

Modeling the Nervous System by Evolutionary Connectionism

Neri Accornero and Marco Capozza

Neuroscience Department, Sapienza University, Viale dell'Università 30, 00185, Rome, Italy
neri.accornero@gmail.com, marco.capozza@gmail.com

Keywords: Artificial Neural Network, Connectionism, Simulation, Evolution, Genetic Algorithms, Nervous System

Abstract: Artificial neural networks (ANN) are excellent biological-like models to simulate nervous system functions. Their spontaneous biological likeness can be further enhanced by letting biological-like procedures, such as evolutionary selection, shape the investigated ANN models. This makes the investigator duties shift from designing the networks to designing the environment where the networks behave, so reducing the impact by possible investigator preconceptions on the models. In this paper we present a few examples about modeling nervous system functions with ANN subjected to evolutionary selection by genetic algorithms. This methodology, called Evolutionary Connectionism, proved feasible and profitable in studying both sensorimotor and cognitive nervous functions through simulation.

1 INTRODUCTION

Artificial neural networks (ANN) are the best model available so far to simulate and understand biological nervous systems, inasmuch as they spontaneously and effectively tend to reproduce essential nervous functions and features, even if in a necessarily simplified way. In this paper we present a summary of some ANN models we developed, to show how powerful this methodology is in simulating nervous system functions, from simple motor reflex up to cognitive behaviour (see Accornero and Capozza, 2009 for a more thorough description on these and other models).

ANN intrinsic biological plausibility can be further increased by letting the modelled networks get shaped by genetic algorithms (Goldberg, 1989; Schaffer, Whitley and Eshelman, 1992) selecting through Darwinian evolution ANN populations adapted to perform required tasks ('genotypic learning'). This makes the investigator duties shift from designing the networks to designing the environment where the networks behave, so reducing the impact of possible investigator preconceptions on the model. This methodology, called Genetic Connectionism (Chalmers, 1990) or Evolutionary Connectionism (Calabretta and Parisi, 2005) and its models called Evolutionary Artificial Neural Networks (Yao, 1993, 1995, 1999), proved to be powerful and effective as well as compatible with actual biological reality.

The first two models we present here with this methodology produce, in an automatic and unsupervised way, ANN populations adapted to perform tasks needed to 'live' in their artificial environment. The first model (ALIFE) simulates a simple ecosystem where agents develop a motor strategy to achieve energy sources (food) in order to live and reproduce. In the second model (GAZE), an ANN population evolves that exhibits a human-like head-eye motor control to pursuit visual targets.

The third model (ARM) does not make use of genotypic learning, inasmuch as its purpose is to introduce a special, predesigned sensorimotor ANN model, which we called the 'triplet-net' model and we consider pivotal to model intentional movement and even operative thinking. This simulation concerns intentional movement, and it supports the 'ideomotor principle' (IMP) (James 1890, Kiesel and Hoffmann 2004, Stock and Stock 2004), i.e. the notion positing that voluntary actions are cognitively represented by their sensory effects and produced by anticipation (desire) of those effects.

In the fourth model (ACON) we extend the triplet-net model from the sensorimotor domain to the cognitive domain, to show once again by evolutionary connectionism how imagination spontaneously evolves as a way to reach goals. This can be interpreted as a beginning of 'thinking' in an artificial system, and it provides us

with a better understanding for mental functions critical to consciousness, such as recalling memories, foreseeing and reasoning.

2 'ALIFE' (ARTIFICIAL LIFE) SIMULATION

2.1 Introduction

This was our first simulation with evolutionary connectionism methods, dating back to 1994, DOS and VGA age. With it we aimed to preliminarily test the efficacy of genetic algorithms applied to ANN populations, before venturing to more demanding experiments.

2.2 Methods

This simulation consisted of an environment, 'world', with 'food' packets and a population of 'agents'. These were small neural networks, subjected to Darwinian evolution, which survived and reproduced by developing sensory and motor skills useful to achieve food.

The simulation world was a two-dimensional grid with 640x480 cells, a folded toroid. In this world agents consumed energy to stay alive and move, and each of them could replenish its energy by 'eating' food. There was no limit to the number of agents that could coexist in a same cell.

Time was discretized and measured in turns. Agents acted once in each turn, where 'acting' meant: receiving visual information, activating the neural network units, moving, eating food i.e. gaining energy, and possibly interbreeding, reproducing, or dying. Subsequent agent generations might coexist and overlap.

Food packets reflected 'light' diffused in the environment, so that agents with visual ability might perceive them, and agents with motor ability could reach them. All food packets contained a same energy amount. A food packet got 'eaten' when an agent came close to it within a given distance. A number of food packets, fixed or cyclically variable in time (seasonal cycles) at investigator's choice, was released into the environment each turn. Every new packet was introduced in a random cell within a predetermined maximum distance from a random other packet already present. Since both maximum horizontal and vertical distance were decidable by the investigator and might differ, the investigator was able to affect the shape and orientation for food clusters (disk clusters, vertical or horizontal bands, etc.). The total amount of food packets in the environment changed each turn, being affected by agent skill to reach them and seasonal pattern.

Agents (max. 200) were constituted by a variable number of neural units (cells) ranging from 1 to 12, arranged as an annulus ('body'), so that all agents had a circular shape with the same size but a possibly different cell number. Any cell might evolve visual ability, motor ability, both, or none; cells with neither visual nor motor ability actually acted as hidden neural units (Rumelhart, Hinton and McClelland, 1986; McClelland and Rumelhart, 1986). Cells mutually connected through connections. Cell abilities and connection weights were selected by evolution.

Sensory cells were activated by light reflected from food, with activation values depending on light intensity (decreasing with distance) and cell visual sensitivity. The area 'seen' by each cell was a circle sector whose orientation depended on the cell position within the agent body. This visual system had no 'retina', just simple directional light sensors, although several sensors could be placed side by side like insect ommatidia.

Each motor cell generated a force vector tangent to the agent body contour with strength depending on cell motor ability and activation. Rotation and translation resulted from the overall composition of all motor vectors on the agent body. Since one single vector could only make an agent rotate, in order to translate agents must activate at least two motor cells at the same time in a coordinated manner. Agent energy consumption to move depended on its total number of motor cells, their motor ability, and their activation levels. Agents consumed energy not only to move but also to stay alive (metabolism).

All agents were initially provided with the same energy content. If an agent energy dropped to zero due to metabolism and movement consumption with insufficient food intake, that agent died and was removed from the environment. If, on the contrary, food eating increased the agent energy over a given threshold, then the agent was reproduced and a mutated copy of it was released into the environment. There was no fitness score other than agent energy levels, which affected 'real-time' (simulation time, of course) agent lifespan and replication. All features constituting an agent (its number of cells, cell position in its body, cell visual and motor ability, and cell connection weight) were coded as a 0-and-1 string in the computer memory. This string represented the agent 'genome'. When an agent reproduced, a new copy of its genome was created with some

bit mutated, and a new agent was generated from the new genome. When two agents became closer within a given distance they interbred, meaning that they exchanged genome string segments ('genetic crossover'). All consequences of this genome exchange were immediately applied to their body structures, similarly to DNA exchanges occurring in bacteria.

The simulation computer program allowed the investigator to determine online all parameters involved in what we described above. The program also allowed viewing and analysing population performance through three different displays: a 'history' display (Fig. 1 left side), reporting in color code the number of agents along time, their overall ability to achieve food, their prevalent motor strategies (rotation and translation amplitude), and food availability; a 'world' display, where the entire world grid was displayed real-time, with agents shown as moving colored dots (color depending on the agent energy reserve), and food packets as small motionless white dots; and an 'individual' display (Fig. 1 right side), where a single agent selected by the investigator was shown real-time along with its body, cells, energy content, visual perceptions and motor actions.

We conducted several simulations using different parameter values. Simulations ended at the investigator's choice, usually after 6-7000 turns.



Figure 1: Historical (left) and individual (right) display offered by the ALIFE simulation program.

2.3 Results

Our aim to find the emergence of a working 'survival behavior' in the simulated populations was achieved almost invariably, even with different 'world' parameters. In all simulations an agent population with sensorimotor reflexes invariably evolved. Agent bodies were composed by sensory cells and motor cells, often with two or three sensory cells grouped at a body pole (which can be regarded as the rostral pole), and two, three, or four motor cells usually located on the left and right body sides in order to obtain a forward movement and the ability to steer.

2.4 Conclusions

This preliminary study showed that even simple genetic algorithms were highly effective in producing ANN adapted to a designed environment, similarly to biological evolution for real nervous systems. It also highlighted that the similarity extent between artificial ecosystems and the real world heavily depended on realism in simulating real physics laws.

3 ‘GAZE’ SIMULATION

3.1 Introduction

This study aimed to verify whether genetic algorithms could assemble ANN able to control visual reflex movements in a simulated head-eye model. This head-eye model, even if simplified, was simulated more realistically than agent bodies in the ALIFE experiment, while the genetic algorithm was more abstract and formal than the simple ‘eat food or die’ principle used in ALIFE.

3.2 Methods

This simulation concerned a simplified two-dimensional head-eye model driven by an ANN to keep the eyes gazing at a moving target. The head-eye system (Fig. 2) consisted of a head rotating on the horizontal plane due to actions by two antagonist muscles simulating the left and right sternocleidomastoid muscles, and two eyes also rotating on the horizontal plane due to actions by two pairs of antagonist muscles (one pair each eye) simulating the medial and lateral eye muscles. Muscles were simulated by Hill spring model (Shadmehr and Arbib, 1992), with their relative length and insertion points acceptably approximating the real ones.

A horizontal bar-shaped target randomly moved with different velocities on a 430 x 888 pixel horizontal plane in front of the eyes, and it sometimes also emitted a virtual sound. The eyes perceived the target through two linear retinas, each one with a fovea. Target sounds were perceived by two ears placed on the left and right head side.

The ANN (Fig. 3) received visual input from the retinas, acoustic input from the ears, proprioceptive inputs on head and eye directions, and proprioceptive input on muscle tensions, for a total of 29 input units. From these inputs the ANN generated output commands for the head and eye muscles (6 output units). Besides input and output units, the ANN also included a varying number of hidden units ranging from 0 to 64, that received connections from the input units and other hidden units, and sent connections to the output units and other hidden units. The input units were simple linear units; all other units were classic sigmoid units (Rumelhart, Hinton and McClelland, 1986), having an analogic output ranging from 0 to 1 and equipped with modifiable learning bias. All features of any ANN were coded as a 0-and-1 string representing that ANN ‘genome’.

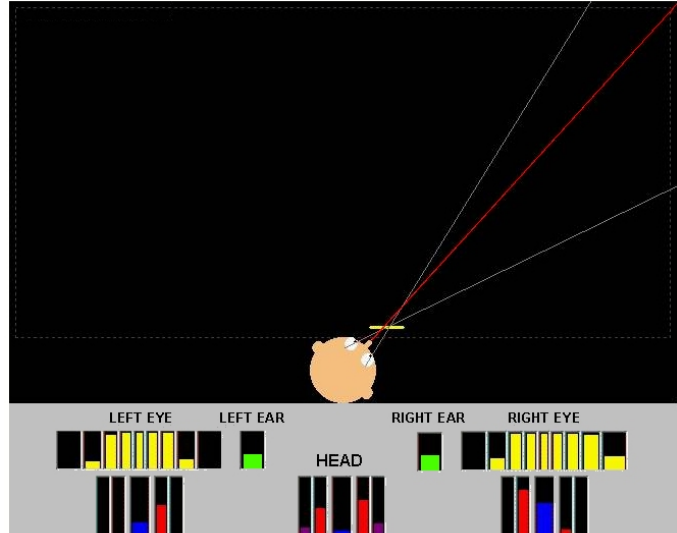


Figure 2: Head-eye system for the GAZE model. The coloured bars show activation levels of agent sensors and effectors. Yellow: visual retina input; green: acoustic input; blue: head and eye direction proprioception; purple: muscle tension proprioception; red: muscle activation.

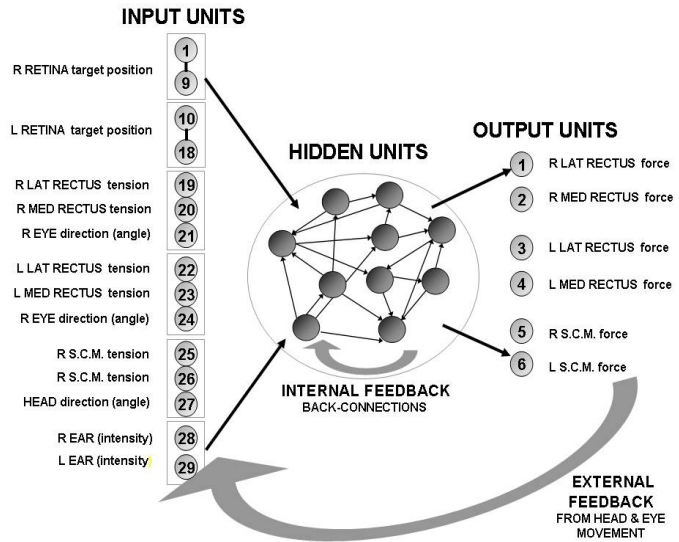


Figure 3: Artificial neural network for the GAZE model. Not all connections are depicted; actually, all input units send connections to all hidden units, and all hidden units to all other hidden units and to output units. R = right; L = left; LAT = lateral; MED = medial; S.C.M. = sternocleidomastoid muscle.

Simulation time was discretized and measured in turns. In each turn the target moved once and an agent acted once, where ‘acting’ meant: receiving visual information, activating the neural network units, and moving the head and eyes.

We set up 100 agents slightly different in their genomes and hence in their neural networks. When the simulation began, each agent was provided with a random number of hidden units ranging from 8 to 32, and the connection weights and the unit biases were initialized with random values. Then every agent was tested about its ability to keep gazing at the moving target during 10000 consecutive turns representing the agent lifespan. Each agent was tested individually; agents did not interact as they did in ALIFE. Three error values were measured in each turn as the angular distance between the direction of the target with respect to the agent and the direction where the agent head and eyes pointed to. Those values were recorded and added into three running (mis)fitness scores (one for the head and two for the eyes) for that agent. Once all 100 agents had been tested for 10000 turns each, a generation was ended, and the agents were compared each other on their three fitness scores singularly and ranked on their overall number of winning results (a non-parametric fitness evaluation). Agent genomes were then either reproduced into the next generation or discharged according with their rank (two offspring from the best agent, no offspring from the worst). Point mutation and genetic crossover were randomly applied to the offspring, with bit mutation rate = 0.0001 and crossover rate = 0.5.

In this simulation subsequent generations did not overlap. The length of the bar-shaped moving target randomly changed across generations. Target shape and trajectory were the same for all agents in a generation. Simulation ended when the mean angular error of the best agent in a generation dropped below 2 degrees for both the head and the eyes. At this point the best agent underwent specific tests with the target moving along pre-defined, non-random trajectories with fast and slow movements. These additional tests were also repeated with the agent head locked so that the agent could only move its eyes to keep gazing at the moving target.

3.3 Results

The designed system proved able to develop the desired motor control. In about 4000 generations the agents became able to keep their head and eyes gazing at the target with a mean error less than 5 degrees. Specific tests (Fig. 4) showed that the system could generate rapid conjugated (binocular foveation) and unconjugated (vergence) movements, and fast

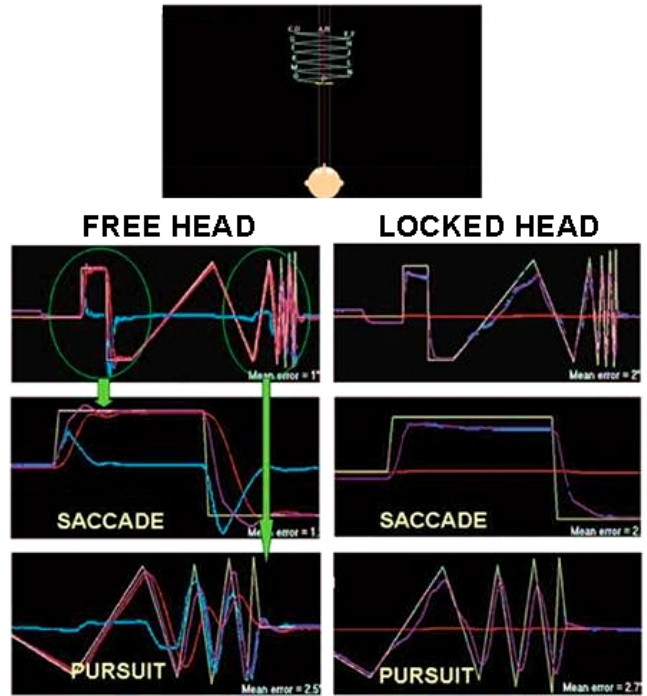


Figure 4: Traces of GAZE fast (saccades) and slow (pursuit) movements following fast and slow target displacements, with the head free to rotate (left column) or locked (right column). Yellow trace = target; red = head; blue = eyes; purple = gaze.

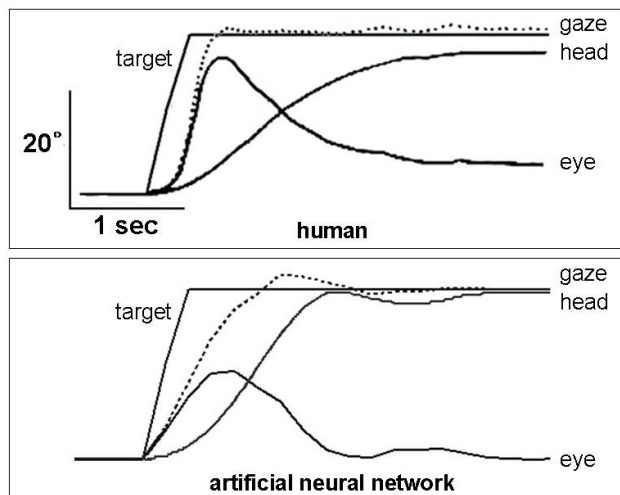


Figure 5: Comparison between human and GAZE movement patterns following a fast target displacement.

(saccades) and slow (smooth pursuit) movements following target position and velocity. When the same tests were repeated with the agent head locked (Fig.4 right), the system compensated efficaciously with ample eye movements in order to track the target, just like biological systems do.

We also observed (Fig. 5) that, after fast target movements, the eyes moved so that gaze rapidly approximated the new target position; shortly after, the head began to move, while the eyes made a compensatory movement backward into primary position, so that the gaze stayed fixed onto the target. This movement pattern closely resembled those exhibited by biological nervous systems in humans and monkeys (Bizzi, 1974). On the contrary, when the target movement was slow enough so that the head could follow, no eye movements were observed, which was similar once again to what happens in biological systems.

3.4 Conclusions

This study confirmed the effectiveness of genetic algorithms applied to ANN, and showed that applying realism at an even limited extent in simulating the biophysical reality of actual biological systems may result in the appearance of emergent behaviors closely resembling the real ones. Our results from the locked head test also demonstrated that these emergent properties even include the ability to produce appropriate behaviors in conditions never previously experimented by the population in evolution. This ability to generalize on limited experience is a striking feature of ANN models and biological nervous systems, and it seems even more likely to appear in ANN simulations with ANN models shaped by biological-like processes such as evolutionary selection.

4 ‘ARM’ SIMULATION

4.1 Introduction

The ideomotor principle (IMP) (James 1890, Kiesel and Hoffmann 2004, Stock and Stock 2004) claims that the nervous system initiates voluntary actions by anticipating their typical sensory consequences. In this study we present an unsupervised ANN system that is as simple and basic as possible and learns to move a three-joint arm in a two-dimensional workspace using this principle. We examine its main features and compare them with those of human motor learning.

4.2 Methods

4.2.1 Model Design

This model does not make use of evolutionary connectionism and genotypic learning. Instead it makes use of phenotypic learning, because its purpose is to introduce a special, *predesigned* sensorimotor ANN model, which we called the ‘triplet-net’ model and we consider pivotal to model motor and behavior learning, intentional movement, and even operative thinking. Therefore in this simulation ANN are not arranged in an evolving population, and there are no subsequent ANN offsprings, but there is just one single ANN instead learning to move a limb in an unsupervised fashion on its own motor experience. This ‘triplet-net’ model will be then brought back to genotypic learning and evolutionary connectionism in the next (‘ACON’) model.

Our simulation consisted of an ANN controlling a three-joint simulated limb moving in a two-dimensional plane (Fig. 6). The network received sensory information on the limb position on its input units, and sent limb commands from its output units. Three vectors defined each limb movement: the initial sensory state S1 (before the movement); the final sensory state S2 (after the movement); and the neuromuscular activations M needed to pass from the initial to the final state. The S1 and S2 vectors were given to the ANN input units, and the ANN had to compute the M vector on its output units. Each S1 and S2 vector element activated one ANN input unit, and each ANN output unit generated one M element, so that each S1 and S2 element had a one-to-one mapping with the activation of one input unit, and each M element with the output of one output unit. Therefore we will speak of S1, S2 and M elements also as S1, S2 and M units.

Because IMP states that intentional limb movements depend on anticipation of their sensory effects, the ANN input units receiving after each movement sensory information on the *final* limb state (S2 units) also received, before each movement, motor commands from a component outside the network that established

where the moving hand should be positioned and therefore acted as ‘Motor Will’. Commands from it to the ANN consisted of sensory representations of the desired final hand position, coded as visuospatial coordinates in agreement with the observation that motion planning in human takes place in the visually perceived space (Flanagan and Rao, 1995; Shadmehr, 2005). Unlike S2 units, the ANN input units receiving sensory information on the *initial* limb state (S1 units) did not receive motor commands, they only received ‘proprioceptive’ sensory information from the limb joint angles.

Given that velocity information was not indispensable to the key IMP mechanism as long as the limb was assumed to start from still and end still, we decided to give the ANN only sensory information about limb position (joint angles and spatial hand position).

4.2.2 Limb

The limb was designed to represent a simplified model of the human right arm comprising three segments, ‘arm’, ‘forearm’ and ‘hand’ articulated with three joints ‘shoulder’, ‘elbow’ and ‘wrist’, with the shoulder situated in a fixed point in space, and the hand able to move freely in the reachable space

4.2.3 Neural Network

The ANN was a two-layer neural network comprising 5 input units and 6 output units, fully connected with anterograde connections from input to output. There were no hidden units. The first three input units (S1 units) received ‘proprioceptive’ information on the opening angle for each of the three joints, normalized between -1 and 1. Before any movements these units received their information from the sensory pathways, and they kept that information alive up to the learning phase occurring after the movement (see below, 4.2.6). The last two input units (S2 units) received from the sensory pathways ‘visuospatial’ information on the hand position, encoded in polar coordinates. Immediately before a movement (movement phase, 4.2.5 step 2) this hand visuospatial information on S2 units was overwritten by Motor Will with activations corresponding to a new desired hand position (a motor command).

4.2.4 Simulation Flow

When the simulation began, the connection weights and the output unit biases were initialized with random values ranging from -0.25 to +0.25. The arm was positioned with all the joints partly opened. After the initialization stage, the simulation proceeded in turns, each turn comprising the two phases, movement and learning, each comprising three steps.

4.2.5 Movement

1. The input units received sensory information from the arm: S1 units received the angles from the three joints, and S2 units the actual spatial location of the hand.
2. Motor Will overwrote S2 input unit activations with activations corresponding to a random desired hand position.
3. The input units activated the output units, and the joint opening angles therefore changed. The actual output values were recorded for use in the ensuing learning phase, during which they yielded the desired output, *target*

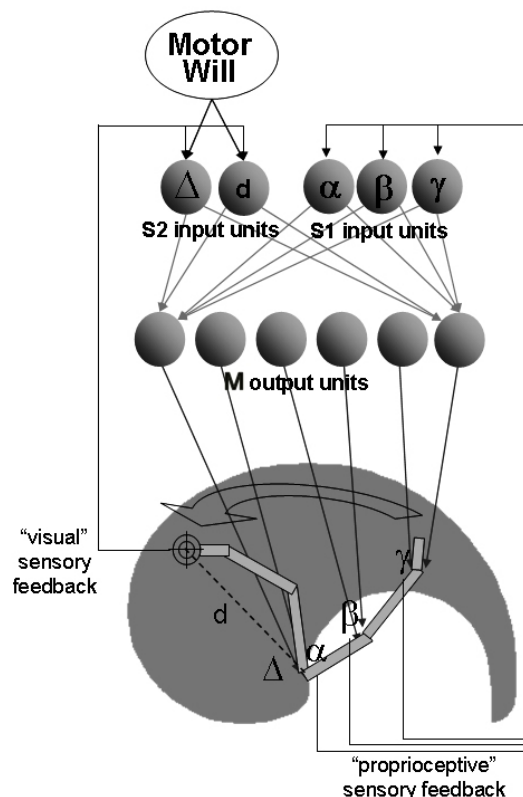


Figure 6: General architecture for the ARM model. The artificial neural network (ANN) controls a 3-joint limb moving in a two-dimensional plane. The ANN receives sensory feedback information on the limb and motor commands from a Motor Will.

activations. The spatial error in pixels between the desired (target) and actual hand position is recorded to evaluate the ANN performance and not to assess motor learning.

4.2.6 Learning

1. The sensory pathways conveyed to the S2 units information on the new hand position.
2. The input units activated the output units again, this time using the new activation values on the S2 units corresponding to the hand position actually reached. These new outputs did not cause arm movements, they served only for learning. These new outputs were the ones the network would produce if the desired movement were actually the movement achieved in movement phase 4.2.5 step 3. The difference between the current outputs and the outputs recorded in that phase was the error to minimize during learning.
3. A standard delta rule (Rumelhart, Hinton and Williams, 1986) was applied to minimize the error vector calculated in the former step.

It should be noted that after the movement only S2 unit activations were updated with the new hand position, while S1 unit activations maintained the values corresponding to the joint angles before the movement. This was essential to the model functioning: at the beginning of the learning phase, S1 units must code the initial state before the movement, S2 units the final state after the movement, and M units the neuromuscular activations that caused the transition from the initial state to the final state. It was not even essential that S1 and S2 units encoded different sensory information (in our simulation, joint angles and hand position respectively): actually, *both* S1 and S2 units might encode *both* the joint angles and hand position. We made S1 and S2 units encode different information just to make the ANN task not too trivial. What was essential to the model was not the type of sensory information given to the S1 and S2 units, but that the learning phase would begin with the S1 units encoding *some* information on the arm state *before* the movement, the S2 units *after* the movement, and the M units the neuromuscular activations causing the movement. This was a triplet taken on experience on the real world, and it was what the ANN had to learn, so that whenever that same S2 final position would occur as a desired position on a same S1 initial state, the ANN would be able to generate those M activations bringing the arm from the S1 state to the S2 state.

4.2.7 Tests

Besides evaluating the ‘online’ spatial error after every movement (section 4.2.5 step 3), after every 5000 movements the program submitted the network to an ‘offline’ test entailing a predefined set of 588 target points (Fig. 7A) commanded by the Motor Will. During testing, the learning phase (section 4.2.6) was skipped. For each of the 588 points the position actually reached by the hand and the corresponding spatial error were recorded for later evaluation offline.

We conducted several simulations. In some simulations we introduced a sort of ‘sensory blind spot’, a wide circular area, covering up to 50% of the workspace and differentially positioned in the various simulations (Fig. 8A), where we skipped the learning phase when the hand ended up in this area.

In other trials, to assess whether learning depended on precise physical values inherent to the system, and to verify whether the controller system adapted to changes in the controlled system, we varied the sensory code used for hand position or the limb segment mass, right from the beginning or after advanced learning (30000 movements).

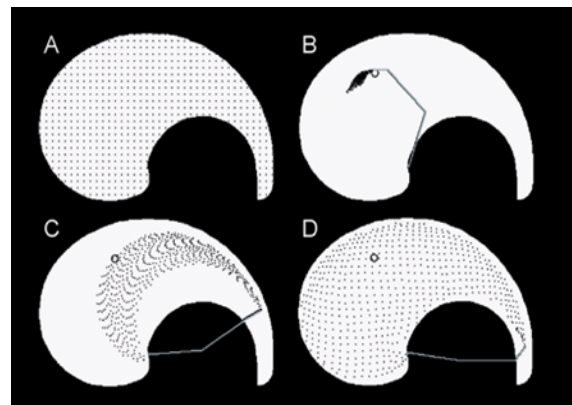


Figure 7: Progressive improvement in performance during the 588 test movements with motor experience. Small circle = hand starting point; black points = hand movement arrival point. A: target points; B: points effectively reached before learning; C: after 5000 random movements and D: after 30000 random movements.

4.3 Results

In all the simulations the tested ANN system improved from a mean spatial error of more than 150 pixels when simulation began to an error of less than 15 pixels after 10,000 movements (few tens of seconds on a modern pc) and fewer than 7 pixels after 30000 movements.

After 30000 movements, spatial error distributions showed that the system performed well over the whole workspace, except in the extreme tail in the drop-shaped area corresponding to extreme extension (Fig. 8B). The sensory blind spot had scarce influence on learning improvements (Fig. 8C, D). These results remained uninfluenced by the hand sensory code used, nor did they significantly suffer from mass changes in limb segments, before or after motor learning.

4.4 Discussion

The simplified ANN simulation, focusing on the basic IMP features insofar as motor commands and sensory feedback reach the same S2 input units, effectively learned to move the arm in the workspace. It learned acceptably well even when we varied influential experimental variables such as the sensory code used for hand position, the mass for the limb segments to move, and when the ANN was able or unable to receive sensory feedback about movements performed in the workspace (sensory blind spot).

Our IMP model reproduced with acceptable approximation some human motor learning properties, such as learning from experience, ability to work regardless of the specific body segment features, ability to adapt to changes in these features, and the fact that even randomly-generated movements contributed to learning (infantile motor babbling). Like the human motor learning system, our ANN underwent completely unsupervised learning. We never used external sample sets, the ANN itself generated learning examples from its random movements and errors. The difference between output unit activations in two different functional phases (movement and learning) was utilized as the learning error, and those values were completely and locally available to the net.

In this model *the movement learned is not the desired movement but the movement effectively done*. The system nevertheless succeeded in performing with reasonable precision even movements never previously done and those finishing in the sensory blind spot. This system ability evidently stems from an ANN's well-known ability to generalize (Caudill and Butler, 1992), a feature allowing our ANN to interpolate and extrapolate information from the movements done, thus filling in unexplored movements and forming the general sensorimotor map valid for all movements.

In this simplified model the ANN consists of three unit groups coding the initial limb sensory state, the final state, and the action causing the limb to pass from its initial to its final state, that we will henceforth call a *sensorimotor triplet*, or simply *triplet*. This triplet model can be extended from elementary movements to more complex behaviors thus unifying the various intentional movement scales under a single principle. It involves several triplet-networks, linked so that the output units for each preceding triplet-net also act as the S2 input units for the ensuing triplet-net ('chained triplet-net' model). The S1 and S2 input units can receive sensory information not only from the musculoskeletal system, but from the whole body and external environment. In this chain, the S2 input units on the first net receive the actions desired by Will (actions that are more abstract than the simple and concrete desire to bring the hand to a desired position), and the ensuing nets progressively increase the level of detail and concreteness for the actions needed to satisfy the desire. Finally, the final net (the net described in the basic model) generates the neuromuscular activations required to perform the selected action(s).

Essentially, we suggest that in the nervous system voluntary actions are triggered by formulating their end-effects as high-level sensory representations of the desired results. This model is consistent with increasing

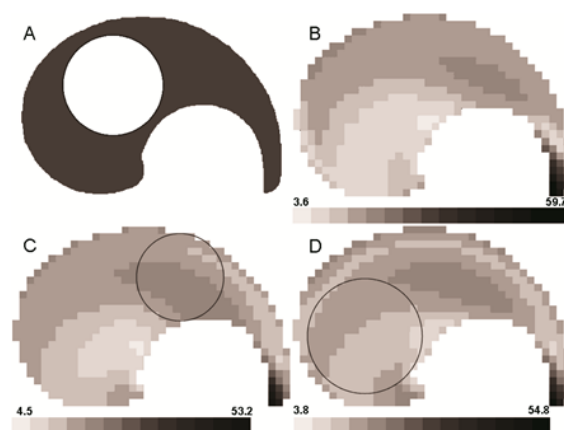


Figure 8: Spatial error distribution for the 588 test movements after 30000 random movements with and without 'sensory blind spot'. A: workspace area (dark grey area) with a generic blind spot (white disk); B: errors (in grey color code) without the sensory blind spot; C, D: with the sensory blind spot (black outline circle) in two different sizes and positions. Values are for spatial error in pixels.

evidence from motor research in primates and humans (reviews in Lebedev and Wise, 2002; Graziano, 2006; Cisek and Kalaska, 2010).

4.5 Conclusions

This unsupervised ANN simulation confirmed, as the IMP claims, that voluntary actions can be initiated by imagining (desiring) their sensory effects. IMP seems a valid model for understanding human sensorimotor mapping, intentional movement and motor learning. Detailing and extending the IMP in what we termed the ‘chained triplet-net’ model makes the IMP also helpful in explaining voluntary behavior besides elementary actions. We are going to add further touches to this picture with the next experiment.

5 ‘ACON’ (ARTIFICIAL CONSCIOUSNESS) SIMULATION

5.1 Introduction

In the ARM model, Motor Will generated random sensory-coded target points where the triplet-net would have to move the hand, and the triplet-net only performed sensorimotor translations needed to execute those motor commands. In this new simulation we extend the ARM model to show how a Motor Will network and a triplet-net together, when provided with a few additional elements and connections, acquire the ability to produce useful intentional behaviour once they are nestled in an environment promoting such a behaviour.

A straightforward definition for a voluntary action is ‘an action taken after foreseeing its effects’ (Parisi, 2006). The ability to foresee relies on memories from past experience. Hypothesizing that memories in brain are contained in the associative areas and can be recalled by other areas (namely the prefrontal areas), we decided to verify whether evolutionary selection can produce a neural network able to trigger recalls in an associative neural network memory and use them to foresee possible action outcomes, so as to ultimately take useful actions. Even though very simplified, this represents an innovative basic model for brain intentional behaviour.

5.2 Methods

5.2.1 Overall Model Design

This model consisted of a memory containing knowledge about the environment, and another component that used that knowledge to establish which actions to take and therefore acts as a Motor Will. The memory component was a triplet-net enriched with two-way connections between its three unit groups, so as to actually become an associative neural net memory, or auto-associator (McClelland and Rumelhart, 1986) rather than ‘just’ being a sensorimotor translator like it was in the ARM simulation. Also, Motor Will was not just a generator for random commands like it was in the ARM simulation, but a fully featured ANN sending the triplet-net meaningful activation sequences driving it to recall memories and make predictions. Since these Motor Will tasks recalled some of those called ‘executive functions’ in psychology, in this model we preferred to call it the *executive-net* rather than Motor Will.

We let the executive-net get shaped and made able to drive the triplet-net memory by evolution (genotypic learning), through genetic algorithms selecting numbers and features for the network units and connections. To this purpose we set up 100 agents, differing in their executive-net units and connections, performing actions in an environment where agents able to recall past experiences and foresee action outcomes would produce more beneficial behaviour, because there was always a better action, depending on which environment location the agent was. Any action was either affecting the agent fitness score, or making the agent change its location. Each agent was given a memory on the effects of each action in each location, as if the agent had explored the environment in advance. This memory was embedded in the triplet-net associative memory.

In order to verify our assumption that performing useful behaviours in the designed environment would require the ability by the executive-net to drive the triplet-net memory, we also carried out a control simulation with agents having no connections from the executive-net to the triplet-net (sort of a lobotomized population).

Lacking those connections, the executive-net could not drive the triplet-net, even if its functions to act into the environment were preserved.

5.2.2 Environment

The environment (Fig. 9) consisted of 48 simple ‘Y’ mazes with three rooms each. In each room only three action were possible: taking the left door, taking the right door, or staying still. In one room, that we call the ‘introductory room’, one door led to the ‘reward room’, the other door to the ‘penalty room’, with no change on the agent fitness score. In the reward room, taking one door resulted in getting a big fitness reward,

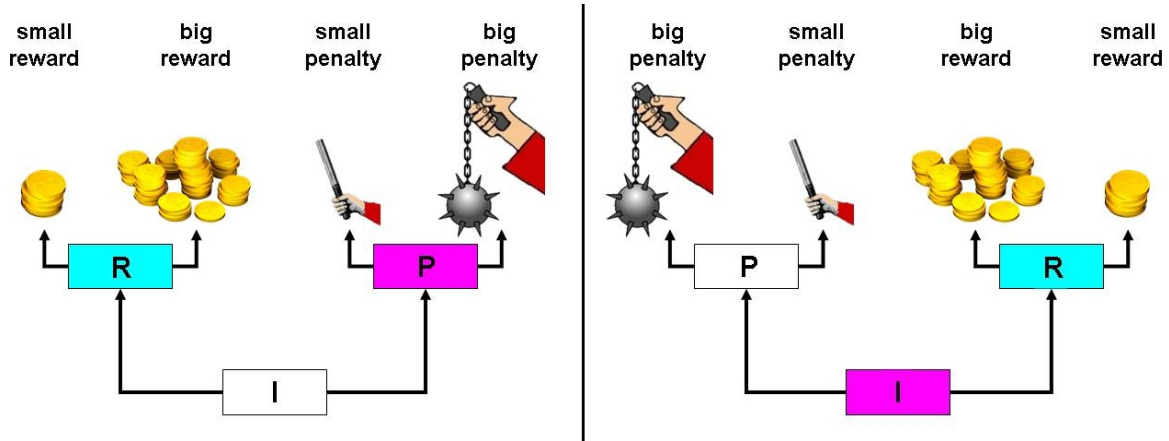


Figure 9: two mazes from the 48 constituting the environment. Each maze comprises three rooms with two doors each. R = reward room; P = penalty room; I = introductory room. Room colors and penalty/reward doors are exchanged in the 48 mazes so that all mazes differ from each other. Each agent is tested in all 48 mazes starting in every room.

the other door a small reward. Finally, in the penalty room one door resulted in getting a big penalty, the other door a small penalty. Reward and penalty doors would not make the agent change room, rather they would make the current maze test end.

Each of the three rooms was colored in a different color, say red white or blue. Each of the 48 mazes was characterized by a unique combination of room colors and door outcomes, and the 48 mazes covered all possible color and door combinations. Any agent could perceive the color of the room in which it was (see also section 5.2.4), but it got no information on whether that room was the introductory or reward or penalty room, until its executive-net would ask its triplet-net about it (by asking about the outcome of taking a door in that room).

5.2.3 Neural Network

The agent neural system (Fig. 10) consisted of an input layer, an

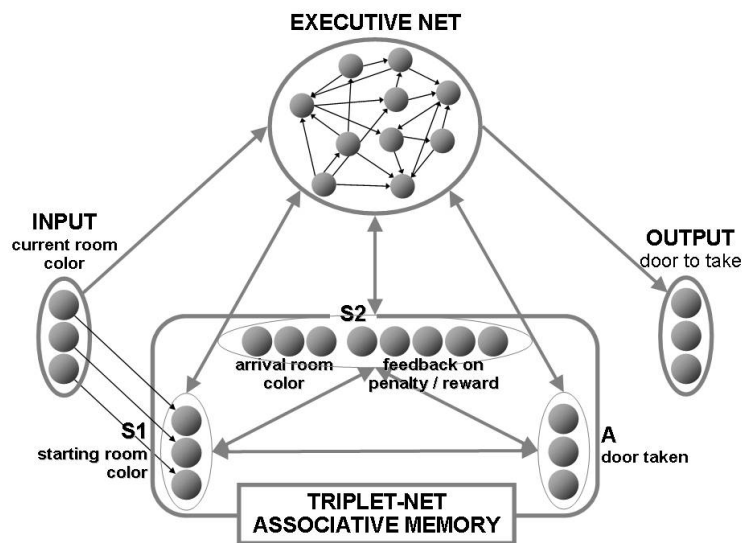


Figure 10: Neural system for the ACON model. Thick grey arrows between unit groups represent full connections between units (any unit in a group sending connections to all units in the other group).

output layer, a triplet-net and an executive-net.

The input layer consisted of three binary neural units encoding the color (blue, red or white) of the room where the agent was, with each unit encoding a possible color as its '1' state. These were simple linear units, just passing their activation values to the triplet-net and executive-net units they were connected to.

The output layer consisted of three analogic neural units encoding the action that the agent would actually perform: activating the first unit would result in the agent taking the left door, the second unit in taking the right door, and the third unit in staying still. The action actually performed was stochastically selected on the difference between the three unit activation values.

The triplet-net was similar to the ARM model, but in this model all its units were binary units, and there were two-way connections between all three unit groups so that the net could work as an associative memory. The three S1 units received information from the input units about the color of the room where the agent was. The S2 units encoded the color of the arrival room, i.e. the room where the agent would arrive because of taking an action (three units like the S1 and input units); they also encoded the fitness reward or penalty related to that action (five units: big reward, small reward, no change, small penalty, and big penalty), for a total of eight S2 units. The motor units (three units like the output units), that in this simulation we prefer to call the 'A' units (A for Action), encoded the action causing to pass from the starting room encoded in the three S1 units (which might differ from the actual room encoded in the input units, when their activations were generated by the executive-net) to the hypothetical arrival room encoded in the first three S2 units. This encoded hypothetical action might differ from the actual action encoded in the output units as commanded by the executive-net. In this model the triplet-net had no connections to the output units, only the executive-net could activate them.

In reality, in this simulation we did not actually implement the triplet-net as a fully featured ANN, but we actually *emulated* it, the same way as in the ARM model we did not actually implement Motor Will, but we rather used a random number generator to emulate it. In this simulation in place of triplet-net actual connections we used a table containing 0 and 1 values emulating the 0-and-1 vectors actually present in the environment, the same vectors that a real triplet-net would learn by experience in the environment. Anytime the executive-net would send the triplet-net an incomplete or wrong vector, the best matching actual vector was searched through the table and returned to the executive-net, thus emulating an associative neural network memory that would fill in or replace all missing or wrong values in the activation vector.

Finally, the executive-net was a non-layered neural network, a set of neural units each of which received connections from the input units, the triplet-net units, and other executive-net units, and sent connections to the output units, the triplet-net units, and other executive-net units. The number of units in the executive-net ranged from 24 to 48 and was determined by evolutionary selection, as also those unit biases, connections and connection weights.

5.2.4 Simulation Flow

When the simulation began, 100 agents were initialized by giving each of them a random number of executive-net units as described in the previous section, and connection weights and unit biases initialized with random values ranging from -0.25 to +0.25. After the initialization stage, the simulation proceeded in turns and generations, with each generation composed by a variable number of turns depending on agent actions.

In each generation all 100 agents were tested in all 48 mazes, starting three times in each of the three rooms for a total number of nine tests in each maze (we chose to perform three tests in each room in order to reduce chance effects given by stochastic processes in the output units, see section 2.3). Agents were tested individually. Before entering a maze, each agent was provided with an emulated triplet-net memory suited to that specific maze, and all unit activations were reset. Each test then proceeded in turns.

In each turn the agent input units perceived the room color where the agent was, and passed it to the triplet-net S1 units and to the executive-net units. The executive-net was activated by it and sent the triplet-net an activation vector overwriting the current S1 activation values from the input units. The triplet-net modified or completed the vector so that it would respect a color (room) + action (door) + outcome combination actually present in the current maze, and sent the result back to the executive-net. On this 'answer' the executive-net 'decided' which action to perform (which door to take, or staying still) and consequently activated the output units. Finally, the action now coded in the output units was executed and its effects (room or fitness score change) applied. All actions were recorded for later evaluation offline. If an agent would stay still for more than

two turns in the reward or penalty rooms, or four turns in the introductory room, he was given a small penalty, the test was ended, and a new test was started.

Once all agents had performed all tests, a generation was ended and the agents were either reproduced into the next generation or discharged, according with their final fitness score. In case of tied scores preference was given to the faster acting agent(s). During reproduction, point mutation and genetic crossover were randomly applied to the offspring, with bit mutation rate = 0.0001 and crossover rate = 1. The main simulation was ended when the correct actions in any mazes and any rooms would exceed 95%, or after 1000 generations with no improvement > 1% in any maze or any room. In order to properly compare the experimental population with the control population, control population simulation was carried on up to the same generation number as the experimental population even after no substantial improvement occurred along over 1000 generations.

5.2.5 Mind Reading

After the simulation ended, the overall best agent in the last generation was tested in all mazes once again, and interactions between its executive-net and triplet net were analyzed to uncover the agent ‘reasoning’ strategy. This was possible because those interactions consisted in activation vectors on the triplet-net units reflecting the vectors actually present in the environment, vectors of which we knew the meaning of each 0 or 1.

5.3 Results

After about 13000 generations the agent population reached 99% correct (‘good’) actions in all rooms in all mazes (Fig. 11 upper row). Reducing the mutation rate to one tenth of the original rate led in a further 1000

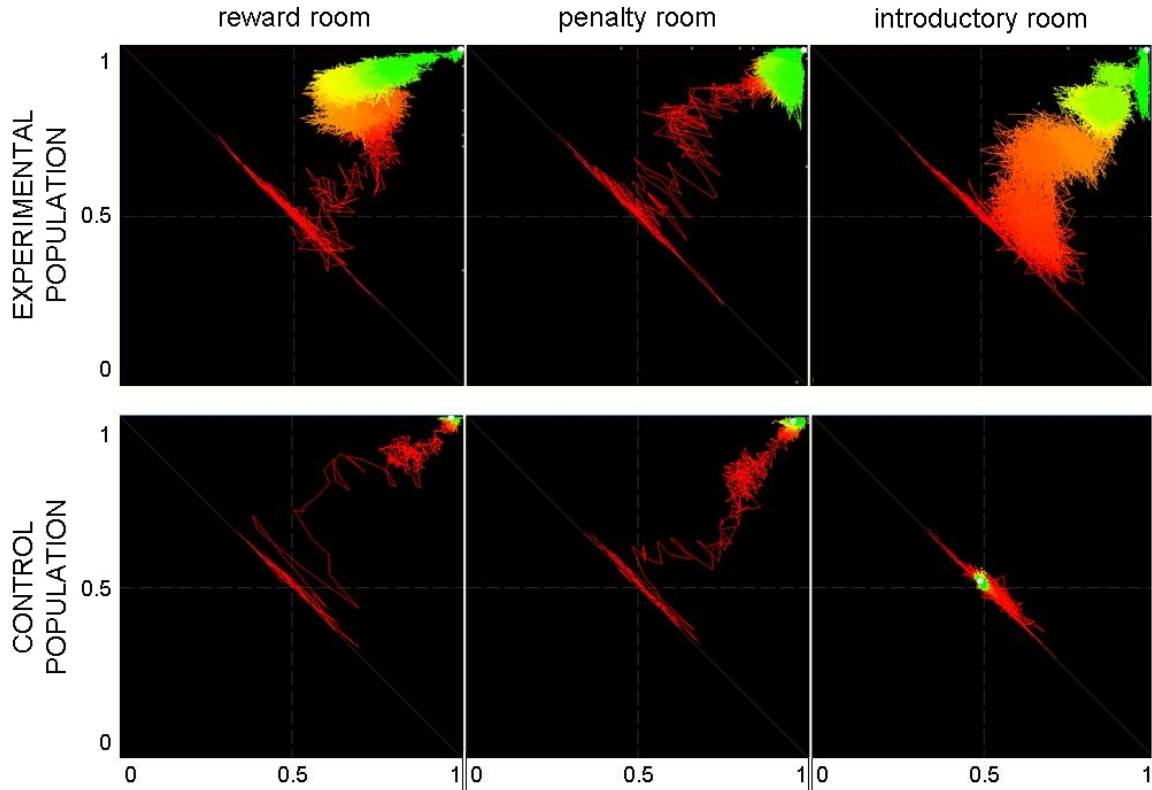


Figure 11: Evolutionary path for 13000 generations, measured as the mean prevalence of correct actions performed by agent population in the reward room (left), penalty room (middle) and introductory room (right). Vertical axis: values for the left door; horizontal axis: values for the right door. Upper row: experimental population; bottom row: control population with no connections from the executive-net to the triplet-net. Each coloured point represents one generation, with red colour for earlier generations and green colour for later generations. Each window center point represents 50% correct actions, that is altogether random actions; the top-right corner represents 100% correct actions for both the left and right door, i.e. evolutionary success. Experimental population was successful in all three rooms. Control population was successful in the reward and penalty room, solvable by automatic action, but it never deviated from random actions in the introductory room, where taking the correct door needed imagining what would happen next.

generations to almost 100% correct actions. The executive-net consisted of 30 units in every agent.

The control population (Fig. 11 bottom row) reached over 95% correct actions in the reward or penalty room after just 500 generations, but it never deviated from 50% (corresponding to all random results) in the introductory room, even after 13000 generations. The executive-net consisted of just 3 units in every agent.

The ‘reasoning’ strategy developed by evolution in the successful population consisted in the executive-net activating the triplet-net with a same sequence of questions in every maze. Of course those questions were just vectors with 0 and 1 values; here we translate in common human language what those 0 and 1 meant.

- a) On the first turn, the first question invariably was: ‘what happens if in my current room I take the left door?’ If the triplet-net returned a reward or penalty, thereby notifying the agent of being in the reward or penalty room, then the executive-net immediately knew which door to take (the same left door if the triplet-net returned the big reward or the small penalty, the opposite door if the triplet-net returned the small reward or the big penalty), and it made the agent acting accordingly.
- b) If the triplet-net instead returned a different room and no reward or penalty, thereby notifying the agent of being in the introductory room, then the executive-net made the agent staying still, and on the second turn it asked the triplet-net a second question: ‘what is going to happen if in the room beyond the right door I will take the left door?’ If the triplet-net reported a reward, either a big or small one, now the executive-net had all information needed to go to the big reward (it would take the right door from the introductory room to the reward room, then it would take the door identified by the second answer as the big reward door).
- c) If otherwise the second answer reported a penalty, then the executive-net commanded the agent to take the other door (i.e. the left door), thereby going to the reward room, and on the third turn it asked the triplet-net a third question: ‘what happens if in this room I take the left door?’ This was like the question in a) and it worked the same way (see above).

With this strategy, agents could solve every maze in just one turn when starting in the reward or penalty room; three turns when starting in the introductory room. This was one turn less than the maximum allowed.

5.4 Discussion

In designing this simulation we chose to emulate triplet-net memories rather than use full-featured ANN memories, because we needed working memories deprived of their learning ability. Even if this was self-contradictory and very different from real nervous systems, we needed it because in real nervous systems repeated experience leads to automatic behaviour, and this occurs because nervous systems can learn. Automatic behaviour is the opposite of intentional behaviour, and since this experiment was focused on intentional behaviour we had to prevent our agents from *learning to act*; we wanted our agents *learn to think* what actions to perform.

We solved this problem by allowing only genetic learning in the executive-nets, and no learning in the triplet-nets. The triplet-nets could not learn because they were emulated nets, with no modifiable internal connections. As of the executive-nets, genetic learning would not let them develop automatic responses, because genetic learning can only capture constant environmental features, but the designed environment contained no constant features, since the 48 mazes that any agent experienced were altogether different in their features. Therefore the only abilities that genetic learning could provide to the executive-nets were the ability to drive the triplet-net with a same sequence of activation vectors useful in any maze, and the ability to select appropriate actions on the answers delivered by the triplet-net.

Our results unequivocally showed that the proposed neural architecture, consisting of an associative memory net containing information on the environment, connected to an executive-net triggering memory recall in the memory net and using those information to achieve goals, is a working model for the way by which nervous systems can generate useful, intentional actions. As we said in the introduction, a voluntary action is an action taken after foreseeing its effects. In the present model, foreseeing action effects is performed by the triplet-net on a hypothesis generated by the executive-net. When a triplet-net completes or modifies the S2 activation values in an activation vector, it is actually making a prediction about the outcome of taking the door encoded in the A units when being in the room encoded in the S1 units – a door and a room generated (‘imagined’, see below) by the executive-net by properly activating the A and S1 units.

In this model we used a triplet-net as the associative memory because a triplet-net is ideal to encode relations between actions and their effects. External activations to the triplet-net may come from sensory input units as well as from executive-net units. Activations coming from input units can only recall memories in

which the starting room is the same as the actual current room, which makes sort of an ‘automatic’ recalling. On the contrary, activations from the executive-net can also evoke memories in which the starting room differs from the current room provided by the sensory input, which is *imagining*. In the designed environment, reward and penalty rooms can be solved with automatic recalling. On the contrary, the introductory room requires two or three successive recalls, the second of which (cf. section 5.3 step b) always concerns a different room than the room currently reported by sensory input, i.e. imagination of being in another room. Both the ask sequence revealed by ‘reading the mind’ in the successful population, and the fact that the control population reached success in the reward and penalty rooms, not in the introductory room, attest that this more abstract imagination depends on the executive-net, even if it occurs in the memory net.

Insofar as the agents performed actions after foreseeing those action effects, according to Parisi (2006) we can assume that they performed voluntary actions. Still, our agents lack a relevant feature typical of the human will: the ability to change one’s own goals. Our agents’ one and only goal invariably was to increase their fitness score, a goal not even made explicit in their ‘mind’ (i.e. not represented as unit activations), but rather embedded by genotypic learning into their executive-net units and connections. This condition with one or a few built-in, unchangeable goals, that the agent is even unaware of, resembles that of animals or artificial intelligence chess playing programs, rather than that of man. Also, imagination in these agents only consists in recalling memories on past experience and using them, these agents cannot imagine anything they did not experience in the environment, neither they can create innovative solutions. Developing agent able to choose their own goals and to produce creative thinking are aims left to future research.

6 CONCLUSIONS

Evolutionary connectionism is a feasible and profitable methodology to study nervous system functions at any level, from low-level sensorimotor functions up to high-level cognitive functions. Connectionist models are supported by simulation (actually, they *consist in* simulations), which makes them something more than mere theories, and even if they are necessarily simplified with respect to their biological counterparts, they perform well in highlighting essential processes for the investigated functions and the way they emerge in biological nervous systems. We described a connectionist model demonstrating what IMP had been suggesting for over a century, i.e. that voluntary actions are initiated by imagining their sensory effects. Detailing and extending this model to what we termed the ‘triplet model’ led us to formulate an original model for cognitive functions such as imagining, foreseeing, and ultimately producing voluntary behavior. Our results with these models make us confident that future research by evolutionary connectionism might help to explain even higher functions, such as the faculty to choose one’s own goals, up to ultimately the very feeling of being conscious.

REFERENCES

- Accornero, N., and Capozza, M., 2009. *Coscienza Artificiale*, Aracne, 303-412.
- Bizzi, E., 1974. The coordination of eye-head movements. *Scientific American*, 231, 100-106.
- Calabretta, R., and Parisi, D., 2005. Evolutionary Connectionism and Mind/Brain Modularity. In Callebaut, W. and Rasskin-Gutman, D. (eds.) *Modularity. Understanding the development and evolution of complex natural systems*. The MIT Press, Cambridge, MA, 309-330.
- Caudill, M., and Butler, C., 1992. *Naturally Intelligent Systems*, MIT press.
- Chalmers, D. J., 1990. The evolution of learning: An experiment in genetic connectionism. In Touretzky, D.S., Elman, J. L., Sejnowski, T. J. and Hinton, G. E. (eds.) *Proceedings of the 1990 connectionist models summer school*, Morgan Kaufmann, San Mateo, CA, 81-90.
- Cisek, P., and Kalaska, J. F., 2010. Neural mechanisms for interacting with a world full of action choices. *Annual Review of Neuroscience*, 33, 269-298.
- Flanagan, J. R., and Rao, A. K., 1995. Trajectory adaptation to a nonlinear visuomotor transformation: evidence of motion planning in visually perceived space. *Journal of Neurophysiology*, 74(5).
- Goldberg, D.E., 1989. *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley Publishing Company, Inc., Reading, MA.

- Graziano, M., 2006. The organization of behavioral repertoire in motor cortex. *Annual Review of Neuroscience*, 29, 105-134.
- James, W., 1890. *The principles of psychology*, Harvard University Press, Cambridge, MA.
- Kiesel, A., and Hoffmann, J., 2004. Variable action effects: Response control by context-specific effect anticipations. *Psychological Research*, 68(2-3), 155-162.
- Lebedev, M. A., and Wise, S. P., 2002. Insights into seeing and grasping: distinguishing the neural correlates of perception and action. *Behavioral and Cognitive Neuroscience Reviews*, 1(2), 108-129.
- McClelland, J. L., and Rumelhart, D. E., 1986. A distributed model of human learning and memory. In Rumelhart, D. E., McClelland, J. L., and The PDP Research Group, *Parallel Distributed Processing. Explorations in the Microstructure of Cognition*, Vol. 2, MIT Press, 209-214.
- Parisi, D., 2006. *Una nuova mente*, Il Mulino.
- Rumelhart, D. E., Hinton, G. E., and McClelland, J. L., 1986. A general framework for parallel distributed processing. In Rumelhart, D. E., McClelland, J. L., and The PDP Research Group, *Parallel Distributed Processing. Explorations in the Microstructure of Cognition*, Vol. 1, MIT Press, 45-76.
- Rumelhart, D. E., Hinton, G. E., and Williams, R. J., 1986. Learning Internal Representations by Error Propagation. In Rumelhart, D. E., McClelland, J. L., and The PDP Research Group, *Parallel Distributed Processing. Explorations in the Microstructure of Cognition*, Vol. 1, MIT Press, 318-362.
- Schaffer, J. D., Whitley, D., and Eshelman, L. J., 1992. Combinations of genetic algorithms and neural networks: a survey of the state of the art. In *International Workshop on Combinations of Genetic Algorithms and Neural Networks (COGANN-92)*, Baltimore, USA, 1-37.
- Shadmehr, R., 2005. *The computational neurobiology of reaching and pointing: a foundation for motor learning*. MIT press.
- Shadmehr, R., and Arbib, M. A., 1992. A mathematical analysis of the force-stiffness characteristics of muscles in control of a single joint system. *Biological cybernetics*, 66(6), 463-477.
- Stock, A., and Stock, C., 2004. A short history of ideo-motor action. *Psychological Research*, 68(2-3), 176-188.
- Yao, X., 1993. A review of evolutionary artificial neural networks. *International journal of intelligent systems*, 8(4), 539-567.
- Yao, X., 1995. Evolutionary artificial neural networks. *Encyclopedia of computer science and technology*, 33, 137-170.
- Yao, X., 1999. Evolving artificial neural networks. *Proceedings of the IEEE*, 87(9), 1423-1447.