Desired answers do not correspond to good teaching inputs in ecological neural networks

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

Abstract

In this paper we will analyze how supervised learning occurs in ecological neural networks (i.e. networks that interact with an external environment). Using an evolutionary method for selecting good teaching inputs we will show how the learning process interacts with the capability of such networks to partially determine the next input stimuli with their outputs. In trying to explain the behavior of these networks we surprisingly find that for obtaining a desired output X it is better to use a teaching input different from X. To explain this fact we claim that teaching inputs in ecological networks have two different effects; (a) to reduce the discrepancy between the actual output of the network and the teaching inputs themselves, (b) to modify the network behavior and as a consequence the network learning experiences. Evolved teaching inputs appear to represent a compromise between these two needs. We finally show how evolved teaching inputs that are allowed to change during the learning process respond differently in different period of learning first giving more weight to the (b) function and progressively later on to the (a) function.

The notion of ecological neural networks refers to an approach to the study of neural networks that views network as behaving, learning, developing and evolving in an environment (see Parisi, Cecconi and Nolfi, 1990). Hence, within this framework, the behavior of a network tends to be studied with reference to the environment in which the network behaves. The most important consequence of behaving in an environment is that the output of an ecological network partially determines the network's input. By acting on the environment with its motor output an ecological network may change the environment (i.e. the network can modify the position or the characteristics of an object in the environment) or it may change its relation to the environment (i.e. by moving itself the network can modify the angle of an object with respect to the direction it faces or even move to a different environment). Therefore sensory input becomes a function of the independent properties of the environment and the network's behavior.

We imagine that a network represents the nervous system of an organism (O) and that O's environment is a two-dimensional square divided up into cells. At any particular moment O occupies one of these cells. A number of food elements are randomly distributed in the environment with each food element occupying a single cell. O has a facing direction. We shall imagine it has a rudimentary sensory system that allows it to receive as input from the environment the angle (relative to where O is currently facing) and the distance of the nearest food element. We shall also equip O 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 cells forward. Finally, when O happens to step on a food cell, it eats the food element which disappears.

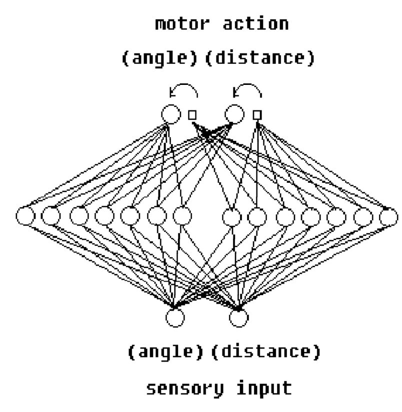

Figure 1. Auto-teaching network

The network underlying O's behavior is a feedforward network consisting of three layers (Figure 1). The input layer includes 2 units which receive sensory information from the environment. These 2 units are fully connected with two sets of intermediate ("hidden") layers of 7 units. The first set of hidden units is connected with 2 output units that code O's movement.

The second set of hidden units is connected with 2 teaching units that code the teaching input for the two output units. Sensory input is encoded by the 2 input units representing the angle and the distance of the nearest food element (both values are scaled from 0.0 to 1.0). Movement is encoded in the 2 output units that specify the amount and direction of turn and the length of the step forward (these two values are also scaled from 0.0 to 1.0).

When O is placed in the environment that has been described above, a sequence of events will occur. Sensory input is received on the input units. Activation flows up through the hidden units to the output units and to the teaching units. The values of the two output units are used to move O thereby changing the sensory input for the next cycle. The values of the two teaching units are used to change the weights that connect the input units to the output units according to the backpropagation algorithm (Rumelhart, Hinton and Williams, 1986). Then the next cycle begins.

To train the weights that connect the input units to the teaching units we used an evolutionary method (for similar approaches see among others: Holland, 1975; Hinton and Nowlan, 1987; Belew, McInerney and Schraudolph, 1990; Ackley and Littman 1991).

We run 10 simulations each starting with 100 networks with the architecture shown in Figure 1 and different randomly assigned 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 each) for a total of 5000 actions. The environment is a grid 40x40 cells with 10 pieces of food randomly distributed in it. Os are placed in individuals copies of these environments, i.e. they live in isolation.

At the end of their life (5000 actions) Os are allowed to reproduce. However, only the 20 Os which have accumulated the most food in the course of their life are allowed to reproduce by generating 5 copies of their genotypes. These 20x5=100 new Os constitute the next generation (G1). Random mutations are introduced in the copying process (crossover is not applied). Four of the weights, randomly selected, connecting the input units to the teaching units are mutated adding to them a random value included between -1.0 and +1.0. On the contrary, the weights connecting the input units to the output units are randomly generated in each offspring (i.e. they are not inherited). After Os of G1 are created they are allowed to live for 5000 cycles. The process continues for 200 generations.

In each cycle the discrepancy between the activation values of the output units and the teaching units is used to change the weights connecting the input units to the output units according to the backpropagation learning algorithm. A learning rate of 0.15 and no momentum was used. Weights are updated each cycle.

If one looks at the eating ability (i.e. number of food elements eaten) of successive generations of Os, one finds a significant increase (see curve (a) in Figure 2). This implies that weights able to generate good teaching inputs evolve and that the random weights connecting the input units to the output units are able to learn from such teaching inputs. On the other hand, if one measures how the evolved teaching inputs are good by themselves (see curve (b) in Figure 2) one finds surprisingly that they are not as good than what they are able to teach. In order to measure how good teaching inputs are we tested the Os determining their motor behavior as a function of the state of their teaching units instead of the state of their output units.

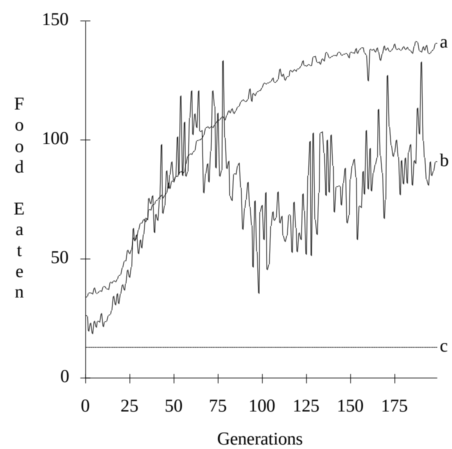

Figure 2. Performance of best Os of successive generations. (a) represents the performance of Os that learn from their auto-teaching units (b) represents the performance the same Os tested in a condition in which their movements are functions of the activation state of the auto-teaching units instead of the output units. (c) represents the performance of Os that move selecting a random action for each time step. Each curve represent the average results of 10 different simulations.

If the evolved teaching inputs had represented the right answers to each stimulus, Os that use their auto-teaching inputs to determine their movements should gather a larger amount of food of Os that use an output that start from being bad (because of the weights that generate it are random) and progressively approximate the teaching input through learning. On the contrary, after generation 75, Os tested with movements controlled by their auto-teaching units start to gather less food than the same Os that learn receiving such values as teaching input. How can we explain this fact?

Let us consider the effect of a teaching input on these ecological neural networks. It has two different effects; one is to reduce the discrepancy between the actual output of the network and the teaching itself; the second is to change network behavior and, as a consequence, to change the successive stimuli the network will experience. This second effect will in turn influence what is learned because what is learned depends on the learning experiences (for the role of the learning experience in back-propagation learning see Plunket and Marchmann, 1991; Elman, 1991).

From the point of view of reducing the discrepancy with the output of the network a teaching input should correspond to the desired answer; but from the point of view of determining good experiences a teaching input can differ from the desired answer. We can then hypothesize that teaching inputs able to change Os behavior in a way that makes them have good learning experiences are selected.

In order to verify this hypothesis we train Os receiving "optimal" teaching inputs (i.e. teaching inputs that correspond to food gathering movements) in two different learning conditions; in condition (a) we exposed Os to the experiences they will spontaneously self-select as a function of their initial random weights and of the changes such weights progressively receive in trying to approximate their "optimal" teaching input, in condition (b) we exposed Os with the same initial random weights to the same list of stimuli they would experience if they would change as a consequence of the evolved (not "optimal") teaching input even if they actually try to approximate the "optimal" one. As "optimal" teaching inputs we used the output of the best Os of each generation at the end of their life (i.e. we used the best performance obtained from evolved Os of successive generations). After this training process we evaluated Os obtained in the two test conditions for 5000 cycles letting them move in the environment without being affected by learning.

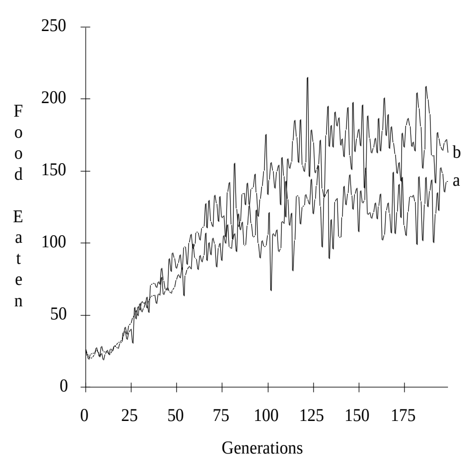

Figure 3. Performance of Os that receive as teaching input the best behavior obtained by the Os of the corresponding generation; (a) represents the performance after training in natural conditions (b) represents performance after a training in which Os experienced the same stimuli of Os trained with evolved teaching inputs. Each curve represents the average results of 10 different simulations.

The results show that Os which receive "optimal" teaching inputs performe better if they are exposed to the experiences induced by the evolved not "optimal" teaching input (see Figure 3, curve (b)). From that we can conclude that the evolved teaching inputs are able to induce good learning experiences. If we let the "optimal" teaching input naturally determine, through changing the weights, Os experience we get poorer performance (see Figure 3, curve (a)). The evolutionary process discovers that and select teaching inputs that are a compromise between this two needs; making Os approximate a good target and making Os have good experience in order to approximate that target.

If our hypothesis is correct we can expect that, if we allow teaching inputs to change during lifetime, the evolutionary process will select teaching inputs that in the first part of Os life will be particular suited for selecting good learning experiences and, later on, will progressively approximate desired answers. In fact, we should expect that learning experiences are particularly important in the first part of the learning process while, when the network progressively approximate the teaching inputs, become more and more important that such teaching inputs correspond to the best possible answer.

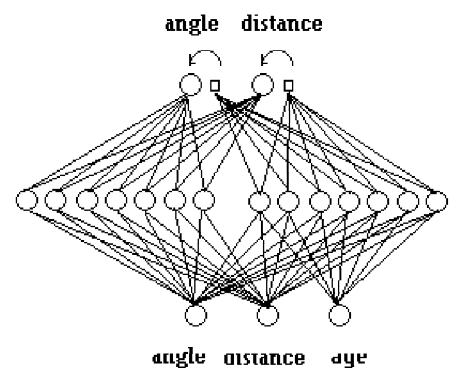

Figure 4. Auto-teaching network with an additional input units that code the age of the network itself.

In order to verify this prediction we ran another set of 10 simulations using Os with an additional input unit that code the age of the O itself (i.e. the learning cycle normalized between 0 and 1). Such an input unit affects only the part of the network that determines the auto-teaching inputs, not the part that determines the motor output of the network itself. In this way the teaching input can change becoming sensitive to different periods of the learning process (see Figure 4). All other parameters remained the same.

If we measure how evolved teaching inputs in such more complex Os change during the learning process with respect to how they approximate desired answer we can see that our prediction was correct (see Figure 5). After a few generations, teaching inputs start to change during lifetime progressively approximating desired answers (i.e. teaching inputs that allow food gathering when directly evaluated as movements).

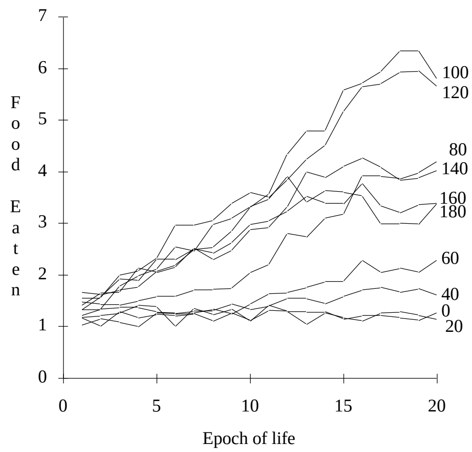

Figure 5. Eating ability during lifetime of Os of successive generations tested with movements generated as a function of the auto-teaching units activation values instead of that of the output units. Each curve is identified by the number of the corresponding generation and represents the average result of 10 different simulations.

Conclusions

We evolved networks that auto-generate their teaching inputs in order to learn to perform a simple approaching task in a simulated environment. Analyzing the evolved teaching input we surprisingly found that such teachings do not correspond to desired answers. Further analysis of the obtained results showed why this happened. Teaching inputs in ecological neural networks (i.e. in networks that partially determine their successive input stimuli as a consequence of their outputs) have two different effects; the first is to reduce the discrepancy between the actual output of the network and the teaching itself; the second is to change network behavior and, as a consequence, to change the successive stimuli the network will experience. As we showed teaching inputs that correspond to desired answers are not able to allow networks to have good experiences. As a consequence, the evolutionary process selects teaching inputs that are a compromise between two needs; making networks approximate a good target and making networks have good experience in order to approximate that target.

We further showed that if we allow auto-generated teaching inputs to change during the learning process, teaching inputs sensitive to different needs in different parts of the learning process are selected. Teaching inputs progressively change, approximating the desired answers during the course of the learning process. Teaching inputs appear particularly suited for inducing good learning experiences in the first part of the learning process when such experiences are more crucial in determining the result of learning. This finding confirms the double function of teaching inputs and suggests that, for networks that interact with an external environment an "optimal" teaching input does not exist but different, teaching inputs are adequate for different periods of the learning process.

Despite the fact that our results and our claims apply only to ecological neural networks (i.e. networks that partially determine their successive inputs as a consequence of their outputs) we think that further investigations should be conducted in order to verify if also in non ecological networks, desired answers do not always correspond to good teaching inputs.

References

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

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

Elman, J.L. (1991). Incremental learning, or the importance of starting small. Center for Research in Language, *Technical Report 9101*, University of California, San Diego.

Hinton, G.E., Nowlan S.J. (1987). How Learning Guides Evolution. *Complex System*, **1**, 495-502.

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

Parisi, D., Cecconi, F., Nolfi, S. (1990). Econets: Neural Networks that Learn in an Environment. *Network*, **1**, 149-168.

Plunkett, K. and Marchman, V. (1991). U-shaped learning and frequency effects in a multi-layered perceptron: implications for child language acquisition. *Cognition*, **38**, 43-102.

Rumelhart, D.E., Hinton G.E., and Williams, R.J. (1986). Learning internal representations by error propagation. In D.E. Rumelhart, and J.L. McClelland, (eds.), *Parallel Distributed Processing*. Vol.1: Foundations. Cambridge, Mass.: MIT Press.