Institute of Psychology
C.N.R. - Rome

Growing neural networks

Stefano Nolfi **Domenico Parisi**
Institute of Psychology
National Research Council
E-mail: stiva@irmkant.Bitnet
domenico@irmkant.Bitnet

December 1991

Technical Report PCIA-91-15

Department of Cognitive Processes and Artificial Intelligence
15, Viale Marx
00137 - Rome - Italy
0039-6-86090231

ARTIFICIAL LIFE III
Santa Fe, New Mexico, June 15-19, 1992

Growing neural networks

Stefano Nolfi Domenico Parisi
Institute of Psychology
National Research Council
15, Viale Marx, 00137 - Rome
email stiva@irmkant.bitnet

ABSTRACT

Growing neural networks are networks which are constructed by executing genetic instructions contained in a genotype. These instructions and their products interacts non-linearly to eventually determine the mature structure. We present simulations in which the mapping from genotype to phenotype is instantaneous and simulations in which it develops in time during a segment of an individual's lifetime, i.e. the individual's developmental age, allowing us to study both neural evolution and neural development. The results shed some light on (a) why modular architectures are likely to emerge, (b) why similar successions of stages tend to appear in both evolution and development, and (c) why a developmental age is preserved evolutionarily although the mature state may appear to be more efficient from the point of view of fitness.

1. INTRODUCTION

In this paper we present simulations of the evolution of populations of neural networks with two objectives. The first objective is to propose a more realistic genetic coding of neural network architecture which approximates more closely biological facts. The structure of a mature nervous system is not specified point by point in the genotype but is an emergent result of a process of growth where genetic instructions and their products interact non-linearly to eventually determine the mature structure. We want to present a genetic coding scheme which reflects more closely these properties of ontogenesis.

A second objective is to study the mapping from genotype to phenotype as a process which takes place in time, i.e. as the temporal maturation of the nervous system (neural development). In most work applying genetic algorithms to neural networks the mapping is instantaneous. Organisms are born with a mature or adult nervous system so that their behavior does not change during life unless some form of learning takes place. In real organisms, however, the mapping from genotype to phenotype is not instantaneous but it takes time extending from conception well into extra-uterine life. In other words, organisms have a developmental or maturation age during which their nervous system and, as a consequence, their behavior, changes. Using the more complex genetic coding scheme referred to above, we want to simulate the evolution of populations of organisms which have a developmental age.

In order to gain a better understanding of the role of our genetic coding scheme as such and of the introduction of a developmental age into our organisms' life, we will divide our presentation into two parts. In the first part we describe some simulations using the new genetic coding scheme for network architecture but with instantaneous mapping from genotype to phenotype. In other words, the genetic instructions that make up the genotype are all executed at the same time so that a complete phenotypic nervous system is instantaneously created at birth. In the second part we introduce biological development as a temporal process. We present new simulations in which the mapping from genotype to phenotype has a temporal course and this temporal course becomes a further property which is subject to selection and evolution.

2. GROWING NEURAL NETWORKS

Neural networks and genetic algorithms are often used together for designing artificial life systems (see, e.g., Miller and Todd, 1990; Nolfi, Elman and Parisi, 1990; Ackley and Littman, 1991). The large interest in this approach is justified because (a) neural networks are good models of the nervous system of real creatures, and (b) genetic algorithms can be used as models of the evolution of neural architecture.

To apply genetic algorithms to neural networks it is necessary to codify a network's architecture and weights into a genetic string. However, the way in which this coding should be realized is not straightforward. Many researchers have used a direct mapping of genes into network connectivities (Harp, Samad, and Guha, 1989; Miller, Todd, and Hedge, 1989; Belew, McInerney, and Schraudolph, 1990; Parisi, Cecconi, and Nolfi, 1990). But these methods, in addition to being biologically implausible, have problems in scalability, i.e. performance degrades significantly as the size of the network increases (Kitano, 1990).

In real life the mapping from genetic information into network structure is a very complex one. Langton (1992) has stressed that we cannot predict which phenotype will emerge from a given genotype because of the large non-linearities present in the mapping process. If the genotype is viewed as a set of instructions, it is not the case that each of these instructions will result in a single network property but the properties of the networks will emerge as the result of many interactions among the various genetic instructions and their products. Furthermore, the genotype/phenotype mapping is a developmental process which occurs in time so that, in principle, it must be possibile to describe various successive stages of "construction" of the final mature network and, more generally, to study the temporal course of neural development.

Another obvious characteristic of the real mapping process is that the resulting phenotype (network) is a physical object in tridimensional space. Many proposed schemes for genotypes encoding network architecture view this architecture as a simple topological structure or connectivity matrix (see, e.g., Miller, Todd, and Hedge, 1989). Real neurons and connections among neurons are located in physical space, so that, e.g., actual distance between two neurons can have an influence on whether the two neurons will eventually become connected or not.

In this paper we propose a method for codifying a network's architecture and weights into a genetic string which is inspired by the characteristics of neural development in real animals. Inherited genetic material specifies developmental instructions that control the growing and branching process of a neuron's axon (see Figure 1). Therefore, the phenotypic network architecture is not directly specified in the genotype and cannot be "read" in the genotype by simply inspecting it (Langton, 1992). It gradually emerges from the execution of the various developmental instructions that constitute the genotype and from their interactions.

During the growing process connections between neurons with their synaptic weights are established (see Figure 2). Both individual neurons and their axonal growth processes are physically located in (bidimensional) space. When a growing axonal branch of a particular neuron reaches another neuron a connection between the two neurons is established. Hence, the spatial location of neurons is critical in determining the resulting connectivity. Figure 3 shows the functional part of the network of Figure 2, i.e. the same network after isolated (nonfunctional) neurons and groups of interconnected neurons have been canceled (neural death).

In Section 3 we will describe some simulations in which the growing process is supposed to take place instantaneously at birth while in Section 4 we describe simulations in which the growing process continues for a certain time after birth. While in the first case no changes occur in the nervous system of an organism during its life, and therefore in the manner in which the organism respond with motor behavior to sensory input from the environment, in the second case we have an opportunity to study

both neural and behavioral ontogenesis.

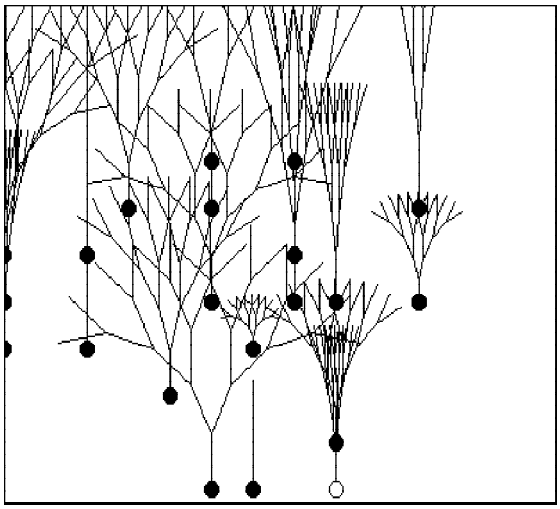

Figure 1: Growth of neuronal axons resulting from a randomly generated genetic string

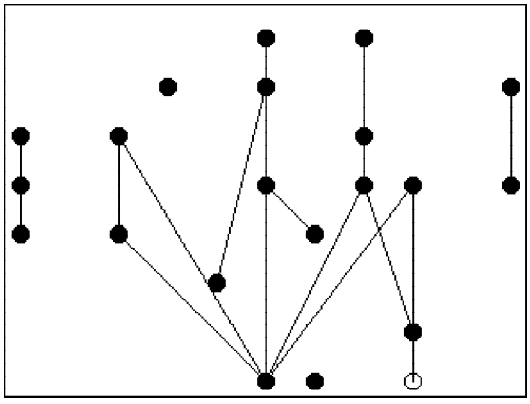

Figure 2: Connections established during the growth process showed in Figure 1. The lowest layer represents sensory neurons, the uppermost layer motor neurons, and the remaining layers internal neurons.

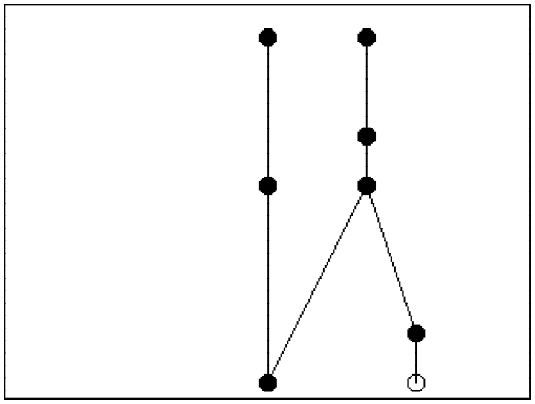

Figure 3: Resulting functional network after elimination of isolated (nonfunctional) neurons and groups of interconnected neurons.

3. EVOLUTION OF ARTIFICIAL ORGANISMS USING GROWING NEURAL NETWORKS

We are interested in the evolution of organisms that can efficiently capture food elements randomly distributed in the environment (Nolfi, Elman, and Parisi, 1990; Parisi, Cecconi, and Nolfi, 1990). The environment is a two-dimensional square divided up into cells. At any particular moment an organism occupies one of the cells. Food elements are randomly distributed in the environment with each food element occupying a single cell. The organism has a facing direction and has available a rudimentary sensory system that allows it to receive as input a coding of the angle (relative to where the organism is currently facing) and distance of the nearest food element. The organism is also equipped with a simple motor system that provides it with the possibility, in a single action, to turn any angle from 90 degrees left to 90 degrees right and then move from 0 to 5 steps forward. Finally, when an organism happens to step on a food cell, it eats the food element which disappears.

To evolve organisms which are able to reach food elements we used a kind of genetic algorithm (Holland, 1975) applied to the organisms' "genotypes". We begin with 100 randomly generated genotypes each yielding a network with a different architecture and a different assignment of connection weights. This is Generation 0 (G0). G0 networks are allowed to "live" for 20 epochs, with an epoch consisting of 250 actions in 5 different environments (50 actions in each) for a total of 5000 actions. The environment is a grid of 40x40 cells with 10 pieces of food randomly distributed in it. Organisms are placed in individual copies of these environments, i.e. they live in isolation.

At the end of their life (5000 actions) organisms are allowed to reproduce. However, only the 20 individuals which have accumulated the most food in the course of their life are allowed to reproduce by generating 5 copies of their genotype. These 20x5=100 new organisms constitute the next generation (G1). Random mutations are introduced in the copying process resulting in possible changes of the architecture and of the values of the weights (crossover is not applied).

After the organisms of G1 are created they are allowed to live for 5000 cycles. The behavior of these organisms differs slightly from that of the preceding generation (G0) as a result of two factors. First, the 100 individuals of G1 are the offspring (copies) of a subset of the individuals of G0. Second, the offspring themselves differ slightly from their parents because of the mutations. These differences lead to small differences in mean food eaten by G1 with respect to G0. At the end of their life the 20 best individuals of G1 are allowed to reproduce 5 times, forming G2. This process continues for 150 generations.

Organisms' genotypes have a fixed length and are divided up into blocks, each block corresponding to a single neuron (see Figure 4). A particular block contains instructions that determine various properties of the corresponding neuron.

(a) The "neuron expression gene" simply controls if the block will be active or not, i.e. if the corresponding neuron will be present or not in the organism's nervous system.

(b) The two "physical position genes" represent, respectively, the x and y spatial coordinates of the neuron in the bidimensional nervous system.

(c) The "branching angle gene" and the "segment length gene" control the angle of each branching of the neuron's axon and the length of each branching segment, respectively.

(d) The "synaptic weight gene" codifies the synaptic coefficients of the connections that will be established by the neuron. (All connections departing from the same neuron have the same weight.)

(e) The "bias gene" represents the activation bias of the neuron itself.

(f) The "neuron type gene" specifies, in the case of a sensory neuron, if the sensory neuron codifies angle or distance of food, and in the case of a motor neuron, if the motor neuron codifies angle of turn

or amplitude of the motor step.

There are three types of neurons, sensory neurons, internal neurons, and motor neurons. Genotypes are 40 blocks in length, i.e. each organism can have a maximum number of 40 neurons. The first 5 blocks in a genotype correspond to sensory neurons, the last 5 blocks to motor neurons, and the 30 intermediate blocks to internal neurons. Internal neurons can be arranged in a maximum of 7 layers.

Given our genetic coding scheme, more than a single input neuron codifying the same sensory information (e.g. food distance) may exist. However, the information provided by these identical neurons can be used in different ways because the neurons may have different connectivities with the rest of the network. On the other hand, if there exist more than a single output neuron codifying the same motor information (e.g. direction of movement), the actual motor output is taken to be the average of the activation levels of these neurons.

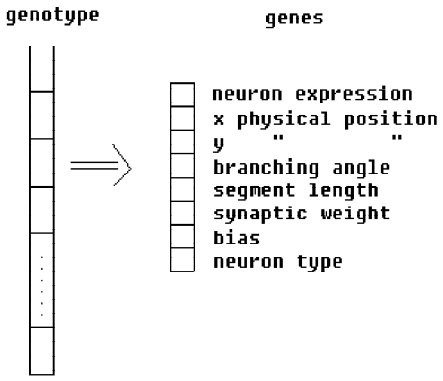

Figure 4: Parameters specified in an organism's genotype.

If we look at the organisms' fitness (i.e. number of food elements eaten) across the 150 generations we see that individuals increasingly able to approach food elements tend to evolve. Figure 5 shows the fitness value of the best individual for each of the 150 generations. The evolutionary increase in fitness implies that generation after generation the organisms of this population are able to adapt their architecture and synaptic weights to the evolutionary task.

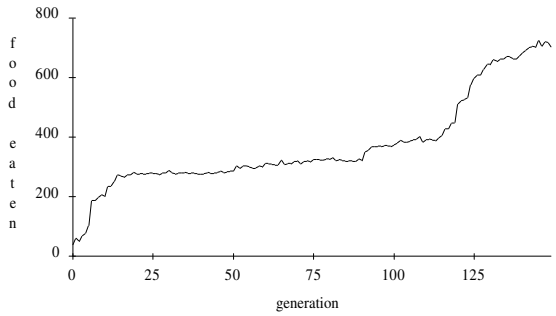

Figure 5: Fitness (=total number of food elements eaten during life) of the best individual in each of 150 successive generations

If we analyze how network architecture changes in these individuals during the course of evolution (neural evolution) we observe some interesting results. Figure 6 shows the architecture of the best individuals of a number of successive generations (for space reasons only some of the 150 generations are shown). Each picture shows the functional architecture of the corresponding organism, i.e. the architecture after isolated neurons and isolated groups of interconnected neurons have been removed.

The first thing one can notice is that evolved architectures are extremely simple. There is some

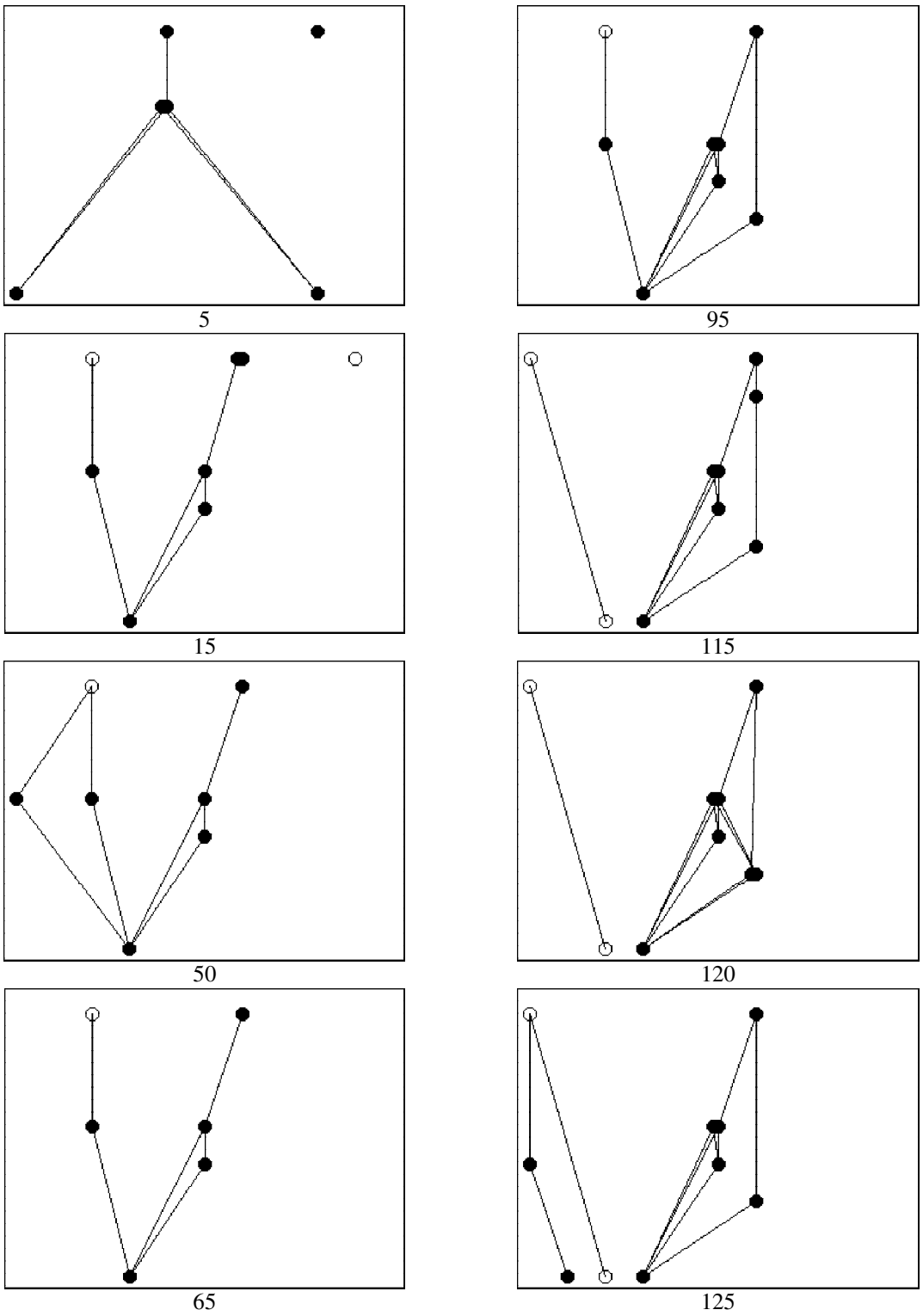

Figure 6. Architectures of the best individuals of successive generations

increase in complexity across generations but even the architectures of the later generations include very few units (much less than the maximum number allowed, i.e. 40) and very few connections. Simulations with the same type of task and environment in which a fixed architecture was used required many more connections and resulted in poorer performance (Nolfi and Parisi, 1991).

Other interesting results concern the observed regularities in the evolutionary development of the organisms' sensory and motor systems. (Notice that, within the sensory layer, filled circles represent neurons sensitive to the angle of the nearest food element and empty circles neurons sensitive to its distance. In the motor layer, filled circles represent motor neurons that codify the angle of turn and empty circles neurons that codify the amplitude of the motor step). In early generations (see G5 of Figure 5) organisms have neural architectures which can take into account only food direction (angle), on the sensory side, and can control only the direction of turn, on the motor side. Some generations later (from G15 to G95) we see the emergence of architectures which can also control the amplitude of the motor step but still rely on sensory information which is restricted to food direction. Only some further generations later, from generation 115 on, organisms that are also sensitive to food distance tend to evolve.

This progress in the evolution of neural architecture makes sense if interpreted in the light of the task and environmental constraints adopted in our simulations. Clearly, sensory information on food direction is more critical than information on food distance if the task is to approach food. Furthermore, it is more critical that sensory information on food direction be mapped into direction of movement than in speed of movement. Hence, the most primitive neural architectures that evolve have a sensory system which is sensitive to food direction only and a motor system which only controls direction of movement.

However, selection pressures pretty soon cause the emergence of mutant architectures which extend motor control to speed of movement. It is interesting, however, that for a long evolutionary time decisions on movement speed remain based on sensory information which continues to be restricted to food direction. The selective advantage of this condition is that, for example, if food is located in the back of an animal, the animal can take the correct decision to move slowly (small step forward), i.e. not to move too much in the direction away from food.

But, of course, decisions on movement speed which must be adopted when food is located in front of the animal would benefit from sensory information about food distance, in order to proceed slowly when food is nearby (to avoid going past it) and more quickly when it is distant. This represents a further selection pressure which results in the emergence of new neural architectures from G115 on. The new architectures include sensory neurons which are sensitive to food distance and which were absent in previous generations. It is also interesting that movement speed is first decided only on the basis of the newly available information about food distance, renouncing to using information about food direction to take decisions about movement speed, as it was the case in previous generations (cp. the internal architecture in G115 and G120). In fact, if movement speed must be based on one type of sensory information only, food distance appears to be more generally useful than food direction. However, in generation 125, a new network architecture finally emerges which, more adaptively, can use both food distance and food direction to take decisions about movement speed.

Finally, we observe that, despite the simplicity of the task, neural architectures progressively structure themselves in functional sub-networks or neural modules. At G115 and G120 the division of the organisms' neural system into two functional sub-networks of neurons is obvious. There is a larger sub-network which computes the organism's direction of movement as a function of the state of the neuron sensitive to food angle, and a smaller sub-network that controls the organism's movement speed as a function of food distance. At G125 this second functional sub-network becomes also sensitive to food angle through the addition of a sub-module made up of two neurons, a sensory neuron and an internal neuron. Notice that this additional sub-module was already present from G15 to G95 and was then lost at G115 when the sub-network that included the sensory neuron for food distance first appeared. Similarly, one can notice how the complex sub-network that controls direction

of movement from G95 on results from the simpler sub-network present since G15 with the addition of two internal neurons.

Architectures which are structured in functional sub-parts can more easily result in better performance through a change in just one of their sub-components. On the other hand, highly integrated, or unstructured, architectures appear to be more difficult to modify through small changes while at the same time maintaining an acceptable level of performance. As a consequence, architectures divided up into functional neural modules will have more chances to generate offspring better than themselves and this should increase the probability of these architectures to survive evolutionarily. This is an indirect type of advantage. As we will see in the next section, a developmental process temporally distributed during life will directly penalize architectures not divided into functional sub-components and will therefore provide another pressure for the emergence of modular neural architecture.

4. GROWING PROCESS AND DEVELOPMENT

In the simulations described in the previous section the process that maps the genome of an organism onto a complete network is supposed to be instantaneous. No changes occur in a network during the entire life of an individual. In other words, an individual is born as an adult and there is no ontogenesis or development.

A more biologically plausible alternative is temporally distributed mapping. The genome defines and constructs an initial network at birth but then it continues for a certain time after birth to determine changes in the network's architecture and weights. This period of genetically-based changes during an individual's lifetime can be called its developmental age.

In order to obtain organisms that develop during their life we changed the representation of their genotypes by replacing the neuron expression genes (see Figure 4) with temporal expression genes (see Figure 7). A neuron expression gene determines if the corresponding neuron will be expressed or not, i.e. if the neuron will be part of the phenotypic network or not. In the new framework a temporal expression gene specifies the particular epoch of life in which the corresponding neuron will be expressed. Neurons that are scheduled to appear later than the individual's death are non-expressed neurons.

We have run a new set of simulations with this new genetic representation. Organisms are allowed to live for 10 epochs, with all the other parameters remaining the same as those of the simulations described in the previous section.

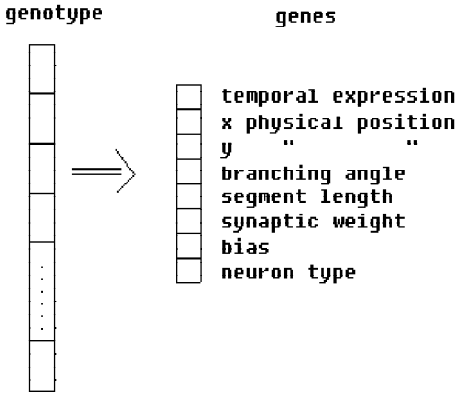

Figure 7: Parameters specified in new genotype.

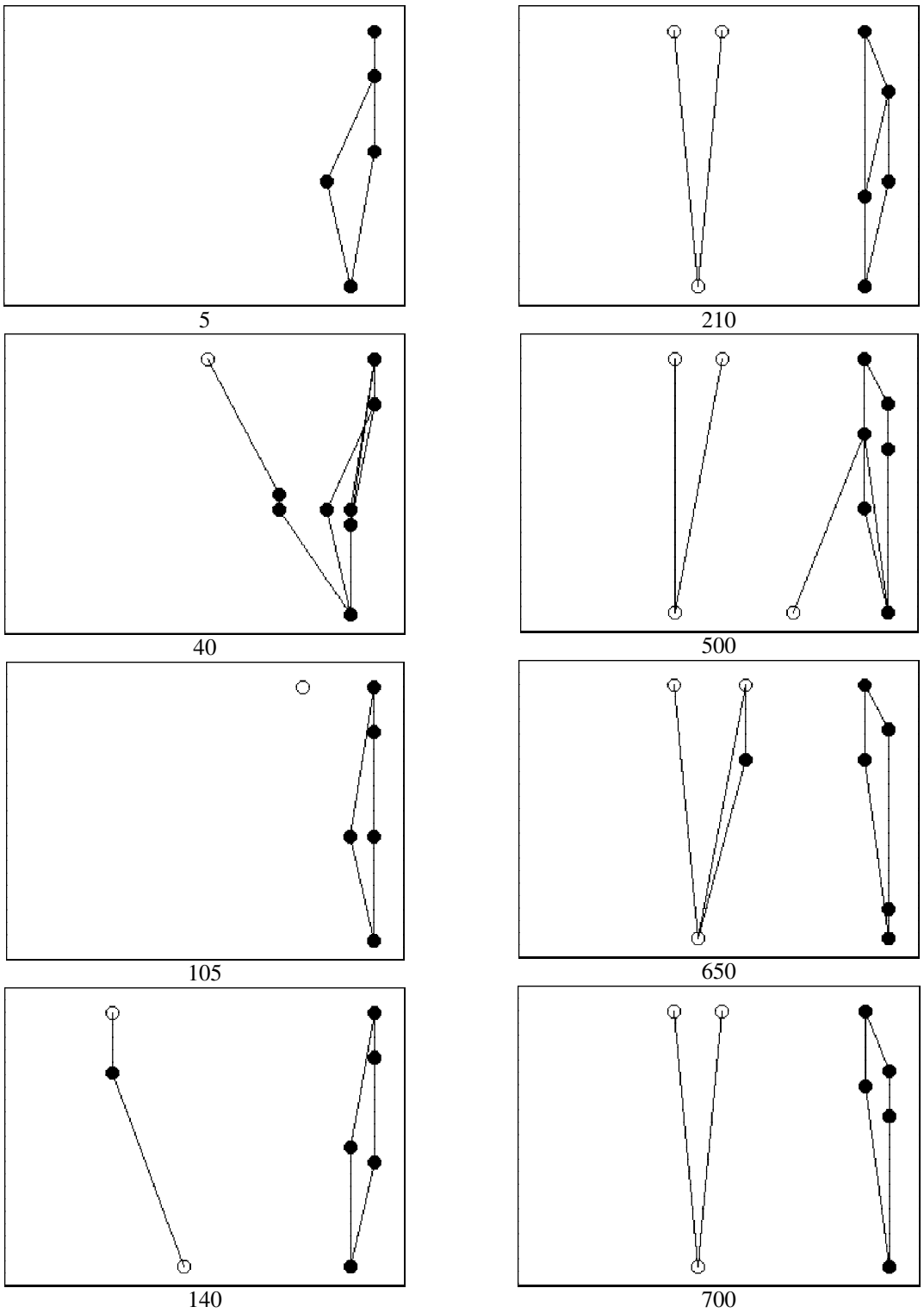

Figure 8. Architecture of the best organisms of successive generations at the end of their life.

Organisms with a developmental age evolve an ability to capture food as do organisms without a developmental age. Furthermore, if we analyze how adult network architectures, i.e. architectures after completion of developmental age, change during the course of evolution (see Figure 8) we obtain an evolutionary pattern similar to that shown in Figure 6. As in the previous case without ontogenesis, evolved architectures are extremely simple and they are structured in functional sub-networks. Furthermore, the complexity of sensory and motor systems increases generation after generation while remaining co-adapted at every stage of evolution.

However, in contrast with the organisms of the preceding simulations, the organisms of these new simulations, in addition to changing at the evolutionary time scale, can also change during their life. In other words, with these new organisms we can also study neural and behavioral development, i.e. how network architecture and behavior change during an individual's lifetime.

Let us consider behavior first and, more specifically, let us examine if there are changes in food capturing ability during life. If we look at these organisms' performance during life we see that, after a few generations, the amount of food eaten increases across epochs of life (see Figure 9). Such as increase continues until generation 500; afterwards performance remains more or less constant during life. This implies that, for a long period of time, the evolutionary process has selected genotypic instructions that results in behaviorally useful changes in the organisms' nervous systems after birth.

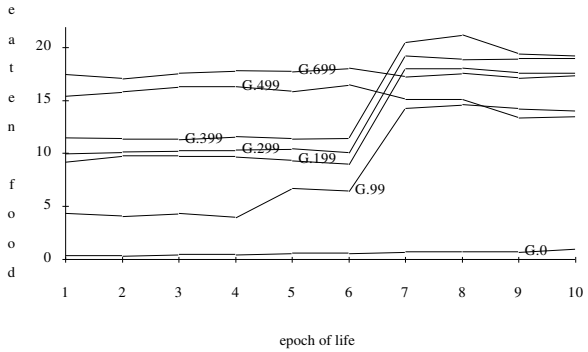

Figure 9. Number of food elements eaten at various epochs of life by organisms of successive generations.

This rises an important problem of a general nature. Since the time of expression of neurons is subject to evolution, we can ask why evolution wants to preserve a developmental age instead of entirely abolishing development and generating organisms that are born adult. The time of neural expression varies from one individual to another and, therefore, selective reproduction is free to choose among genotypes resulting in developmental ages of various lengths. In the extreme case, individuals could evolve that have no developmental age at all, i.e. that are born as mature adults. This would seem to be an interesting possibility from an evolutionary point of view since, presumably, an immature "brain" generates behavior which is less efficient from the point of view of fitness (number of food elements captured) than behavior generated by a mature brain. Therefore, if the selective process is free to choose whether to maintain a developmental (i.e. immature) age in the population or not, it might be in the interest of evolution to completely eliminate developmental age in order to obtain mature brains from birth.

In contrast, what we actually observe is that developmental age is preserved in our population. In these simulations an individual's life lasts for 10 epochs. Figure 10 shows the epoch in which developmental age ends (no further change in functional neural architecture) for the average individual across 700 generations. After an initial decline the end of development stabilizes at around epoch 5. Hence, developmental age stably covers half of the life of our organisms. We conclude that there must be some evolutionary advantage in having a developmental growth of the nervous system distributed in time. We will return to the issue of what this advantage might be below.

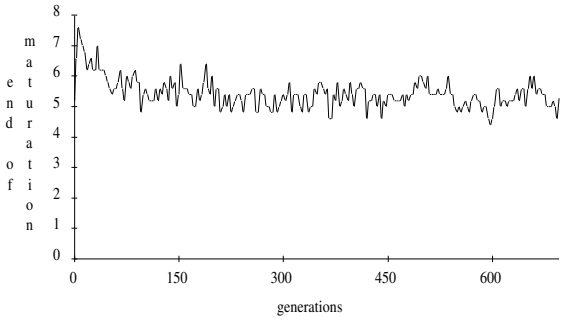

Figure 10. Epoch in which development ends across generations. Average results of 10 different simulations.

We now turn to actual neural development, i.e. to observed changes in neural architecture during an individual's life. Figure 11 shows how the architecture of a particular individual organism, the best individual of generation 140, changes during the organism's life. There is an increase in the complexity of the neural architecture across epochs of life that generally corresponds to the emergence of new functional subnetworks. In the case of the individual shown in Figure 11 the adult network can be divided into two or three functional subnetworks that appear at different times of the development process. It is interesting to note that this succession of subnetworks that appear developmentally is similar to the evolutionary succession of sub-components that we identified in the previous simulation with instantaneous development. We will return to this point in a moment.

The selection of architectures made of functional sub-components can be considered as more clearly motivated in the case of a developmental process distributed in time than in the case of instantaneous development. Because organisms are evaluated during all their life, components that emerge at different times in development are likely to be selected according to two criteria: (a) how useful they are in themselves, and (b) how useful they are when combined with other components. Generally, one might assert that systems that develop and are not constructed directly in their final form, tend to have modular structure because primitive forms must be viable in themselves and these primitive forms tend to persist and become integrated into later more complex forms. In organisms with instantaneous ontogenesis, as those of our preceding simulations, the pressure to become modularized is only evolutionary: simpler forms, i.e. organisms in early generations, must be viable to survive and reproduce. However, when we add temporally distributed development, as in the organisms of the present simulations, there is a further developmental pressure to become structured in modules because each single organism is directly evaluated in all phases of its life from simple, "immature" form to complex adult form.

In any case, the necessity to be viable and efficient at all stages of evolution or development with regard to the same type of "task" (in our case, capturing food) can induce a similarity in the succession of forms both evolutionarily and developmentally. In other words, we find in our simulations some evidence for the so-called "biogenetic rule", that is, the rule according to which "ontogenesis recapitulates phylogenesis". (Medicus (1992) states that the biogenetic rule is generally accepted with reference to neural development while remaining controversial with reference to behavioral development.) What is even more interesting, our simulation may offer an explanation of why evolution and development may undergo similar successions of stages in that they are subject to the same type of pressures.

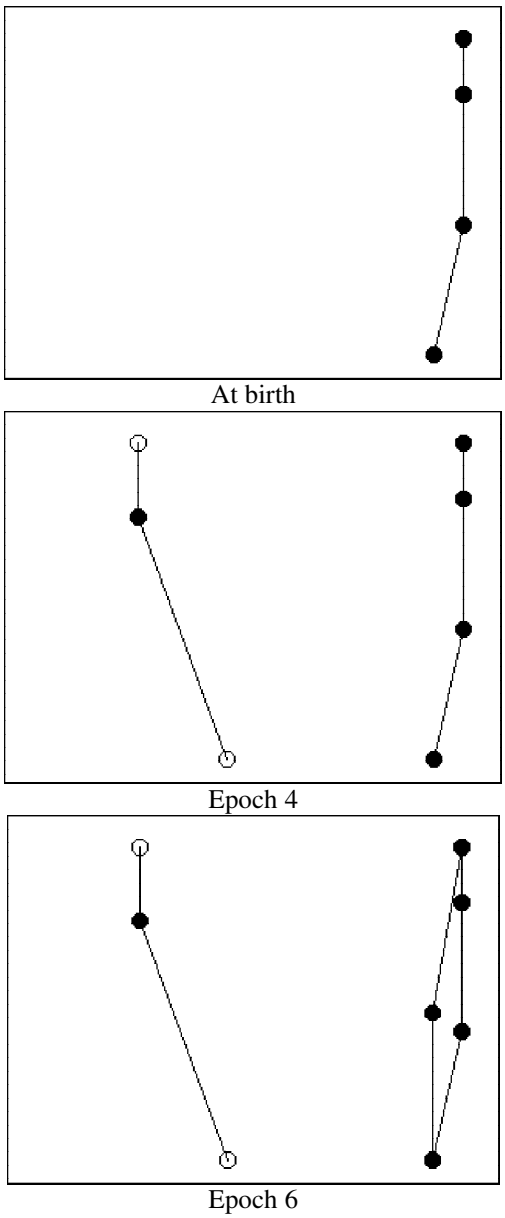

Figure 11. Changes in the neural architecture of the best organism of generation 140 during its development. The first picture shows the architecture at birth. The other two pictures show how the initial architecture changes at the fourth and then at the sixth epoch of life as a consequence of later execution of genotypic instructions.

We now return to the question of why a developmental age is maintained evolutionarily although it may appear as a liability rather than as an asset from the point of natural selection.

If we look at how the number of neurons present in the neural architecture of individuals at different times during their development change evolutionarily we find an interesting pattern. The total number of neurons (both functional and nonfunctional) present at birth increases evolutionarily with sudden jumps (see Figure 12) while the number of neurons present in adult organisms remains at a constantly high level (see Figure 13). Furthermore, most of the neurons that exist at birth are functional while only half of the neurons present at the end of life are functional. Finally, the number of functional neurons present at birth shows very little variation in successive generations, aside from the already noted jumps, while the number of functional neurons present at the end of life is much more variable

from one generation to the next.

We think that this pattern of results may explain why the expected tendency to preserve mutations that anticipate the end of developmental age does not necessarily results in a progressive reduction of developmental time. The appeareance of functional neurons tends to be anticipated evolutionarily and this results in an improvement in performance from birth (see Figure 9). However, new neurons, both functional and nonfunctional, continue to appear at later stages of development at least in selected organisms. How can we make sense of this?

After the initial stages of evolution in which a certain functional architecture and a sufficient level of performance have been established, changes due to mutations are more likely to have negative rather than positive influences on fitness. Therefore, in organisms with temporally distributed development one should expect these changes to occur later in development to limit their disruptive effects on fitness. In fact, because of the risks associated with them, changes in architecture have more chances to be preserved if they affect only a limited portion of an organism's life. Therefore, novelties introduced by evolution have more probability to be late developmental characteristics. Only if these novelties remain stable for many generations and are not replaced by other solutions they are likely to be anticipated in the developmental time scale. On the other hand, the possibility of change in later developmental stages tends to be preserved because it represents an exploratory "growth cone" for the evolutionary process, i.e. it maintains open the possibility of finding useful mutations and then stabilizing them. Thus, the usefulness of constant introduction of new functional parts at later developmental stages compensates for the tendency to anticipate the end of developmental age.

An additional proof of the fact the novelties tend to appear in the late part of the developmental age can be found in the observed result that the number of functional units that appear after birth is much more variable from one generation to the next than the number of functional units that appear at birth (see Figure 12 and 13). Functional units that appear late in development represent novelties and as such they have more probabilty to be replaced by other better solutions, or simply to become extinct, than evolutionary stable functional units.

We conclude that opposite forces determine the end of developmental age in populations of organisms in which development is subject to natural selection. The usefulness of early mature brains for generating behavior which is efficient from the point of view of fitness, is a force which favors early end of developmental age and, in principle, could determine the complete elimination of development. However, elimination of development would mean losing an important tool for exploring evolutionary novelty. Such novelty is risky since it is likely to involve negative rather than positive influences on obtained fitness. Therefore, there is a tendency for novelties to be explored in the later stages of development, when these novelties can influence fitness more marginally. We have then here another force which acts against reducing the length of developmental age and, more radically, against eliminating development completely.

Of course, the explanation of why development exists in the biological world is much more complex than we have indicated. For example, physical growth is an intrinsic factor that causes organisms to pass through an immature phase before reaching maturity. Furthermore, young individuals may be screened against direct evaluation of their fitness through adult care - which may reduce the importance of shortening developmental age. However, it seems to us that the analysis above may have identified some significant factors that determine the length of developmental age.

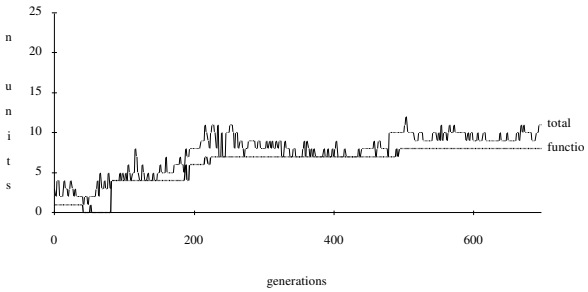

Figure 12. Total number of neurons and number of functional neurons at birth for the best individuals of successive generations.

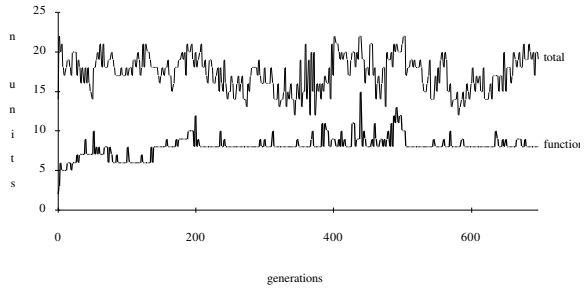

Figure 13. Total number of neurons and number of functional neurons at the end of life for the best individuals of successive generations.

5. CONCLUSIONS

We have presented simulations of the evolution of populations of neural networks that grow during life. The coding of genetic information is such that the resulting phenotypic network cannot be directly read from the genetic code but is a function of the complex interactions among the various segments of the genetic code. For example, if a particular neuron will succeed in establishing a connection with another neuron is not an information residing in each neuron separately but it depends on the interaction between the properties of the neurons concerned (where they are spatially located, the nature of their growth processes, etc.). In future simulations we want to emphasize the strongly interactive nature of neural development and neural structure. For example, we want to give the target neuron some control on the establishment of a connection, thereby simulating the role of "growth factors" such as NGF which are produced by target neurons (Tessier-Lavigne and Placzek, 1991).

Although our phenotypic networks cannot be predicted from knowledge of their genotypes, the mapping remains deterministic. In other words, given a genotype one and only phenotype can result. (We believe that it is necessary to distinguish between unpredictability and non-determinism in the genotype/phenotype mapping. See Langton, 1992.) If learning or behavior can change neural architecture, then the mapping will become non-deterministic in addition to being unpredictable.

Despite simplifications and implausibilities with respect to biological facts, we have found that an evolutionary process based on selective reproduction and mutation applied to genotypes represented in this way results in artificial organisms that present many interesting properties:

(a) The evolutionary task can be accomplished with extremely simple architectures in terms of number of units and connections. In other words, leaving evolution free to select the most appropriate architecture(s) for a task, may result in more efficient architectures than those that can be designed intentionally.

(b) Evolved architectures tend to be structured in functional sub-networks, i.e. neural modules tend to emerge evolutionarily.

(c) One observes a reciprocal adaptation, at the evolutionary time scale, between the internal organization of the system, the sensory filter of the environment, and the motor output repertoire. In other words, different functional components become more complex evolutionarily but, because of co-evolution, they tend to be co-adapted at every level of complexity.

Furthermore, we think that by making the mapping from genotype to phenotype a real temporally occurring process which takes place during the life of individual organisms, the way is open to study genetically controlled neural and behavioral development as an additional dimension of change besides evolution and learning. Artificial life, as the computational study of real and possibile life forms, is centrally concerned with change. But what is characteristic of biological change is that it takes place at various levels and in various forms, and the interaction among different dimensions of change is critical.

In any case, the results of our simple simulations appear to already shed some light on a number of significant issues, i.e. (a) why neural evolution and neural development tend to generate modular architectures; (b) why there may be a tendency for ontogenesis to go through successive stages that are similar to successive stages in phylogenesis; and (c) why the life cycle of many organisms includes a developmental or immature period, although the mature state may appear to be more effective from the point of view of fitness.

Acknowledgements

This research was supported by P.F. "TRASPORTI", C.N.R., Italy

REFERENCES

Ackley, D.E. and Littman, M.L. Interactions between learning and evolution. *Proceedings of the Second Conference on Artificial Life*. Addison-Wesley: Reading, MA. 1991.

Belew, R.K., McInerney, J., Schraudolph, N. Evolving networks: using the genetic algorithm with connectionist learning. CSE *Technical Report* CS89-174. University of California, San Diego. 1990.

Harp, S., Samad, T., and Guha, A. Toward the genetic synthesis of neural networks. *Proceedings of the Third nternational Conference on Genetic Algorithms.*San Mateo, Kaufmann, 1989.

Holland, J.J. *Adaptation in natural and artificial systems*. Ann Arbor, Michigan: University of Michigan Press. 1975.

Kitano, H. Designing neural networks using genetic algorithms with graph generation system. *Complex Systems*, **4**, (1990): 461-476.

Langton, C.G. Artificial life. In L. Nadel and D. Stein (eds.), 1991 Lectures in Complex Systems. Reading, Mass., Addison-Wesley, 1992.

Medicus, G. The inapplicability of the biogenetic rule to behavioral development. *Human Development* **35**, (1992): 1-8

Miller, G.F. and Todd, P.M. Exploring adaptive agency I: theory and methods for simulating the evolution of learning. In D.S. Touretzky, J.L. Elman, T.J. Sejnowski and G.E. Hinton (eds.), *Proceedings of the 1990 Connectionist Models Summer School*. San Mateo, CA: Morgan Kaufmann. 1990.

Miller, G.F., Todd, P., and Hedge, S. Designing neural networks using genetic algorithms. *Proceedings of theThird International Conference on Genetic Algorithms*. San Mateo, Kaufmann, 1989.

Nolfi, S., Elman, J, and Parisi, D. Learning and evolution in neural networks. CRL *Technical Report* 9019. University of California, San Diego. 1990.

Nolfi, S., and Parisi, D. Auto-teaching: networks that develop their own teaching input. *Technical Report* PCIA-91-03. Institute of Psychology, C.N.R. 1991.

Parisi, D., Cecconi, F., Nolfi, S. Econets: neural networks that learn in an environment. *Network*, **1**, (1990): 149-168.

Tessier-Lavigne, M., and Placzek, M. Target attraction: are developing axons guided by chemotropism?, *TINS*, **7**, (1991): 303-309.