Neural Processing Letters **19:** 37–48, 2004.

Ecological Neural Networks for Object Recognition and Generalization

RAFFAELE CALABRETTA[1], ANDREA DI FERDINANDO[1,2,3] and DOMENICO PARISI[1]
[1]*Institute of Cognitive Sciences and Technologies, National Research Council, Rome, Italy. e-mail: rcalabretta@ip.rm.cnr.it*
[2]*University of Rome 'La Sapienza', Rome, Italy*
[3]*University of Padova, Padova, Italy*

Abstract. Generalization is a critical capacity for organisms. Modeling the behavior of organisms with neural networks, some type of generalizations appear to be accessible to neural networks but other types do not. In this paper we present two simulations. In the first simulation we show that while neural networks can recognize where an object is located in the retina even if they have never experienced that object in that position ('where' generalization subtask), they have difficulty in recognizing the identity of a familiar object in a new position ('what' generalization subtask). In the second simulation we explore the hypothesis that organisms find another solution to the problem of recognizing objects in different positions on their retina: they move their eyes so that objects are always seen in the same position in the retina. This strategy emerges spontaneously in ecological neural networks that are allowed to move their 'eye' in order to bring different portions of the visible world in the central portion of their retina.

Key words. ecological neural networks, generalization, modularity, what and where task

1. Introduction

Organisms generalize, i.e., they respond appropriately to stimuli and situations they have never experienced before [3]. As models of an organism's nervous system, neural networks should also be capable to generalize, i.e., to generate the appropriate outputs in response to inputs that are not part of their training experience. However, neural networks seem to find it difficult to generalize in cases which appear not to pose any special problems for real organisms. What is the solution adopted by real organisms for difficult cases? Can this solution be applied to neural networks? (For a review of generalization (or invariance) in neural networks, see [5].)

In this paper we present two simulations. In the first simulation we show that while neural networks can identify the position of a familiar object in a 'retina' even when they have seen other objects but not that particular object in that position, they have difficulty in recognizing the identity of the object in the new position. In the second simulation we explore the hypothesis that organisms find another solution to the problem of recognizing objects in different positions on their retina: they move their eyes so that objects are always seen in the same position in the retina.

2. Simulation 1: Generalization in the What and Where Task

Imagine a neural network [7] which in each input/output cycle sees one of a set of different objects which can appear in one of a set of different locations in a retina and for each object it must respond by identifying 'what' the object is and 'where' the object is located (What and Where task). The network has an input retina where different objects can appear in different locations and two separate sets of output units, one for encoding the What response and the other one for encoding the Where response. Using the backpropagation procedure Rueckl et al. [6] have trained modular and nonmodular networks in the What and Where task. In both network architectures the input units encoding the content of the retina project to a set of internal units which in turn project to the output units. In modular networks the internal units are divided into two separate groups of units; one group of internal units projects only to the What output units and the other group projects only to the Where output units. In nonmodular networks all the internal units project to both the What output units and the Where output units. The results of Rueckl et al.'s simulations show that, while modular neural networks are able to learn the What and Where task, nonmodular networks are not. Notice that the What subtask is intrinsically more difficult than the Where subtask as indicated by the fact that the What subtask takes more learning cycles to reach an almost errorless performance than the Where subtask when the two tasks are learned by two separate neural networks. Therefore, in the modular architecture more internal units are allotted to the What subtask than to the Where subtask. When the two tasks are learned together by a single neural network, modular networks learn both tasks equally well, although the What subtask takes longer to learn than the Where subtask, while nonmodular networks learn the easier Where subtask first but then they are unable to learn the more difficult What subtask.

The advantage of modular over nonmodular networks for learning the What and Where task has been confirmed by Di Ferdinando et al. ([2]; see also [1]) who use a genetic algorithm [4] to evolve the network architecture in a population of neural networks starting from randomly generated architectures. The individual networks learn the What and Where task during their life using the backpropagation procedure and the networks with the best performance (least error) at the end of their life are selected for reproduction. Offspring networks inherit the same network architecture of their parent networks with random mutations. Since modular networks have a better learning performance in the What and Where task, after a certain number of generations all networks tend to be modular rather than nonmodular.

In the simulations of Rueckl et al. and Di Ferdinando et al. a neural network is trained with the entire universe of possible inputs, i.e., it sees all possible objects in all possible locations. In the present paper we explore how modular and nonmodular networks behave with respect to generalization in the What and Where task. The networks are exposed to a subset of all possible inputs during training and at the end of training they are tested with the remaining inputs. Our initial hypothesis

was that as modular networks are better than nonmodular ones with respect to learning they should also be better with respect to generalization.

In Rueckl et al.'s simulations 9 different objects are represented as 9 different patterns of 3×3 black or white cells in a 5×5 retina. Each of the objects is presented in 9 different locations by placing the same 3×3 pattern (same object) in 9 different positions in the 5×5 retina (cf. Figure 1).

Both modular and nonmodular networks have 25 input units for encoding the 5×5 cells of the retina and 18 output units, 9 of which localistically encode the 9 different responses to the Where question 'Where is the object?' while the other 9 units localistically encode the 9 different responses to the What question 'What is the object?' Both modular and nonmodular networks have a single layer of 18 internal units and each of the 25 input units projects to all 18 internal units. Where the modular and the nonmodular architectures are different is in the connections from the internal units to the output units. In modular networks 4 internal units project only to the 9 Where output units while the remaining 14 internal units project to the 9 What output units. In the nonmodular networks all 18 internal units project to all output units, i.e., to both the 9 Where output units and the 9 What output units (Figure 2).

Modular networks learn both subtasks whereas nonmodular networks learn the easier Where subtask but are unable to learn the more difficult What subtask. When asked where is a presented object, nonmodular networks give a correct answer but they make errors when they asked what is the object.

We have run a new version of the What and Where task in which the networks during training are exposed to all objects and to all positions but they do not see all possible combinations of objects and positions. In the What and Where task there are $9 \times 9 = 81$ possible inputs, i.e., combinations of objects and locations, and in Rueckl et al.'s and Di Ferdinando et al.'s simulations the networks were exposed

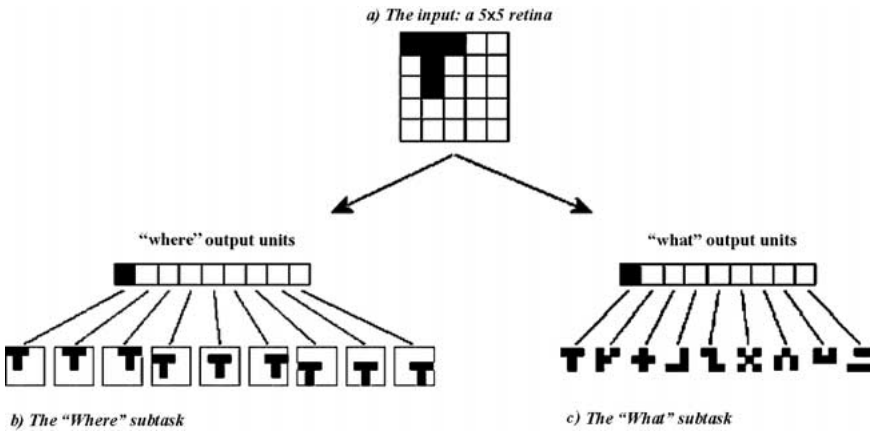

Figure 1. The What and Where task. (a) the input retina; (b) the Where subtask; (c) the What subtask.

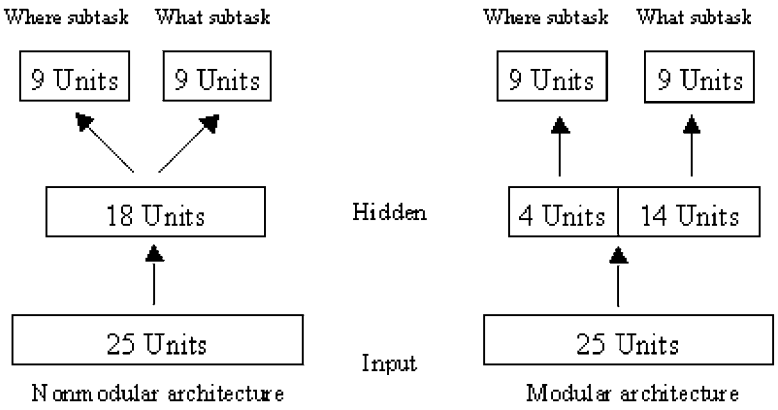

Figure 2. Modular and nonmodular network architectures for the What and Where task.

to all 81 inputs. In the present simulation the networks are exposed to only 63 of these 81 inputs and they learn to respond to these 63 inputs. The 63 inputs were so chosen that during learning each object and each location was presented seven times to the networks. At the end of learning the networks are exposed to the remaining 18 inputs and we measure how the networks perform in this generalization task.

As in Rueckl et al.'s simulations, in the new simulations which use only a subset of all possible inputs during training modular networks are better than nonmodular networks at learning the task. However, recognizing an object in a new position in the retina turns out to be equally difficult for both networks. When they are asked to identify an object which is being presented in a new location (What generalization subtask), both modular and nonmodular neural networks tend to be unable to respond correctly. Instead, both modular and nonmodular networks can perform the Where generalization subtask rather well. When they are asked to identify the location of an object which has never been presented in that location, all networks tend to respond correctly (Figure 3).

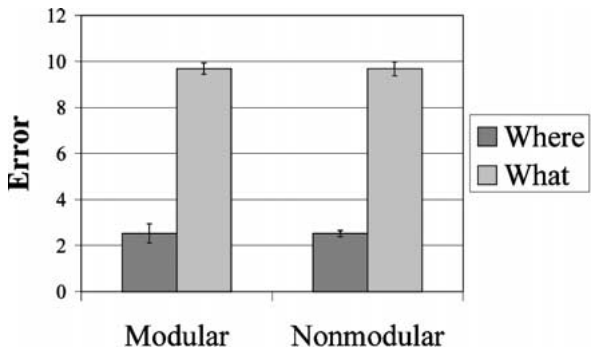

Figure 3. Error in the What and Where generalization task for modular and nonmodular architectures.

The failure to generalize in the What subtask could be a result of overfitting and we could use various methods to try to eliminate this failure. However, in the conditions of our simulations we find a clear difference between generalizing in the Where subtask vs. generalizing in the What subtask and this difference obtains for both modular and nonmodular networks. How can we explain these results? As observed in [6], 'the difficulty of an input/output mapping decreases as a function of the systematicity of the mapping (i.e., of the degree to which similar input patterns are mapped onto similar output patterns and dissimilar input patterns are mapped onto dissimilar output patterns)', and systematicity is higher in the Where subtask than in the What sub-task. This makes the Where subtask easier to learn than the What subtask and it can also make generalization easier in the former task easier than in latter one.

Another way of looking at and explaining the difference between the two tasks from the point of view of generalization is to examine how the different inputs are represented in the 25-dimensional hyperspace corresponding to the input layer of 25 units. Each input unit corresponds to one dimension of this hyperspace and all possible input patterns are represented as points in the hyperspace. Since there are 81 different input patterns, there are 81 points in the hyperspace. If the input patterns during training are 63, as in our present simulations, there are 63 points in the hyperspace. In both cases each point (input activation pattern) can be considered as belonging to two different 'clouds' of points, where a 'cloud' of points includes all points (input activation patterns) that must be responded to in the same way. Each input activation pattern generates two responses, the What response and the Where response, and therefore it belongs to two different 'clouds' . If we look at the $9+9=18$ 'clouds' of points, it turns out that, first, the 9 'clouds' of the What subtask tend to occupy a larger portion of the hyperspace than the 9 'clouds' of the Where subtask, i.e., to be larger in size, and second, the Where 'clouds' tend to be more distant from each other than the What 'clouds', i.e., the centers of the 'clouds' of the What subtask are much closer to each other than the centers of the 'clouds' of the Where subtask (Figure 4).

The consequence of this is that when a new input activation pattern is presented to the network during the generalization test, the corresponding point is less likely to be located inside the appropriate What 'cloud' than inside the appropriate Where 'cloud' . In fact, for the What subtask none of the 18 new input activation patterns is closer to the center of the appropriate 'cloud' than to the centers of other, not appropriate, 'clouds', while that is the case for 16 out of 18 new input activation patterns for the Where subtask. Therefore, it is more probable that the network makes errors in responding to the What subtask than to the Where subtask.

This results concerns the 'cloud' structure at the level of the input hyperspace, where the distinction between modular and nonmodular architectures still does not arise. However, it turns out that the difference between modular and nonmodular architectures which appears at the level of the internal units does not make much of a difference from the point of view of generalization. While modular

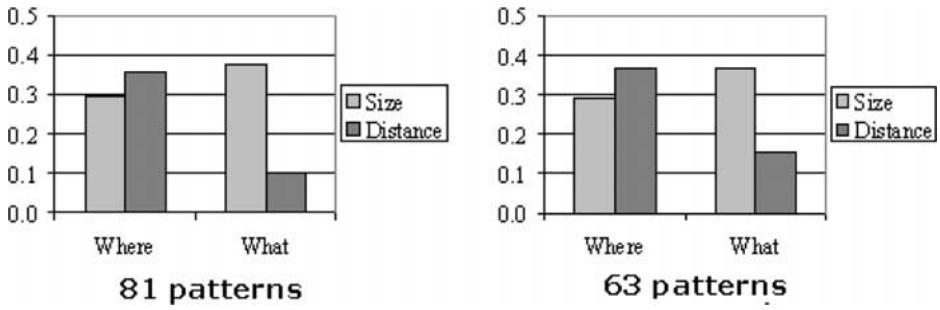

Figure 4. Size and inter-cloud distance of the 9 Where and 9 What 'clouds' of points in the input hyperspace for the total set of 81 input activation patterns and for the 63 input activation patterns used during training. Each point in a 'cloud' represents one input activation pattern and each 'cloud' includes all activation patterns that must be responded to in the same way.

architectures can learn both the What and the Where subtasks and the nonmodular architectures are unable to do so, both modular and nonmodular architectures are unable to generalize with respect to the What subtask while they can generalize with respect to the Where subtask. An analysis of the 'cloud' structure at level of the internal units can explain why this is so.

In the nonmodular architectures there is a single internal hyperspace with 18 dimensions corresponding to the 18 internal units. In the modular architecture there are two separate internal hyperspaces, one with 4 dimensions corresponding to the 4 internal units dedicated to the Where subtask and the other one with 14 dimensions corresponding to the 14 internal units dedicated to the What subtask. After training both modular and nonmodular networks with 63 of the 81 input activation patterns, we give the networks the remaining 18 activation patterns to test their generalization capacities and we examine the internal activation patterns evoked by these 18 input activation patterns. In particular, we measure the distance of each of the 18 points in the internal hyperspace from the center of the correct 'cloud', that is, the 'cloud' that each point should belong to in order for the network to be able to respond appropriately. The results are that for both modular and nonmodular networks it rarely happens that an internal activation pattern evoked by one of the 18 new input activation patterns is closer to the center of the appropriate 'cloud' than to the center of other, not appropriate, 'clouds' for the What subtask. In contrast, for the Where subtask in almost all cases the new point falls in the appropriate 'cloud' in both modular and nonmodular networks. This explains why, while both modular and nonmodular networks generalize correctly when asked Where an object is located, the What generalization subtask turns out to be impossible for both modular and nonmodular networks.

This simulation has shown that, contrary to our initial hypothesis, both modular and nonmodular neural networks have difficulties in recognizing an object if the object is presented in a position in the retina in which they have never seen that particular object during training, although they have seen other objects in that position

and that object in other positions. Real organisms are apparently able to recognize an object whatever the position in which the object appears in the retina. How do real organisms solve this problem? To suggest an answer to this question we turn to another simulation which, unlike the previous simulation, simulates an ecological task, i.e., a task in which the neural network controls the movements of a body in a physical environment and the movements partly determine the inputs that arrive to the neural network from the environment.

3. Simulation 2: Generalization in an Ecological Task

In this simulation we use ecological neural networks, i.e., networks that live and interact with a physical environment and therefore can modify with their motor output the input that they receive from the environment. The networks have a simpler task of object recognition and generalization, with only 10 (+1) possible inputs during training and 2 inputs for generalization, and they are smaller networks (fewer connection weights) than the networks used in the preceding simulations. However, the proportion of training vs. generalization inputs is more or less the same as in the task used in the preceding simulations (about 15–20%) and in any case the results are tested for statistical reliability. We have adopted a simpler task because ecological networks are better trained using a genetic algorithm rather than using the backpropagation procedure used in the preceding simulations, and the genetic algorithm works better for finding the connection weights of smaller networks. In any case, we have repeated the simulation using the backpropagation procedure with this easier task and we have obtained identical results as those of simulation 1: generalization with the What task is very difficult.

An artificial organism with a single 2-segment arm and a single movable eye lives in a bidimensional world which contains 4 objects (Figure 5).

At any given time the organism sees only one of the 4 objects. When an object is seen, the object can appear either in the left, central, or right portion of the visual field. The arm sends proprioceptive information to the organism's neural network specifying the arm's current position. The organism with its visual field and its arm are schematized in Figure 6. The task for the organism is to recognize which object is in its visual field by pressing (i.e., reaching with the arm's endpoint) the

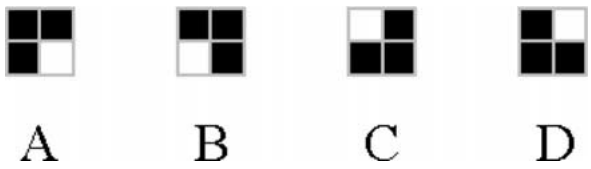

Figure 5. The 4 objects.

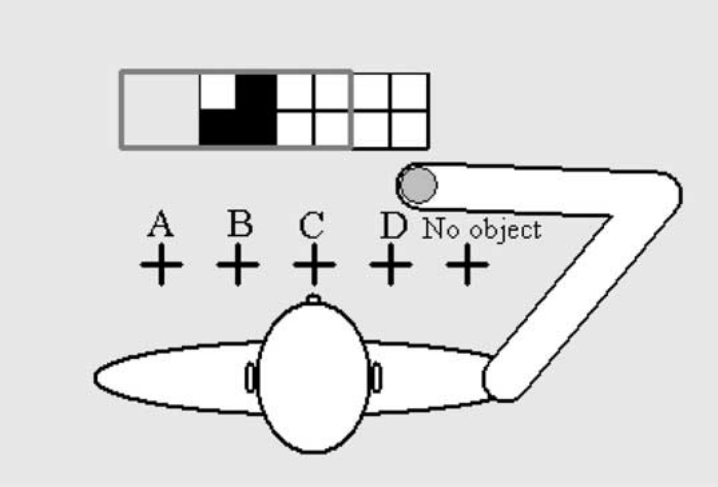

Figure 6. The organism with its two-segment arm in front of a screen (black rectangle) containing an object in its left portion. The organism has moved its eye to the left so that the object is seen at the center ('fovea') of its visual field (gray rectangle). The 5 buttons are represented as crosses localized in different positions in space.

appropriate one among 4 buttons and pressing a fifth button when no object appears.

The neural network controlling the organism's behavior has one layer of input units, one layer of output units, and one layer of internal (hidden) units. The input layer includes two distinct sets of units, one for the visual input and one for the proprioceptive input. The organism has a 'retina' divided up into a left, a central, and a right portion. Each portion is constituted by a grid of $2 \times 2 = 4$ cells. Hence, the whole retina includes $4 + 4 + 4 = 12$ cells. Each cell of the retina corresponds to one input unit. Therefore, there are 12 visual input units. Each of these units can have an activation value of either 1 or 0. An object is represented as a pattern of filled cells appearing in the left, central, or right portion of the retina (cf. Figure 6), with the cells occupied by the pattern determining an activation value of 1 in the corresponding input unit and the empty cells determining an activation value of 0.

The proprioceptive input is encoded in two additional input units. These units have a continuous activation value that can vary from 0 to 3.14 measuring an angle in radiants. The organism's arm is made up of two segments, a proximal segment and a distal segment. One proprioceptive input unit encodes the current value of the angle of the proximal segment with respect to the shoulder while the other proprioceptive unit encodes the value of the angle of the distal segment with respect to the proximal segment. In both cases the maximum value of the angle is 180 degrees. The current value of each angle is mapped in the interval between 0 (0° angle) and 3.14 (180° angle) and this number represents the activation value of the corresponding proprioceptive unit. Since the visual scene, that contains one of the 4 objects or no object, does not change across a given number of successive time steps whereas

the organism can move its arm during this time, the visual input for the organism does not change but the proprioceptive input may change if the organism moves its arm.

The network's output layer includes two distinct sets of units, one for the arm's movement and the other for the eye's movement. The first set of units contains two units which encode the arm's movements, one unit for the proximal segment and the other unit for the distal segment. The continuous activation value of each unit from 0 to 1 is mapped into an angle which can vary from -10° to $+10^\circ$ and which is added to the current angle of each of the arm's two segments. This causes the arm to move. However, if the unit's activation value happens to be in the interval between 0.45 and 0.55, this value is mapped into a 0° angle, which means that the corresponding arm segment does not change its current angle and does not move. Hence, after moving the arm in response to the visual input for a while, the network can decide to completely stop the arm by generating an activation value between 0.45 and 0.55 in both its output units.

The second set of output units contains only one unit which encodes the eye's movements. The continuous activation value of this unit is mapped into three possible outcomes: if the activation value is in the range 0–0.3 the eye moves to the left; if it is in the range 0.3–0.7 the eye does not move; if it is in the range 0.7–1 the eye moves to the right.

The 12 visual input units project to a layer of 4 hidden units which in turn are connected with the 2 arm motor output units and with the single eye motor output unit. Therefore, the visual input is transformed at the level of the hidden units before it has a chance to influence the motor output. On the contrary, the proprioceptive input directly influences the arm motor output. The two input (proprioceptive) units that encode the current position of the arm are directly connected with the two output units which determine the arm's movements. The entire neural architecture is schematized in Figure 7.

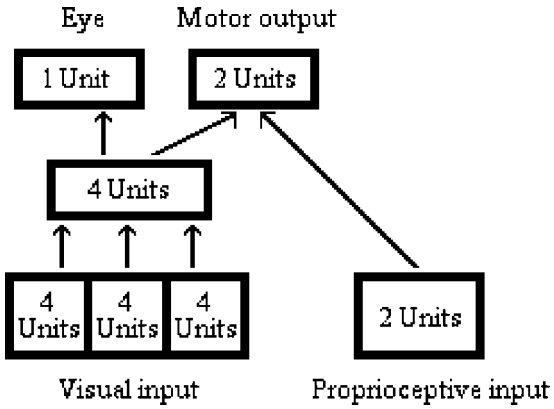

Figure 7. The network architecture.

We have studied 4 different conditions:

1. The eye can be moved and the network is trained on all the $12+1$ cases
2. The eye can be moved and the network is trained on only 10 of the 12 cases with the object
3. The eye is fixed and the network is trained on all the $12+1$ cases
4. The eye is fixed and the network is trained on only 10 of the 12 cases with the object

At the beginning of each trial, the eye is centered on the central portion of the visual field. While in conditions 1 and 2 the eye is free to move either to the left or to the right, in conditions 3 and 4 it remains fixed. However, in this position it can see the entire content of the visual field.

In conditions 1 and 3 the neural network experiences during training all the 12 possible inputs (4 objects in 3 positions) plus the zero input (no object is seen). The entire life of an individual lasts for 20 trials, which sample randomly the universe of $12+1$ possible inputs. In conditions 2 and 4 only 10 of the 12 possible inputs with the object are experienced by the neural networks during training. Two cases are excluded, i.e., object A in the left portion of the visual field and object B in the right portion (cf. Figure 5).

We used a genetic algorithm [4] to find the appropriate connection weights for our neural networks. The initial genotypes encode the connection weights of each of 100 neural networks as real numbers randomly chosen in an uniform distribution between -0.3 and $+0.3$. This is generation 1. At the end of life each individual is assigned a fitness value which measures the total number of correct buttons reached by moving the arm. The 20 individuals with the highest fitness are selected for non-sexual reproduction, with each individual generating 5 offspring that inherit the same connection weights of their single parent except that 10% of the weights are randomly mutated by adding a quantity randomly chosen in the interval between -0.1 and $+0.1$ to the weight's current value. This is generation 2. The simulation is terminated after 10,000 generations and is replicated 10 times by randomly selecting different initial conditions.

The results show that from the point of view of fitness (percentage of correct buttons pressed) there is no difference between the organisms with movable eye and the organisms with fixed eye. The possibility of moving the eye apparently gives no advantage in terms of fitness. This applies both to conditions 1 vs. 3 and to conditions 2 vs. 4.

If one examines the evolved organisms' behavior in conditions 1 and 2 (movable eye) one sees that when an object is presented in either the left or the right portion of the visual field most organisms first move their eye in order to bring the object in the central portion of the visual field and then they reach for one of the buttons.

Whereas there is no difference in fitness between organisms evolved in condition 2 and organisms evolved in condition 4, there is a difference in terms of generalization. Organisms which can move their eye are able to respond appropriately

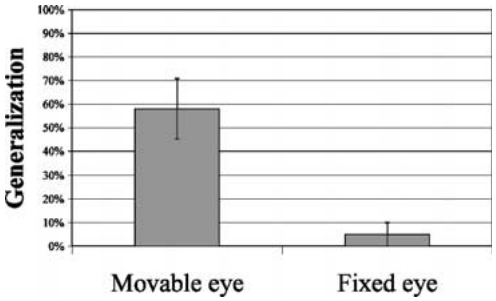

Figure 8. Percentage of correct responses in the generalization test in the condition with the eye fixed and in the condition with the eye movable.

both to the 10 visual patterns that they have experienced during training and to the 2 visual patterns that they have not seen during training. In other words, they are able to generalize. These organisms recognize the presence of an object in one of the two peripheral portions of their visual field, move their eye so that the object is brought to the central portion ('fovea'), and only then they recognize the identity of the object and press the appropriate button. In contrast, the organisms that are not allowed to move their eye are able to respond appropriately to the known visual patterns but not to the new visual patterns. They are unable to generalize (Figure 8).

Notice that in the movable eye condition when an individual fails to adopt the attentional strategy of moving its eye, the individual tends to make generalization errors.

4. Conclusion

Generalization is a critical capacity for organisms. To survive and reproduce organisms must be able not only to respond appropriately to inputs that they have already experienced in the past but also to new inputs. If we model the behavior of organisms using neural networks, some type of generalizations appear to be accessible to neural networks but other types do not. Neural networks that have been trained to recognize both the identity of an object and the object's position in the retina but do not have experienced all possible combinations of objects and positions, can recognize where an object is located in the retina even if they have never experienced that object in that position but are unable to recognize the identity of the object. Since real organisms seem to have no difficulty in recognizing the identity of objects seen in new positions in their retina, how are they able to do so? We suggest that organisms do not directly recognize the identity of an object in these circumstances but they first change the position of the object in the retina by moving their eyes so as to bring the object in a position in which they have already experienced the object (the central portion of the retina, i.e., the fovea) and then they have no difficulty in recognizing the identity of the object in that position. This is a simple

hypothesis, of course, but that may be able to explain why the fovea appears to have more computational resources (more densely packed neurons) than periphery.

This strategy emerges spontaneously in neural networks that are allowed to move their 'eye' in order to bring different portions of the visible world in the central portion of their retina. The networks learn to recognize the identity of objects only with the central portion of their retina while the peripheral portions are able to determine if an object is present peripherally but not the identity of the object. This specialization of different portions of the retina results in a 2-step behavioral strategy which allows neural networks to recognize the identity of objects whatever their initial position in the retina and to avoid the costs of enabling all retinal neurons to acquire the capacity to recognize the identity of objects. An object which appears peripherally is simply spotted by the peripheral portions of the retina and this simple information concerning the object's presence is sufficient to cause the eye to move so as to bring the object in the central portion of the retina where the identity of the object can be recognized.

References

1. Calabretta, R., Di Ferdinando, A., Wagner, G. P. and Parisi, D.: What does it take to evolve behaviorally complex organisms?, *BioSystems* **69**(2–3) (2003), 245–262.
2. Di Ferdinando, A., Calabretta, R. and Parisi, D.: Evolving modular architectures for neural networks, In: R. French & J. P. Sougné (eds), *Connectionist Models of Learning, Development and Evolution*, pp. 253–262, Springer-Verlag: London, 2001.
3. Ghirlanda, S. and Enquist, M.: One century of generalisation, *Animal Behavior*, in press.
4. Holland, J. H.: *Adaptation in Natural and Artificial Systems: an Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, MIT Press: Cambridge, MA. 1992.
5. Hummel, J. E.: Object recognition, In: M. A. Arbib (ed), *Handbook of Brain Theory and Neural Networks*, MIT Press: Cambridge, MA, pp. 658–660, 1995.
6. Rueckl, J. G., Cave, K. R. and Kosslyn, S. M.: Why are what and where processed by separate cortical visual systems? A computational investigation, *Journal of Cognitive Neuroscience* **1** (1989), 171–186.
7. Rumelhart, D. and McClelland, J.: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, MIT Press: Cambridge, MA, 1986.