.

Some adaptive advantages of the ability to make predictions

Daniele Caligiore[1], Massimo Tria[2], and Domenico Parisi[3]

Institute of Cognitive Sciences and Technologies
National Research Council
44 Via San Martino della Battaglia, 00185 Rome, Italy
[1] dan.cal@libero.it
[2] tria@unisi.it
[3] domenico.parisi@istc.cnr.it

Abstract. We describe some simple simulations showing two possible adaptive advantages of the ability to predict the consequences of one's actions: predicted inputs can replace missing inputs and predicted success vs. failure can help deciding whether to actually executing a planned action or not. The neural networks controlling the organisms' behaviour include distinct modules whose connection weights are all genetically inherited and evolved using a genetic algorithm except those of the predictive module which are learned during life.

1 Introduction

Organisms respond to current sensory input from the environment with movements that change the environment or their body's physical relation to the environment. These changes at least partially determine the successive inputs that the environment sends to the organism's sensory organs but this causal relation is ignored by purely reactive organisms which only respond to current input. In contrast, more complex organisms can predict what the next sensory input from the environment is going to be, given the current sensory input and the movement with which they plan to respond to this sensory input. (For possible neural structures underlying the ability to predict in primates, see [1], [2], [3].) What are the adaptive advantage(s) of this predictive ability? What can organisms with a predictive ability do that organisms without this ability cannot do?

The possible adaptive advantages of being able to predict the sensory consequences of one's movements have already been discussed in the literature. For example, Clark et Grush [4] propose that responding to the predicted proprioceptive input resulting from one's movements may allow organisms to move faster because they don't have to wait for the actual proprioceptive input. In this paper we describe some simple simulations that address this question by demonstrating two possible roles of the ability to predict: predicted inputs can replace missing inputs from the environment and predictions of success or failure can help the individual to take decisions. If any thing prevents some critical input from reaching the organism's sensors, the organism can still behave appropriately by responding to a predicted input that replaces the missing input. If an organism can predict whether or not a planned response will produce some desired result, the organism can decide to actually execute the response in case of predicted success and avoid executing the response in case of predicted failure. In Sections 2 and 3 we describe some simulations using the first scenario and the second scenario, respectively. In Section 4 we draw some brief conclusions.

2 Predicted Inputs Replace Missing Inputs

To survive and reproduce an organism must reach (and eat) the food elements that are randomly distributed in the environment. At any given time the organism's sensory organs encode the position of the single nearest food element and the organism must respond by turning towards and approaching the food element. The organism's behaviour is controlled by a sensory-motor neural network with one input unit encoding the location of the food element which is currently nearest to the organism, one output unit encoding the movement with which the organism responds to the sensory input, and two internal units (Figure 1a). An initial population of organisms is generated by assigning random connection weights to the neural network that controls each organism's behaviour and a genetic algorithm is used to evolve in a succession of generations networks which have the appropriate connection weights that allow them to perform the task.

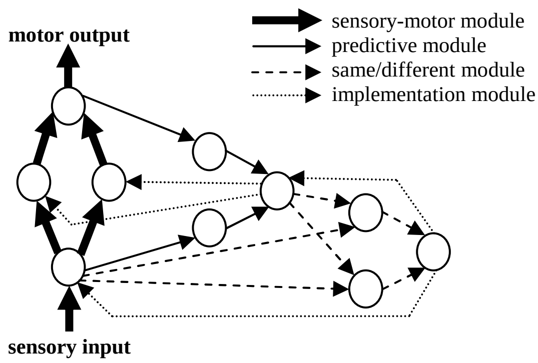

Fig. 1. (a) Sensory-motor network (or module) (thick arrows). (b) Predictive module (thin arrows). (c) Same/different module (broken arrows). (d) Implementation module (dotted arrows)

Now imagine that for a variety of reasons (failures of attention on the part of the organism, something going across between the food and the organism, etc.) in some cycles the input from the nearest food element is replaced by some other, irrelevant, input. We simulate all these different circumstances by assigning a randomly generated activation level to the neural network's input unit in a certain percentage of input/output cycles. If the organism's neural network is a simple network mapping sensory input into motor output, in these 'blind' cycles the organism is lost. The input which replaces the input from food is randomly generated but the organism has no way of knowing this and it responds to the randomly generated input as it were input from food. We expect that in these circumstances the organism's overall behaviour will be significantly less effective. But consider a somewhat more complex organism with a neural network composed of two sub-networks or modules: the module that we have already described which maps sensory input into motor output and a new module that predicts what the next sensory input will be, given the current sensory input and the planned movement with which the organism will respond to the current input (Figure 1b). This predictive module has one input unit encoding the current sensory input from the environment and another input unit encoding the planned motor response of the organism to the current sensory input, two internal units, and one output unit encoding the predicted sensory input from the environment that will appear in the next cycle, i.e., after the planned movement is physically executed. (For other simulations using this neural model of the ability to predict, see [5], [6]; for other models of learning to predict, see [7], [8]; Ackley and Littman's [9] work on evolved reinforcement-producing neural networks that guide learning is also relevant here.)

While the network's entire architecture is fixed and the connection weights of the sensory-motor module evolve and are genetically inherited, the connection weights of the predictive module are learned during life. (The weights of both modules could evolve and be genetically inherited but learning to make predictions during life tends to increase the flexibility of one's predictive abilities.) The predictive module's weights are randomly generated at birth and, early in its life, each individual organism learns to predict the next sensory input using the backpropagation procedure. In each input/output cycle the predicted input is compared with the actual input (which functions as teaching input) and the discrepancy between the two (error) is used to gradually change the predictive module's connection weights in such a way that after a certain number of learning cycles the predictive module is able to make correct predictions.

How is this predictive ability used? When a 'blind' cycle occurs, the organism replaces the missing input from food with the predicted input and responds to the predicted input rather than to the randomly generated input. We assume that early in life the organism has learned to generate correct predictions, which implies that the missing input and the predicted input are more or less the same. Therefore, the organism can respond to the predicted input as it would have responded to the actual input from food, with similar results. We expect that an organism endowed with this predictive ability will behave more or less as effectively in the world with 'blind' cycles as in the world without 'blind' cycles.

How can the organism know when the current input originating in the environment is from food and therefore is the input to which it should respond, and when the input is not from food but from some other source and therefore it should respond to the predicted input rather than to the input originating in the environment? We imagine that the organism's neural network includes two additional modules: a same/different module and an implementation module. The same/different module judges whether the current input from the environment is the same or different with respect to the predicted input. If the two are the same, this means that the current input is from food and the sensory-motor module should respond to the actual input from the environment. If the current input and the predicted input are different, this means that the current input is from some other sources and the sensory-motor module should respond to the predicted input rather than to the current input from the environment. The implementation module implements this judgment by telling the sensory-motor module which input to use. We will now describe these two modules.

The same/different module (Figure 1c) has one input unit encoding the current input from the environment and one input unit encoding the predicted input which was the output of the predictive module in the preceding cycle. In response to these two inputs the same/different module generates an output that encodes a judgment as to whether the two inputs are the same or different. (This same/different task can be interpreted as a continuous XOR task.)

The implementation module (Figure 1d) relays this same/different judgment to the sensory-motor module. To make it possible for the predicted input, rather than the actual input from the environment, to control the organism's behaviour, the output unit of the predictive module, which encodes the predicted input, has connections linking it to the two internal units of the sensory-motor module. Through these connections the predicted input can determine the organism's behaviour by replacing the actual input from the environment. The implementation module has an input unit encoding the judgment "same or different" of the same/different module and this unit sends connections to both the input unit of the sensory-motor module and the output unit of the predictive module (Figure 1d). In this way the implementation module can evolve weights for these two connections that tend to inhibit the output unit of the predictive module (encoding the predicted input) when the judgment is "same" (the current input is from food) and to inhibit the input unit of the sensory-motor module (encoding the actual input from the environment) when the judgment is "different" (the current input is randomly generated).

In the simulations that we will describe the connection weights of the sensory-motor module, those of the judgment module, and those of the implementation module, are all genetically inherited and they are developed using a genetic algorithm. Only the connection weights of the predictive module are learned during life using the backpropagation procedure.

The simulation scenario is the following. We start with a population of 100 individuals whose behaviour is controlled by a neural network with random connection weights. The total duration of an individual's life consists of 3500 input/output cycles of the individual's neural network. These 3500 cycles are divided up into 70 episodes of 50 cycles each and, at the beginning of each episode, the individual is placed all alone in a bidimensional continuous environment of 100x100 spatial units, in a randomly chosen position and with a randomly chosen orientation. (The division of life into separate episodes was introduced to increase variability.) The environment contains 20 randomly distributed food elements. When the individual happens to be within 2 spatial units from a food element, the individual eats the food element. The food element disappears, the individual's fitness is increased by one unit, and a new food element is introduced in a randomly selected location in the environment, so that the total number of food elements is always 20.

An individual has a facing direction and a visual field of 180 degrees. The neural network controlling the individual's behaviour has one input unit, two internal units, and one output unit. The input unit encodes the location of the nearest food element in the individual's visual field as a continuous value ranging from 0.2 to 0.8, with a value of 0.5 when the food is right in front of the organism, a value of 0.2 when the food is 90 degrees to the right, and a value of 0.8 when the food is 90 degrees to the left. The distance of the food is not encoded and the organism can see a food element whatever the distance. The input unit sends one connection to each of two internal units and the two internal units send their connections to the single output unit (Figure 1a). The output unit encodes the individual's movements, and more specifically the individual's turning to either left or right. The output unit's activation value is continuously mapped into the interval between 0.2 and 0.8, with 0.2 encoding a maximal right turn of 90 degrees, 0.8 a maximal left turn of 90 degrees, and 0.5 the preservation of the current facing direction. In all cycles, after the turning movement has been executed, the individual moves forward 0.5 spatial units in the new facing direction.

At the end of life each individual is assigned a fitness which corresponds to the number of food element eaten by the individual and the 10 individuals with the highest fitness generate 10 offspring each. An individual has a genotype which encodes the connection weights of the individual's neural network as real numbers and each offspring inherits a copy of its single parent's genotype. The value of each connection weight is mutated with a probability of 20% and the mutation consists in adding to or subtracting from the weight's current value a number randomly selected between 0 and 1. The 10x10=100 offspring constitute the second generation. All simulations last for 1000 generations and all simulations are replicated 10 times.

2.1 Simulation 1

Simulation 1 is a baseline simulation in which a population of organisms possessing only a sensory-motor module evolves in two different types of environments: an environment without periodic random inputs and an environment with periodic random inputs. We expect that the population that evolves in the second environment will have a significantly worse performance than the population that evolves in the first environment.

In the first environment an individual receives input from the nearest food element in all cycles. In the second environment in each cycle there is a 30% probability that the input from food will be replaced by a random input. Therefore, in the cycles in which the input from food is missing and is replaced by a random input, the organism will respond in a way which will tend to reduce its fitness.

The results show that, in fact, the average fitness of the population living in an environment where all inputs are from food is higher than that of the population living in the environment where some inputs can be random (Figure 2).

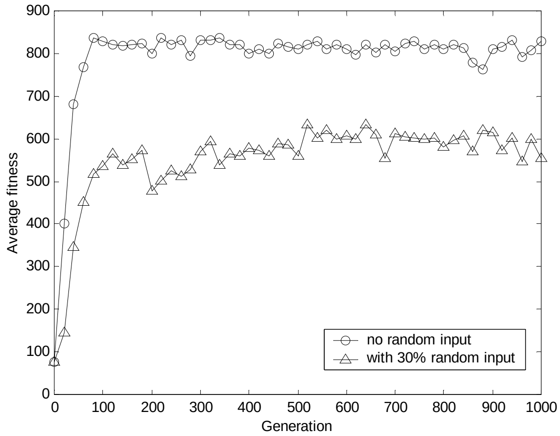

Fig. 2. Average fitness of a population living in an environment in which all inputs are from food and a population living in an environment in which inputs from food are replaced by random inputs 30% of the times

2.2 Simulation 2

In this and the following simulations the population lives in an environment where some inputs can be random. However, the organisms' neural network is more complex than that of Simulation 1. In Simulation 2 the organism's neural network includes a predictive module in addition to the sensory-motor module and each individual learns early in its life how to predict correctly the next sensory input given the current input and the planned response to the current input. In Simulation 2 it is the researcher who, in the cycles with random input, substitutes the current input with the predicted input.

The results of the simulation show that the organisms are very fast at learning to predict correctly the next sensory input from food given the current input from food and the turning movement with which the organism plans to respond to the current input. The prediction error goes to almost zero after only four episodes of an individual's life, which means that during most of its life an organism is able to generate correct predictions of the next input from food. Since in the cycles in which the input from food is missing the researcher replaces the random input with the predicted input, this has the consequence that random inputs cannot disrupt the organism's performance. In fact, the results of the simulation indicate that the performance of these organisms in an environment where some inputs are random tends to be as good as the performance of the organisms living in an environment without random inputs (Figure 3).

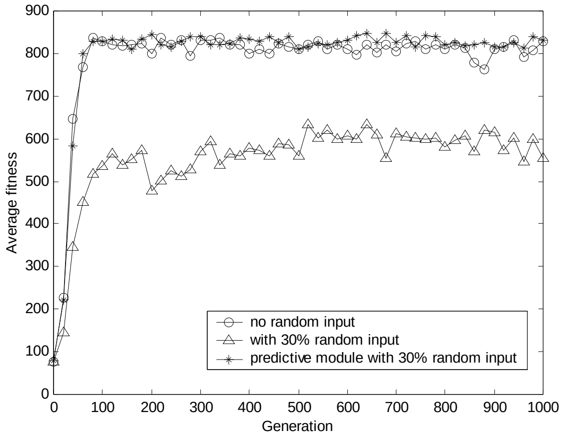

Fig. 3. Average fitness of a population living in an environment with 30% random inputs when the organisms learn early in life to predict the correct input from food and the random input is replaced by the predicted input from food. The two curves of Figure 1 are also shown for comparison

2.3 Simulation 3

In Simulation 2 the organisms learn to predict the next input from food but it is the researcher who substitutes random inputs with predicted inputs. In Simulation 3 we add a same/different module to the organisms' neural network which gives the organisms more autonomy. The same/different module judges whether the input from the environment is "same or different" with respect to the predicted input, allowing the organism to know if the current input from the environment is from food or random. The connection weights of the same/different module are also encoded in the inherited genotype and they evolve together with the connection weights of the sensory-motor module. However in Simulation 3 it is still the researcher who, if the same/different module's output is "same", causes the sensory-motor network to respond to the input from the environment, whereas if the judgment module's output is "different", he or she substitutes in the sensory-motor module the actual input from the environment with the predicted input generated as output by the predictive module.

The results of the simulation show that the genetic algorithm is able to develop appropriate connection weights for the same/different module, allowing the organism to decide most of the time correctly whether the predicted input and the actual input are the same or different. The researcher replaces the input from the environment with the predicted input if the judgment is "different" and it allows the sensory-motor module to respond to the input from the environment if the judgment is "same". Since the evolved weights of the same/different module are not perfect, the organisms' performance tend to be less good than that of the organisms living in the environment without random inputs but significantly better than the performance of the purely sensory-motor organisms living in the environment with random inputs (Figure 4).

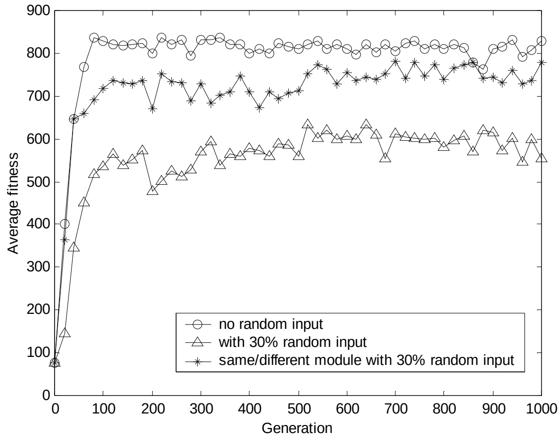

Fig. 4. Average fitness of a population living in an environment with 30% random inputs when the organisms learn early in life to predict the correct input from food and are able to judge if the input from the environment is “same” or “different” with respect to the predicted input. If the judgment is “same”, the researcher will cause the organisms to respond to the input from the environment, whereas if the judgement is “different”, the researcher causes the organisms to respond to the predicted input rather than to the input from the environment. The two curves of Figure 1 are also shown for comparison

2.4 Simulation 4

This is the final simulation in which, unlike the preceding simulations, the researcher has no role in determining the organism’s behaviour, the organisms are completely autonomous, and every aspect of their behaviour emerges spontaneously through evolution and learning. The implementation module is added to the organisms’ neural network and the genetic algorithm is responsible for all the connection weights of their network, except those of the predictive network which are learned during the individual’s life and therefore are not genetically inherited.

The results of the simulation show that it is possible to develop completely autonomous organisms that know when it is appropriate to respond to the input from the environment and when it is appropriate to ignore the input from the environment and respond to the predicted input. After a certain number of generations the implementation module develops the appropriate connection weights that allow the implementation module to inhibit the actual input from the environment and to cause the predicted input to determine the organism’s behaviour in the cycles in which the input from the environment is random and therefore is different from the predicted input. On the other hand, when the input from the environment is from food and therefore is the same as the predicted input, the implementation module’s connection weights allow the module to inhibit the predicted input and to leave to the actual input from the environment control on the organism’s behaviour. These entirely autonomous organisms also perform significantly better than the purely sensory-motor organisms living in the environment with random inputs (Figure 5).

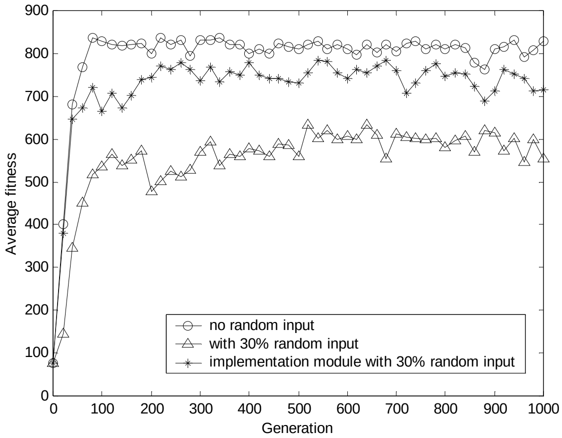

Fig. 5. Average fitness of a population living in an environment with 30% random inputs when the organisms learn early in life to predict the correct input from food and they are able both to judge if the input from the environment is “same” or “different” with respect to the predicted input and to use this judgment to decide whether to respond to the actual input from the environment or to the predicted input. The two curves of Figure 1 are also shown for comparison

We have also done a control simulation aimed at clarifying a question which inevitably arises with organisms that are able to predict the next input from the environment and to respond to this input rather than to the actual input from the environment. If the organisms’ predictions are generally correct, why should the organisms ever want to respond to inputs from the environment instead of simply responding to predicted inputs? An organism which can predict correctly the next sensory input from the environment which will result from its actions, might pay attention and respond only to the very first input from the environment and then ignore all subsequent inputs, always responding to the predicted inputs rather than to the actual inputs. Such an organism would live in a mental world rather than in the real world but its performance in the real world would be as successful as that of an organism responding to the real world.

This is not very plausible, however. Real organisms cannot live entirely in their mental (predicted) world, completely ignoring the inputs from the external environment. The reason is not only that the real world is much more variable and unpredictable than their mental (simulated) world but also that their prediction abilities are not perfect. In fact, even in our very simple and predictable world it is not possible for our simple organisms to always live in their mental world, ignoring the real world. Even if their predictions are generally correct, they are not completely correct - as indicated by the fact that the error in the backpropagation learning procedure never goes exactly to zero - and the errors of successive predictions tend to be cumulative. To demonstrate this point we have run another simulation in which the organisms are allowed to receive an input from the environment (from food) only in the single first cycle of each episode and they respond to the predicted inputs in all subsequent cycles of the episode. The results show that the average fitness at the end of the simulation is less than 200 points compared to almost 600 points of the population in which the organisms have access to the actual input from the environment in more than two/thirds of the input/output cycles (Figure 2).

3 Predicted Success or Failure Help to Take Decisions

In our second scenario, to survive and reproduce an organism has to throw a stone towards a prey animal in such a way that the stone reaches and hits the prey. Stones can be of 10 different weights and the prey can be at 10 different distances. Therefore, in any given occasion to hit the prey the organism has to throw the stone with the force appropriate to the weight of the stone currently in its hand and to the current distance of the prey. The organism's behaviour is controlled by a sensory-motor neural network (Figure 6a) with one input unit discretely encoding the weight of the stone (10 numbers equally spaced between 0.1 and 1.0), another input unit discretely encoding the distance of the prey (10 numbers between 0.1 and 1.0), and one output unit continuously encoding the force of the throwing behaviour (between 0.1 and 1). An output value which is less than 0.1 is interpreted as a refusal to throw the stone in that trial. A table defines the "physics" of the situation by specifying, for each pair of stone weights and throwing forces, the distance covered by the stone. The prey is considered as hit by the stone if the stone falls within a threshold distance from the prey. The network's connection weights are genetically inherited and are evolved using a genetic algorithm with the same parameter values of our preceding simulations. We compare this simulation with another simulation in which the organism's neural network includes a predictive module and an implementation module (Figure 6b). The predictive module generates a yes/no prediction as to whether or not the planned force with which the stone will be thrown will allow the stone to actually hit the prey. The implementation module relays this prediction/judgment to the sensory-motor module, inhibiting the throwing behaviour if the prediction is "failure" and allowing its physical execution if the prediction is "success". This more complex neural network represents an advantage for the organism if executing physical movements implies an expenditure of both time and energy for the organism. By not executing throwing behaviours that would result in failures, the organism will spare both time and energy (and perhaps avoid the flight of the prey) and therefore would increase its fitness. To implement this idea an individual's fitness is decreased by a fixed quantity for each physically executed throw.

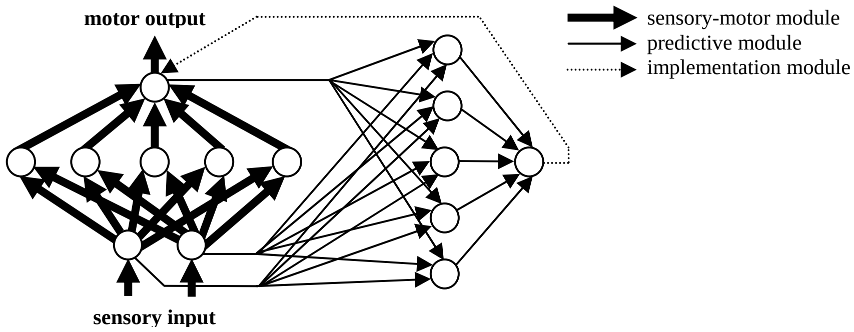

Fig.6 (a) Sensory-motor module (thick arrows). One input unit encodes the weight of the stone currently in the organism's hand, another input unit encodes the distance of the prey, and the output unit encodes the force with which the stone will be thrown. (b) Predictive module (thin arrows) and implementation module (dotted arrows). The predictive module has three input units, respectively encoding the stone's weight, the distance of the prey, and the force of the planned throwing behaviour. The module's output unit encodes a yes/no prediction on the success or failure of the throwing behaviour. The implementation module is made up of a single connection linking the output unit of the predictive module to the output unit of the sensory-motor module. The implementation module inhibits the throwing behaviour if the prediction is "failure" and it releases the execution of the behaviour if the prediction is "success"

The results of the simulation show that this is actually the case. Compared with organisms with a simple sensory-motor network, organisms with added predictive and implementation modules reach a higher fitness at the end of the simulation (5000 generations) (Figure 7).

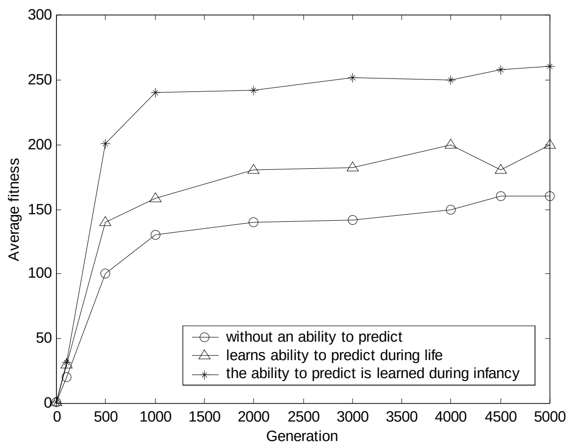

Fig.7 Average fitness (number of successful throws) across 5000 generations for a population without an ability to predict if a planned throw will be a success or a failure, a population which learns this ability during life, and a population in which the ability to predict is learned during a period of life in which the individual's fitness is not being measured (infancy)

As in the preceding simulations, both the weights of the sensory-motor module and the single weight of the implementation module are genetically inherited and they evolve in a succession of generations, whereas the weights of the predictive module are learned during life.

The model that we have described may also suggest a possible evolutionary explanation for the emergence of “infancy”. The interpretation of infancy as a “safe” period of learning has been proposed and discussed in variety of context and by many authors, e.g., in evolutionary psychology [11], attachment theory [12], and in Hurford’s [13] model of early language learning periods. In the present context infancy can be defined as the initial period of an individual’s life in which the fitness of the individual is not being evaluated by the selection mechanism because the individual is provided with the needed resources by other individuals (parents) so that the individual is free to learn some abilities (e.g., the ability to make predictions) that will be useful when the individual becomes an adult and its behaviour will be crucial for the individual’s survival and reproduction. To test this model we have compared two simulations. In one simulation an individual’s fitness is measured since the individual’s birth, and therefore it includes the period of the individual’s life in which the individual has not yet learned to make correct predictions and therefore cannot exploit the fitness advantages of being able to predict (not executing throws that would result in failures). In other words, there is no infancy. Individuals are born as adult in the sense that no one takes care of them and their fitness is evaluated from birth. In the other simulation we add infancy. The individual learns to predict during a number of additional input/output cycles that precede its regular life as an adult. The individual’s fitness is measured only when the individual becomes an adult and it already knows how to correctly predict the consequences of its actions. The results of the new simulation, also shown in Figure 7, demonstrate that learning useful abilities during a period in which other individuals provide the individual with the needed resources, i.e., during infancy, leads to a higher fitness. This may be an important selective pressure for the emergence of infancy.

4 Discussion

Complex organisms may be able to predict what sensory input will result from their planned but still non executed motor responses to the current sensory input. Why should organisms develop this capacity? What might be its adaptive value? In this paper we have described some simple simulations aimed at providing some answers to these questions. The ability to predict the next sensory input might allow an organism to replace a missing input with the predicted input. If for some reason the appropriate input from the environment is missing (due to obstacles, distractions, or other reasons), the organism can respond to the predicted input which corresponds to the missing input. Another adaptive advantage of the ability to predict is to be able to judge whether or not a planned action will produce the expected result enabling the individual to avoid physically executing expensive actions whose predicted result is not the desired one. Given these, and other (see, e.g., [9]), advantages of being able to predict the results of one’s actions we can expect that organisms possessing the appropriate prerequisites, such as the ancestors of humans, will evolve neural architectures (such as the very simplified architecture of Figure 1) that make it possible for them to predict the results of their actions and to use their predictions to generate more effective behavior. The pressures for evolving an ability to predict may have also been pressures for the emergence of infancy as the initial period of an individual's life especially dedicated to learning to predict.

References

1. Blakemore, S-J., Frith, C.D., Wolpert, D.M.: The cerebellum is involved in predicting the sensory consequences of action. NeuroReport **12** (2001) 1879-1884
2. Nixon, P.D., Passingham, R.E.: Predicting sensory events. The role of the cerebellum in motor learning. Experimental Brain Research **138** (2001) 251-257
3. Schubotz, R.I., von Cramon, D.Y.: Predicting perceptual events activates corresponding motor schemes in lateral premotor cortex: An fMRI study. NeuroImage **15** (2002) 787-796
4. Clark, A., Grush, R.: Towards a Cognitive Robotics. Adaptive Behavior **7** (1999) 5-16
5. Parisi, D., Cecconi, F., and Nolfi, S.: Econets: Neural networks that learn in an environment. Network **1** (1990) 149-168
6. Nolfi, S., Elman, J.L., and Parisi, D.: Learning and evolution in neural networks. Adaptive Behavior **3** (1994) 5-28
7. Jordan, M.I., Rumelhart, D.E.: Forward models: supervised learning with a distal teacher. Cognitive Science **16** (1992) 307-354
8. Sutton, R.S.: Learning to predict by the method of temporal differences. Machine Learning **3** (1988) 3-44
9. Ackley, D.H., Littman, M.L.: Interactions between learning and evolution. In C.G. Langton, C. Taylor, C.D. Farmer, and S. Rasmussen (eds.), Artificial Life II. Reading, MA, Addison-Wesley (1992), pp. 487-509
10. Nolfi S., Tani J.: Extracting regularities in space and time through a cascade of prediction networks: The case of a mobile robot navigating in a structured environment. Connection Science **11** (1999) 129-152
11. Buss, D.M. (ed.): The Handbook of Evolutionary Psychology. New York, Wiley, 2005
12. Bowlby, J.: Attachment and Loss. Volume 1: Attachment. London, Hogarth Press, 1969
13. Hurford, J.: The evolution of the critical period for language acquisition. Cognition **40** (1991) 159-201