satisfied, its relevance is not affected, but the opposite value of the associated state variable is forecasted.

As introduced in Section 3, the *first substage* is characterized by the simplest forms of adaptation, where we recall that adaptation is a keyword for defining the effect that the environment has on the mental structures. The agent makes use only of inherited reflexes, such as eyelid and pupillary reflexes, prehension or suction, which get consolidated by their repeated usage. These reflexes are included in innate schemata.

Innate schemata are in fact based on the exercise of inherited reflexes and are obviously strongly associated with the actions performed by the baby in its environment. Schemata are gradually refined, updated and restructured as a result of the interaction with the surrounding world and specifically of the experimental results of the baby activities. This permits an improved adaptation of the baby to its external conditions. Finally, the baby constructs its first basic knowledge by exercising reflexes.

The starting structure of our model (and Dreschler's, so far) is thus the schema. Initially schemata are composed of only inherited reflexes (primitive actions); these schemata are named *reflex schemata*.

During its first stage our agent tries each reflex schema in different, randomly generated situations, thereby discovering the effect of the action on the environment. We develop in this way new schemata with void context, but with an item in the result representing the effect of the action. These new schemata are called *partial schemata* (fig. 4.2):

fig. 4.2 – A partial schema

New schemata are constructed, and items are placed in their result, following a statistic procedure. Specifically, after computing the ratio between the probability of making the item satisfied after the execution of the action and the probability of satisfying it after any other action. If the ratio exceeds a given threshold, a new schema is made (interestingly, in analogy of what is proposed in Shalizi, 2002), which represents a possible effect of that action on the world. We have an *assimilation* of the external world within the already present structures (reflex schemata), resulting in new, partial schemata.

After the second month of life, the *primary circular reactions (second substage)* appear. They consist of the iterated repetition of a movement after having acquired an idea of its results. The objective of the repetition is the discovery of the conditions necessary to obtain the results previously identified. This leads to a gradual differentiation of partial schemata.

During the first stage, the empty schemata are modified and the effect of the action performed in the environment is encoded in the result. Each time an event seems relevant it is included in the result of the current empty schema, thereby obtaining a partial schema. The system then looks for

the initial conditions which are necessary in order to obtain the newly discovered result by means of the action included in the schema.

Notice that it is required to first determine the result of the action of an empty schema in order to later identify the possible initial contexts which enable the action to take place. This is necessary as it is almost impossible to start from the initial context in order to determine the result. The possible contexts are in fact too many and also too complex, since they include many elements which are irrelevant for the triggering of the action.

Our model of the process of the second stage is as follows. We separately consider each partial schema and we randomly generate a number of possible environmental conditions. In each condition, we control each item to determine whether its being verified (or not verified) is relevant for the success of the schema, i.e., for the achievement of the schema result when this is applied in an environmental condition which (does not) include the item in the context. The conditions, that is the items, to be actually included in the schema context are determined by a specific statistic procedure.

The schemata generated in this stage are structured as in fig. 4.3.
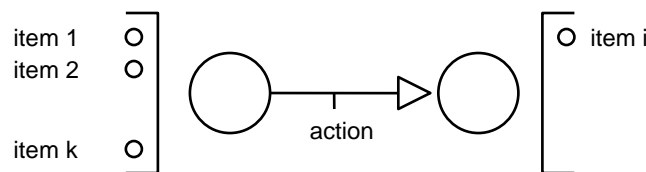

fig.4.3 – A complete schema

We associated with each schema a *reliability* variable which quantifies the success probability of the schema when this is fired having all context conditions satisfied.

Schemata constructed this way are very different, and some of them could be applied rarely, in situations which are quite uncommon. It is necessary to count the number of times when a schema is successful when applied, in order to determine the schemata which are more likely to be successful, thus are more reliable.

As mentioned, the engine behind any improvement in the agent intelligence, according to psychogenetics, is a continual strive for the equilibration of cognitive structures.

The equilibrium is affected by external perturbations, which hinder the exercise of the schemata. "*Perturbations initially consist of simple lacks: a momentary unsatisfied need, a lack of milk at the*

moment, etc. Afterwards, a perturbation is associated with what can be called spatiotemporal differences between the object and the subject: for example the looked-at object exits from the visual field, or a new global configuration, different from the one in which assimilation was being exercised" (Piaget, 1975).

The need, i.e., the objective which motivates the baby to act, has been modeled in our system by a *satisfaction* parameter (a scalar value). This permits to discriminate among the possible alternatives when a choice has to be made. Satisfaction refers to a single need and is assigned to a primary objective. In our tests, we always rewarded the action of grasping an object, better, the action which has the result Object_in_Hand. This satisfaction, when multiplied by the schema reliability, will be then backpropagated to the schemata which contributed to the reach of the objective.

Schemata can chain, thus defining composite actions. This can happen when an item representing the result of a schema is included in the context of another schema (fig. 4.4).
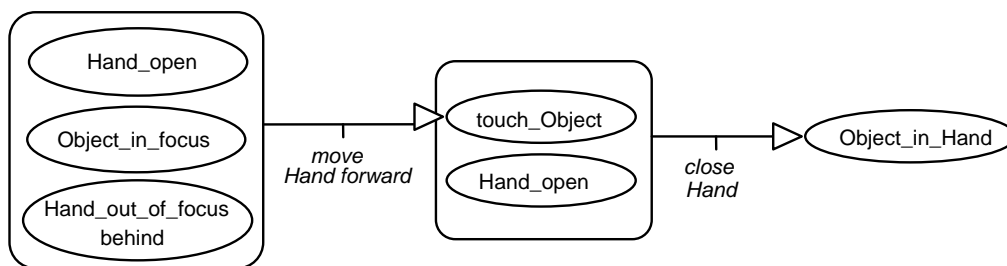
fig. 4.4 – Example of schemata chaining

Following schemata chaining, we can assign a satisfaction to the utilizable schemata. In our model, reward assignment is made only once, after the construction of basic schemata. This differs from Q-learning or classifier systems, where reward must be assigned at each state variation, that is, after performing an action of after a limited number of transitions.

The (expected) satisfaction is the basic trigger for activating schemata in the different environmental conditions, with no need to refer to further rewards or external mechanisms. It is furthermore important to notice that in our model we do not need any sub-objective in order to attain the final goal.

During the last phase of the second substage, a child begins to coordinate vision and prehension. The coordination of motor schemata is obviously of utmost importance, for example to reach goals such as grasping an object which can be seen, but which is far from the hand.

Our model can chain schemata in order to reach the final objective, which is in the following always grasping an object. Initially, we only have schemata with primitive non-composite actions, that must be chained. Schemata, which candidate for activation are those having their context satisfied in the current situation. The actually fired schema is the one which maximizes the value of the product of reliability and satisfaction.

Following the application of the fired schema, there could be a variation in the environment or in the agent internal state, thus obtaining a new situation. The process is iterated, choosing the same or other schemata (one same schema can be applied more than once if its context keeps being satisfied) until the final goal is reached. Finally, the schemata which were applied are chained and stored in a new schema, having the same context as the first applied schema and the final result equal to the final goal. The action of such a schema is a *composite action* (fig. 4.5).

fig. 4.5 - Schema with composite action

The satisfaction of the new schema is given by the satisfaction of the final one in the chain, while the reliability is the least among those of the chained schemata.

Schemata obtained this way can represent sensorimotor sequences like *a a a b b c c c d d*, where each letter denotes a schema. This is directly generalized into $a^*b^*c^*d^*$, passing from the repetition of one schema to a different one after a context variation. Moreover, our model permits the creation of new sequences starting from other ones:

$$a\,a\,b\,b\,b\,c\,c\,c\,d\,d$$
$$\quad\quad\quad \rule{}{} \quad e\,e\,f\,f$$

This happens when an existing sequence is started. For example, if schemata *a a b b b* are executed, but the following *c* schema then does not have its context conditions satisfied. A search for new schemata which can continue the sequence is thus started, in order to reach the final goal.

In this way, new sensorimotor sequences are identified, making the procedure adaptive to new situations.

Our model furthermore includes a *proximity* mechanism, which controls whether the result of the application of a schema is coherent with the current objective. The motivation behind this lies in the hypothesis that an agent, which has an object in its visual field, is structurally able to determine whether an eye movement makes the object nearer to the foveal region, or whether a movement of the hand makes it nearer to the object, when both can be seen. As we will point out in the following, this mechanism is needed at the current stage of development of our model, but we are studying a possibility to make it redundant, thus to eliminate it.

Anyway, the use of proximity helps the agent a lot in reaching an object, and in fact it can be actually considered as a sort of procedure for reaching a goal (in accordance with all reinforcement learning algorithms based on the S/R approach), but becomes a drawback for example in situations where the agent must bypass an obstacle to reach an object immediately behind it (fig. 4.6). The proximity mechanism then cannot find any schema to apply, since any action results in an increase of the distance between the hand and the object.
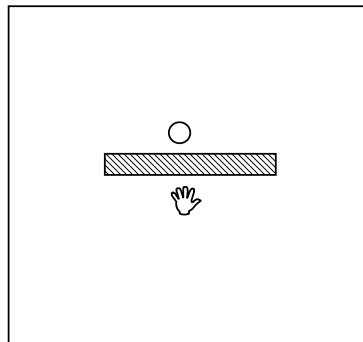
fig.4.6 – An object behind an obstacle

4.1 Algorithm and computational complexity

The algorithm implementing the described learning model is made by seven main procedures. The functionalities they implement are as follows.

1) *Exercise of reflexes*: implements the exercise of reflexes (first unintentional adaptations, first substage in the stage of sensorimotor intelligence) and stores the results of the tests for each item.

2) *Results of actions*: determines the results of the actions performed in the environment.

3) *Item success*: determines, for each partial schema and for each item, whether the considered schema is successful when applied with the item included positively or negatively in the context.

4) *Construction of contexts*: determines the possible contexts of each partial schema.

5) *Backpropagation of satisfaction*: backpropagates the final satisfaction onto the schemata which contributed to its achievement.

6) *Schemata chaining*: chains the schemata to get new sensorimotor sequences.

7) *Reliability*: computes the reliability of each schema.

   We determined the computational complexity of each of the listed procedures, obtaining the results reported in table 4.1, where n is the number of tests, m is the number of items, #S is the average number of items in the context and h is the average number of schemata with the same item in the result.

table 4.1 – Computational complexity of the main procedures

Procedure	*Complexity*
1) Exercise of reflexes	$O(m \cdot n)$
2) Results of actions	$O(m^2 \cdot n)$
3) Item success	$O(m \cdot n)$
4) Construction of contexts	$O(\#S \cdot m \cdot n)$
5) Satisfaction backprop.	$O(\#S \cdot m \cdot h^m)$
6) Schemata chaining	$O(\#S \cdot m)$
7) Reliability	$O((\#S)^2 \cdot m \cdot n)$

   Notice how the whole procedure has polynomial complexity, except for the backpropagation procedure. Procedures 1 to 5 are executed only once; moreover, the tests in different environmental

conditions cause an update of the cognitive structures, which is implemented only in the schemata chaining procedure.

5. Computational results

This Section presents results obtained from an experimental study to assess the effectiveness of our algorithm. The algorithm is always applied in the context of a simulated agent in an unknown environment with a task to fulfill (grasping an object), and possibly with obstacles to bypass.

All results have been obtained using an Intel Pentium 4 3.0 GHz machine, both for the rule generation phase and for the goal-directed actions.

We tested the algorithm in two different situations: in the first one, the agent must move and resolve the given task in an environment containing only one object, while in the second one an obstacle is present, too.

The environment is a bidimensional grid of dimension 27×27, therefore consisting of 729 cells, each of which represents a region in the environment which can contain *i)* an object, *ii)* a part of an obstacle, or *iii)* the hand of the agent.
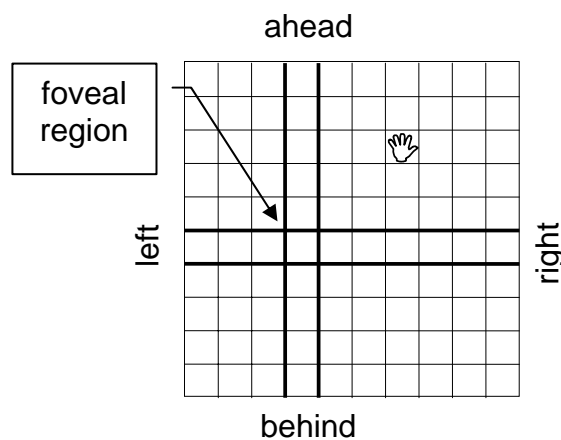
fig. 5.1 – The nine regions of the visual field

We obtained the initial configurations by randomly generating the coordinates of the position of *(a)* the object, *(b)* the foveal region, *(c)* the hand and *(d)* the state of the hand (closed or open).

Obviously, it could be possible to generate all the possible initial situations from the combinations of (a) (b) (c) (d), but that would induce a high computation time and it would add little to the results, being obviously a non scalable approach.

In the simple configuration of an environment with only one object present, the total number of configurations is $729 \times 729 \times 729 \times 2 \approx 750 \times 10^6$, while when there is also an obstacle composed of

three connected cells, the number of configurations rises to $\approx 500 \times 10^9$. The corresponding possible perceptive configurations are 9×9×2=162, where the 9 indicates the possible positions of the object and of the hand when the coordinates are centered in the foveal region of the agent (the visible area is subdivided into 9 regions, see fig. 5.1) and the 2 represents the possible states of the hand. The number of the possible perceptive configurations is thus small, which makes plausible the generation of a number of schemata that can cover all possible configurations.

When the environment contains one object and one obstacle composed by three cells, the total number of the combinations becomes 729×729×729×(27×25)×2 $\approx 500 \times 10^9$ and the corresponding possible perceptive configurations are 9×9×2×15 = 2430, where 15 are the perceptive configurations for the obstacle. Needless to say, the number of the possible perceptive configurations is greater than the previous one, and the generation of a number of schemata that can cover all possible configurations begins to be implausible.

From the analysis of the possible perceptive and environmental configurations and from the tests carried out, we can conclude that the algorithm always succeeds in an environment without obstacles, while it has problems in dealing with some situations where obstacles are present.

Below, we sketch an actual trace observed in a situation with no obstacles. In this scenario, the object is out_of_focus behind the foveal region (which is the single cell at the intersection of the highlighted column and row) and the hand is in the out of focus region at ahead-right (see fig. 5.2(a)).

(a)   (b)   (c)
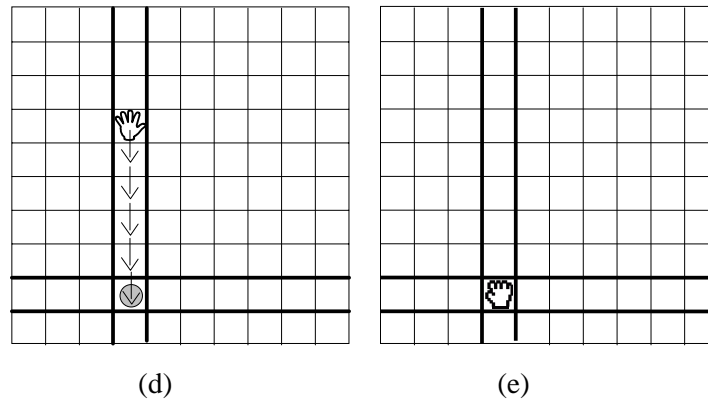
(d)   (e)
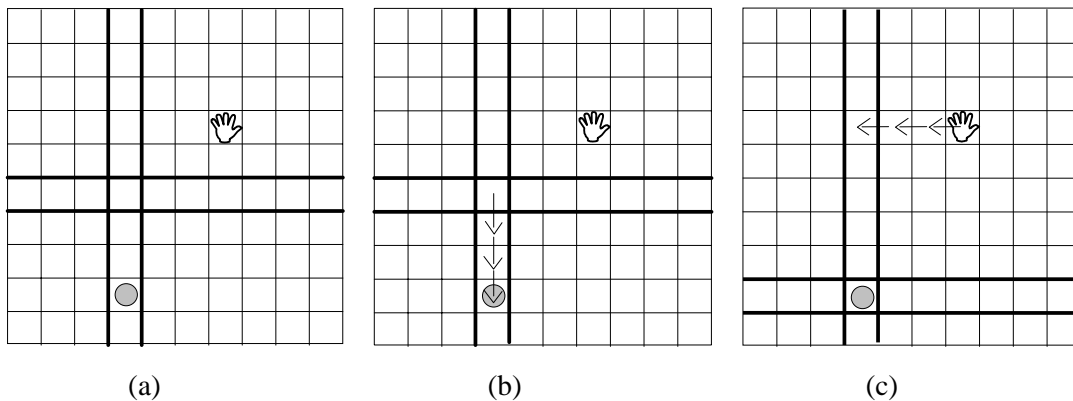
fig 5.2 - The agent grabs the object in an environment without obstacles

In fig 5.3, we present which schemata have been applied by the agent in order to grab the object. Every single schema is activated as many times as are the movements to be executed.
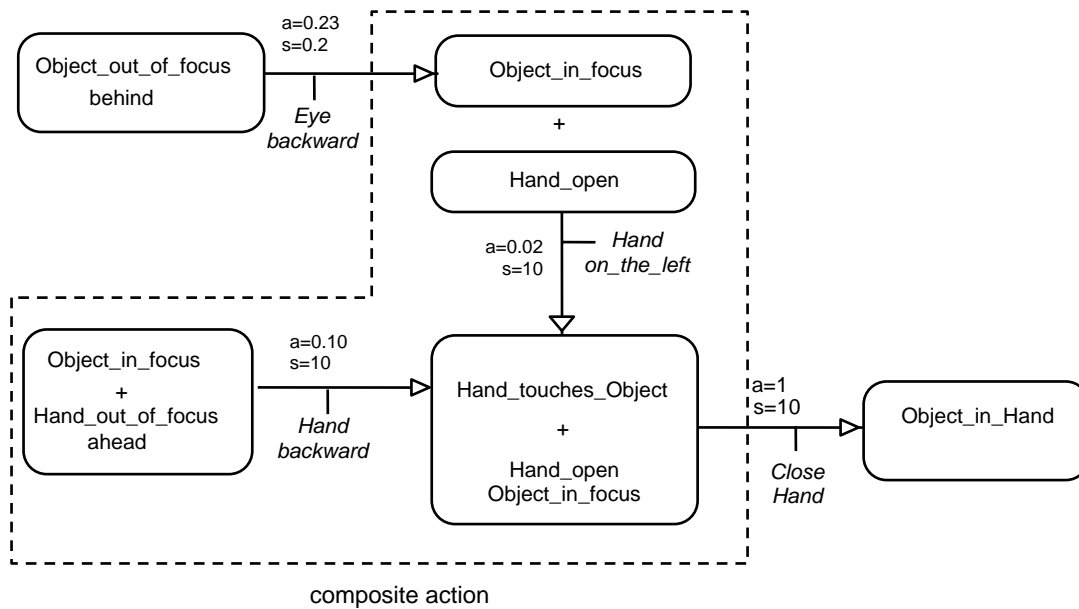

composite action

fig 5.3 - Sequence of used schemata in the run of reference

For example, the schema Hand_open/ move_Hand_on_the_left/ Hand_touches_Object is applied until the hand reaches the in foveal zone (see fig.5.2 (e)).

From this moment, the context changes and the schema Object_in_focus+Hand_ahead/ move_Hand_behind/ Hand_touches_Object can be applied, since its product satisfaction × reliability is greater than that of other schemata which are candidate for activation.
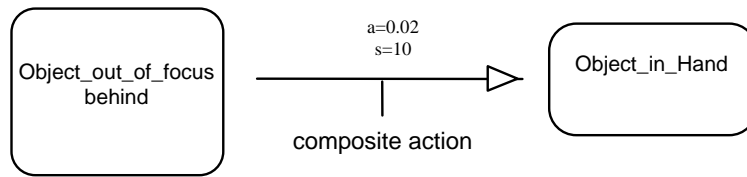
The new created composite schema is:

fig 5.4 - A composite action in an environment without obstacles

In the following second case, we show a typical behavior of the agent in situations where its hand and the object are separated by an obstacle.

First, we present a situation in which the agent catches the object.
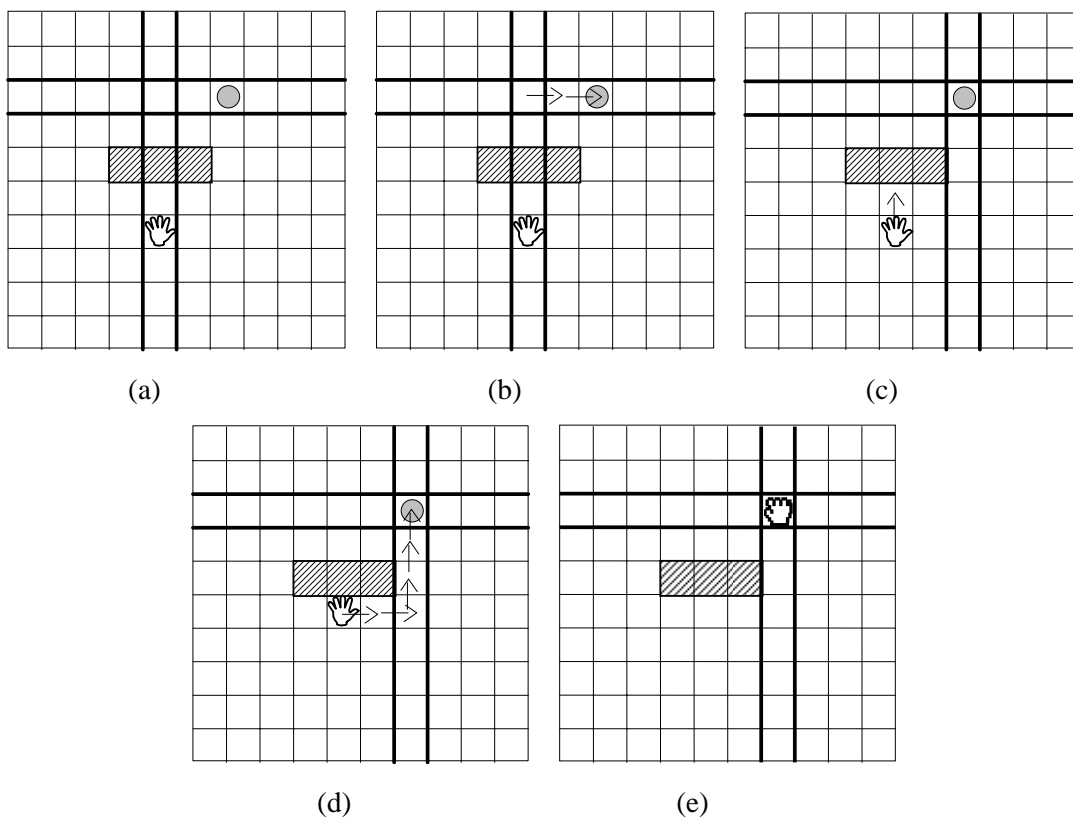

(a) (b) (c)

(d) (e)

fig 5.5 - The agent gets the object in an environment with one obstacle

Notice that in fig. 5.5 (d) the proximity mechanism is able to help bypass the obstacle since there is always a movement direction which reduces the distance from the hand to the object. Below, we show the schemata applied by the agent in order to get the object.
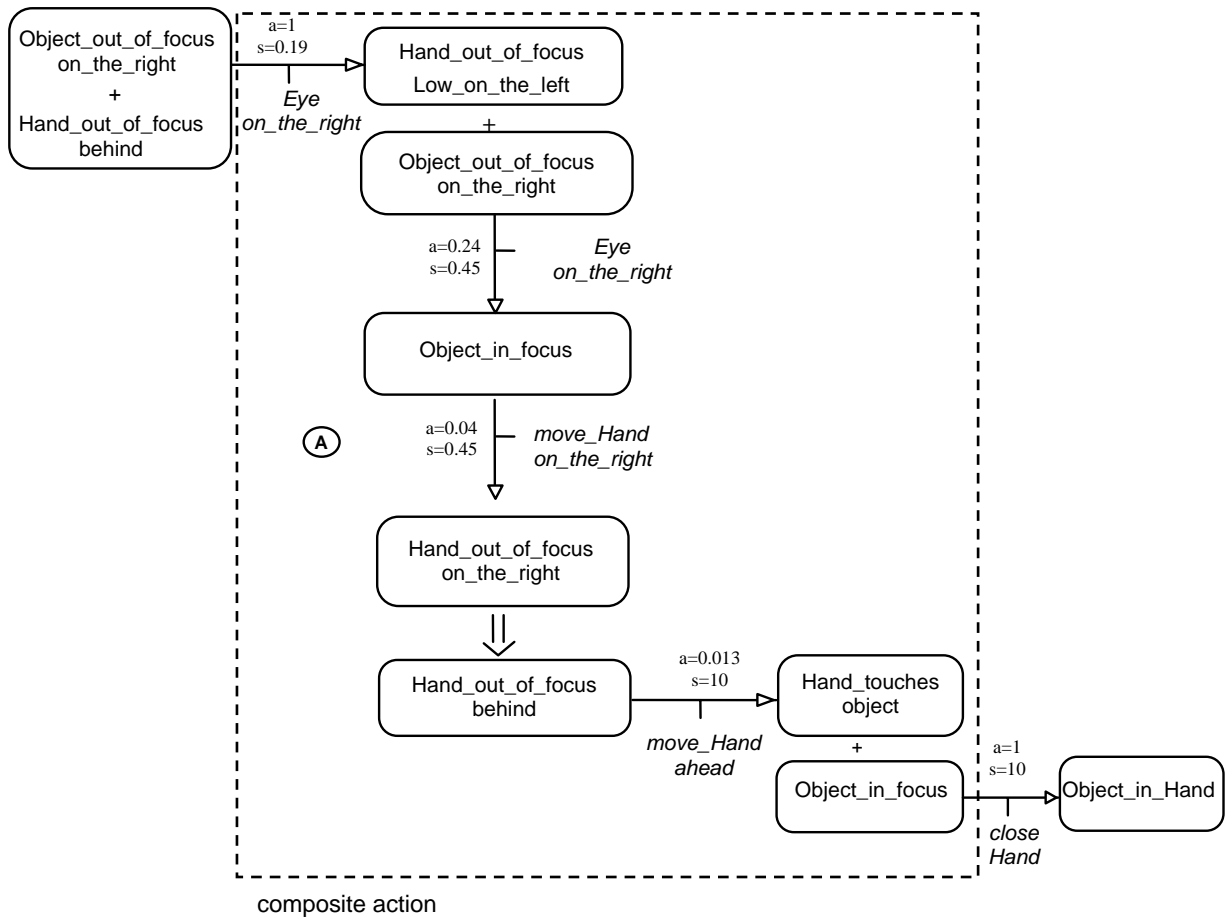
fig 5.6 - sequence of applied schemata

Notice that the result of schema (A) is never obtained. In fact, schema (A) serves only to move the hand towards right in order to obtain a new situation where it will apply the schema:

Hand_out_of_focus_behind/ move_Hand_ahead/ Hand_touches_Object

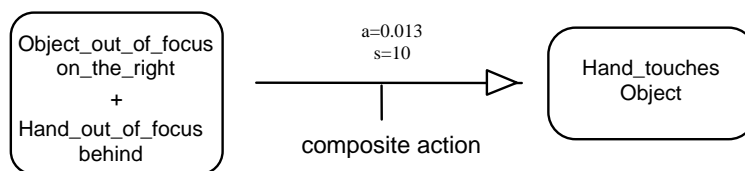Following this, the new schema with the composite action becomes:

fig 5.7 - A composite action in an environment with one obstacle

The average number of generated schemata in the case of environments with no obstacles is close to 500, and the time needed to construct them has been 0.97 seconds. The time required to define the reliability of each schema has been of 25.52 seconds, while the backpropagation procedure lasted 13.62 seconds. Thereafter, the agent needed an average of 0.002 seconds to reach the object, starting from different randomly generated positions of the hand and of the object.

The number of schema generated in the case of an environment with an obstacle has been of 2947, the construction phases required proportionally more time, while the goal is then achieved in an average of 0.03 seconds.

6. A subsymbolic Machine Learning interpretation

While the algorithmic transposition of Piaget's model so far presented can be effective in synthetic experiments, we are interested in validating it in real environments, too. The presence of noise is the key distinguishing feature of real-world contexts. The noise is present in different forms, from data corruption to bad sampling. It is worth noting that this problem could affect considerably the symbolic manipulation of the schemata. For example, if the agent knows the relative position of the target object with incorrect approximation, the use of the most reliable schema could not produce the expected result, and this is what reliability is there for. While the test of our symbolic approach in real world settings will be the object of future development of our research, we like to introduce here some considerations about unexpected structural similarities among our approach and some of the best performing ones in real worls settings.

In order to tackle noisy tasks, it is customary in fact to consider a different class of machine learning algorithms, which manage numbers instead of symbols in order to allow some degree of fault tolerance. These are the so-called subsymbolic machine learning methodologies, where Artificial Neural Networks (Bishop, 1995) and Support Vector Machines (based on the Statistical Learning Theory, Vapnik, 1995) are the foremost alternatives. One of the main features of these methods is the ability to face classification, regression and density estimation tasks without explicit symbol manipulation. In other words, training (or parameter fitting) is performed directly by analyzing data in numerical form (data driven). Neither symbolic nor semantic association are needed, and all the information is packed in a vector of features, where each feature represents a characteristic of the process under analysis. Under this conditions, the dataset, from which the

model is trained, constitutes a geometrical space (features space) populated by samples of the target process. For a comparative overview of ANN and SVM (see Kecman, 2001).

Interestingly, subsymbolic ML is a mathematically well-founded theory which seems able to formally relate to Piaget's theory, providing insight both from a statistical and from a computer science point of view. Conversely, we notice how the Piaget model fits very well within the fundamental assumptions of the subsymbolic ML theory. In the following, we will try to rewrite the model presented before, reframed in the context of subsymbolic ML. To this aim, we will use a terminology gathered both from ANN and from SVM (in bracket).

In essence, in subsymbolic ML we consider learning as the effect of an algorithm able to construct an internal structure that captures the main information embedded inside the raw data presented in input.

Usually, knowledge gained through learning in part consists of descriptions of what we have already observed, and in part is obtained by making inferences from (past) data in order to predict (future) examples. This second part is called generalization, or induction. Obviously, if data have no regularities, no law incorporated into them, we will not be able to find any new knowledge. In other words, in random data there is no knowledge to be found.

The aim of learning is thus both to obtain a predictive function that demonstrates capable of making good predictions in the given domain (environment) and simply to understand the observe processes. Needless to say, bunches of algorithms formalize this approach to learning in computable procedures. In the following, we will sketch the ANN (SVM) method.

In the case of ANN (SVM), the set of feasible network architectures (kernel functions) represents an a priori knowledge about the problem domain. In order to better capture the laws inside the input data, the network construction proceeds by adding hidden layers (degrees of freedom) to the basic perceptron architecture (linear kernel function). After the setting-up of initial ANN (SVM) architecture, the learning algorithm incorporates three main components:

1. a training set S: a collection of ℓ objects $S = \{(x_1),\ldots,(x_\ell)\}$, $\ell \in N$ each of them represented by a n feature vector $x_i = (x_1, x_2, \ldots, x_n), x \in \mathfrak{R}^n$. The examples in such dataset represent samples of the environment;

2. an input/output mapping function $f_{w,b}$ whose parameters are to be calibrated by the learning algorithm;

3. a set of ℓ binary/real labels $\{(y_i)\}$ where $\ell \in N, y \in \{+1,-1\}$ or $y \in \Re$. The labels represent the desired result when applying mapping $f_{w,b}$ to the sample x_i such as $y_i = f_{w,b}(x_i)$. We need the labels in order to validate the goodness of learned parameters of function $f_{w,b}$.

The initial network architecture consists only in a collection of not initialized blocks of computational units, like neurons (preliminary Support Vectors) without association of weights (alpha value). Even when not trained, when applied to given dataset this architecture might produce positive results, but this is an unlikely event resulting from a random guess. The choice of an appropriate learning strategy like Error Back Propagation (Maximal Margin Hyperplane) leads the random guess to a more effective strategy, by modifying the weights associated to each neuron (alpha associated to each vector) and adding hidden layers (support vectors) to the architecture.

In doing so, the learning algorithm tries to improve the mapping function according to some predefined criterion. In particular, a loss function allows the algorithm to choose among feasible configurations driven by the information embedded in the dataset. The loss function estimates numerically the goodness of the current parameter configuration.

When needed, it is also possible to exploit the loss value in order to compute the statistical significance of the training performed. In order to evaluate the loss function, the learning algorithm exploits the availability of labels for the train samples. The labels represent the desired output of the mapping function when applied to associated sample. We call this process supervised learning, stressing that an external supervisor (who made the labels) implicitly leads the learning process.

However, when the dataset is huge, it can be difficult for the learning process to converge quickly to a stable state. In this case, it is useful to introduce some mechanism for enhancing the convergence rate. One methodology, namely feature reduction, consists in manipulating the metric of the feature space trying to maximize the measure of similarity of neighboring objects with the same label.

When unseen samples of the same environment are correctly mapped by the trained function, the learning procedure is said to have reached a good generalization. Generalization is the key feature of automatic learning. The generalization ability guarantees that the trained model of knowledge can succeed in extracting useful information about the environment, as represented by the input data. On the other side, the learning algorithm must take care to avoid overfitting, which is at the opposite of generalization, and which can be imagined as implementing a fixed look-up table, one

entry for each input example. Each example is then correctly learned, but there is no way to process unseen example. Hence, each new example is mapped onto one of the previously seen ones.

By adopting the substitution of terminology presented in table 6.1, the subsymbolic learning strategies surprisingly become very close to the model we presented before. In particular, the network architecture (and the one inducted by SVs) fits very well the basic schema. The optimization strategy of Error Back Propagation (Lagrangian Optimization) resembles the process involved in creating new schemata by statistically chaining existing schemata. In addition, the generalization of ML recalls very closely Piaget's assimilation, as far as it represents the mechanism able to adapt funded knowledge to unknown environment.

table 6.1 – Comparison of terminology among psychogenetics (a), subsymbolic Machine Learning (b) and neuroscience (c)

Psychogenetics	**s.s. Machine Learning**	**Neuroscience**
Action	$f_{w,b}$ (mapping)	Linear combination of units
Adaptation	Learning	Tuning units
Assimilation	Generalization	Generalization across huge variation
Basic schema	ANN architecture	Neural Network architecture
Context	$S = \{(x_1),\dots,(x_\ell)\}, \quad \ell \in N$	Stimuli
Equilibration	Supervision	Feedback
Exercise of reflexes	Train algorithm	Gaussian-like tuning (normalization) plus soft max operation
Item	$x_i = (x_1, x_2,\dots,x_n), x \in \Re^n$	Firing rates
Neighborhood increment	Overfitting	Look-up table memory
Neighborhood mechanism	Space metric (kernel)	Anisotropic sampling of fovea
Partial Schema	Trained ANN architecture	Trained Neural Network architecture
Reliability	Confidence	Specificity of neurons
Result of action	$y_i = f_{w,b}(x_i)$	Visual recognition or motor act
Result item	y_i	Visual identification or muscle activation sequences
Satisfaction	Loss function	NOT PRESENT at biological level
Schemata verification	Validation	Normalizing the inputs

7. A biological interpretation from neuroscience

The psychogenetic model presented in this paper shares common results also with biological experiments that neuroscientists are performing in order to discover how biological learning works in vision and motor control. Current researches on monkey brains are revealing that a deep binding exists among different sensorial subsystems, such as vision and motor control. Further, a common biological framework may be involved, at the brain level, for controlling all human senses. Unexpectedly, the biological point of view seems to corroborate a possible multidisciplinary reframing of our model. Table 6.1 summarizes our belief for a possible substitution of terminology to revisit both our model and machine learning formalization from a biological perspective. Interestingly, in the biological framework we did not find any interpretation for the mechanism of satisfaction. Perhaps, the satisfaction is present only in cognitive high-level control systems, i.e., in the mind and not in the brain, and it is missing in biological low-level subsystems like vision and motor control. One could argue that the minimization of energy is the satisfactory stimulus involved in those subsystems, but analysis at such physical micro-level is out of the scope of biology.

In addition, some researches (Poggio, 2004) suggest that human intelligence is not only a complex look-up table of labeled images or a phone-directory-like list of motor acts. Furthermore, the learning process is more than building a memory and not only a right chaining of sequences of muscle activation (or Piaget's schemata).

In the light of these considerations, the need of satisfaction may be the cognitive origin that induces biological subsystems to overcome the drawbacks of the static look-up table model. To this purpose, the generalization mechanism may be the effective tool developed by Nature to sustain a wide range of behaviors across huge variations of environment while keeping the learning simple. Statistical Learning Theory could explain this insight.

8. Conclusions

This paper reports about the results of a first phase of a research aimed at studying the possibility of using psychogenetics, which is a fundamental theoretic learning model of cognitive psychology, as a basis for designing a new approach to unsupervised learning. In particular, we referred here to the theory of child development during its first 4 months of life according to J.Piaget.

The resulting algorithm has been used in a context where we simulate the behavior of an agent in an unknown environment, with a task to attain and some possible obstacles. The agent learns to reach its goal by means of exercising the available primitive actions (move an hand or the focus

region of an eye ahead, behind, left or right, open or close the hand). It observes the effect of its actions on the environment and on this basis it constructs schemata, i.e., behavioral rules which denote predictions on what will happen if a given action will be made. The set of the schemata so constructed will be the agent knowledge base. Schemata are activated on the basis of their expected satisfaction, as resulting from their previous usage, and of their relative reliability.

Our model proved able to construct reliable knowledge bases for the agent which can reach, in several different settings, the given goal.

Current computational results can be furthermore considered a feasibility test for a more general project for the design of a cognitively well-founded learning paradigm. Future development are thus many, including improved mechanism for dealing with obstacles and the study of models for the successive stages of child sensorimotor intelligence development according to Piaget.

Further, we discussed the possible parallels between our model and subsymbolic machine learning and neuroscience. It is worth noting that these relationships are only a very preliminary work to propose an alternative key for reading our computational model of psychogenetics. Nevertheless, we deem these similarities to be intriguing.

Dedication

The first author of this work has been a Ph.D. student of Marco Somalvico. The unfortunate event which brought to this special issue fostered me to complete this work, which I think is a good example of the spirit of scientific curiosity that animated Marco and his group. Despite a doctorate dedicated to combinatorial optimization in fact I clearly remember him encouraging me to pursue these ideas, at the time only sketched. I hope this will help also others to (rarely) divert from their mainstream search following an intriguing momentary inspiration.

<div align="center">

References

</div>

Barto, A.G. (1992). *Reinforcement Learning and Adaptive Critic Methods*, In D.A. White, D.A. Sofge, (Eds.), *Handbook of Intelligent Control*, 469-491, Van Nostrand Reihold, New York.

Bishop, C. M. (1995). Neural Networks for Pattern Recognition. Clarendon Press, Oxford.

Chomsky, N. (1959). *Verbal Behaviorism*, Language V. 35, 26-58.

Drescher, G.L., (1991). *Made-Up Minds: A Constructivist Approach to Artificial Intelligence*, Massachusetts Institute of Technology Press.

Hayes, (1978). *The Naive Physics Manifesto*, Artificial Intelligence, 41.

Holland, J.H. (1975). *Adaptation in Natural and Artificial Systems*, Ann Arbor: The University of Michigan Press.

Holland, J.H., et al. (1986) *Induction: Processes of Inference, Learning, and Discovery*, Cambridge, MA, MIT Press.

Holland, J.H. (1986) *Escaping Brittleness: The possibilities of general-purpose learning algorithms applied to parallel rule-based systems*. In: R.S. Michalski, J.G. Carbonell & T.M. Mitchell (eds),Machine Learning: An Artificial Intelligence approach, Vol II, 593-623, Los Altos, CA: Morgan Kaufman.

Kecman, V. (2001). Learning and Soft Computing, Support Vector machines, Neural Networks and Fuzzy Logic Models, The MIT Press, Cambridge, MA, the book's web site is: http://www.support-vector.ws.

Maltoni, R. (1994). *Algoritmi di Apprendimento Automatico per l'Individuazione di Strategie Decisionali Ottime*, Tesi di Laurea, Università di Bologna, Dipartimento di Scienze dell'Informazione.

Maniezzo V., Navarra A. (1996), *A Psychogenetic Model for Learning Sensorimotor Sequences*, Proc. First European Workshop on Cognitive Modeling, Fachbereich 13 - Informatik, Technische Universität Berlin, pag.76-81.

McDonald, M.A.F. and Hingston, P. (1997). *Discounted Reinforcement Learning Does Not Scale*, Journal of Computational Intelligence, 13 (1), 126-143.

Minsky, M.L. (1954). *Theory of Neural-Analog Reinforcement Systems and Its Applications to the Brain-Model Problem*, PhD Thesis, Princeton University.

Minsky, M. L. (1975). A framework for representing knowledge. In P. H. Winston (Ed.), The Psychology of Computer Vision (pp. 211 –277). New York: McGraw Hill.

Nosofsky, R.M. (1992). *Attention, Similarity and the Identification-Categorization Relationship*, Journal of Experimental Psychology: General, 115, 39-57.

Piaget, J. (1936). *La Naissance de L'intelligence chez l'Enfant*, Neuchâtel: Delachaux et Niestlé.

Piaget, J. (1937). *La Construction Du Réel Chez L'enfant*, Neuchâtel: Delachaux et Niestlé.

Piaget, J. (1945). *La Formation Du Symbole Chez L'enfant*, Neuchâtel: Delachaux et Niestlé.

Piaget, J. (1947). *La Psychologie De L'intelligence*, Paris: Colin..

Piaget, J. (1964). *Six Etudes De Psychologie*, Gonthier. (Trad. Ital. *Lo Sviluppo Mentale del Bambino*, Giulio Einaudi, Torino, 1967):

Piaget, J. (1975). *L'équilibration Des Structures Cognitives: Problème Central Du Développement*, Paris: Universitaires De France.

Piaget, J., Inhelder B. (1963). *Traité de Psychologie Expérimentale* Cap. XXXIV, T.VII, *Les Opérations Intellectuelles et leur Développement*.

Piaget, J., Inhelder B. (1967). *La Genèse des Structures Logiques Élémentaires*, Neuchâtel: Delachaux et Niestlé.

Poggio, T. and Bizzi, E. (2004). Generalization in Vision and Motor Control, Nature, Vol. 431, 768-774.

Quillian, M.R. (1968). Semantic memory. In, Minsky, M.L. (ed.). Semantic information processing. Cambridge, Mass.: MIT Press, 216-270.

Rumelhart, D.E., Hinton, G.E., Williams, R.J. (1986). *Learning Internal Representations by Error Propagation*, In Rumelhart, D.E., McClelland, J.L. (Eds.) *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Volume I, Chapter 8, The Mit Press.

Schank, R.C., Abelson, R.P. (1977). *Script, Plans, Goals and Understanding: An Inquiry into Human Knowledge Structures*, Lawrence Erlbaum Associates.

Shalizi, C. R., Shalizi, K. L. and Crutchfield, J. P. (2002). "Pattern Discovery in Time Series, Part I: Theory, Algorithm, Analysis, and Convergence", Santa Fe Institute Working Paper 02-10-060.

Simon, H.A. (1972). *The Theory of Problem Solving*, Information Processing 71.

Sutton, R.S. (1988). *Learning to Predict by the Methods of Temporal Difference*, Machine Learning, 3, 9-44.

Vapnik, V. (1995). The Nature of Statistical Learning Theory, Springer Verlag Inc, New York, NY.

Watkins, C.J.C.H. (1989). *Learning From Delayed Rewards*, PhD Thesis, King's College, Cambridge.

Watkins, C.J.C.H., Dayan, P. (1992). *Technical Note: Q-Learning*, Machine Learning, 8, 279-292.