Understanding the Emergence of Modularity in Neural Systems

John A. Bullinaria

School of Computer Science, University of Birmingham
Birmingham, B15 2TT, UK

j.a.bullinaria@cs.bham.ac.uk

Abstract: Modularity in the human brain remains a controversial issue, with disagreement over the nature of the modules that exist, and why, when and how they emerge. It is a natural assumption that modularity offers some form of computational advantage, and hence evolution by natural selection has translated those advantages into the kind of modular neural structures familiar to cognitive scientists. However, simulations of the evolution of simplified neural systems have shown that, in many cases, it is actually *non*-modular architectures that are most efficient. In this paper, the relevant issues are discussed and a series of simulations are presented that reveal crucial dependencies on the details of the learning algorithms and tasks that are being modelled, and the importance of taking into account known physical brain constraints, such as the degree of neural connectivity. A pattern is established which provides one explanation of why modularity should emerge reliably across a range of neural processing tasks.

Keywords: Modularity, Neural networks, Evolution, Learning

1. Introduction

Understanding the structure and information processing mechanisms of the human brain is of fundamental importance for cognitive scientists. The aim of the research presented in this paper is to gain a better understanding of the brain structures that might emerge to deal efficiently with the processing of multiple tasks within a single neural system. Given the obvious potential for disruptive interference, it seems quite reasonable that two independent and qualitatively different tasks could be simultaneously processed more efficiently if they were carried out separately by two dedicated modules, rather than together in a homogeneous (fully distributed) system. Hence, a process of evolution by natural selection would be likely to cause that advantage to result in the emergence of modularity. There is certainly considerable cognitive neuropsychological evidence that human brains do operate in such a modular manner (e.g., Shallice, 1988). This evidence involves patterns of impaired behaviour being used to inform normal cognitive structure (Caramazza, 1986), with task deficits mapped to regions of brain damage. In particular, one of the corner-stones of this approach has been the inference from double dissociation (Teuber, 1955) to modularity, and over recent years double dissociation between many tasks have been established, with the assumed implication of associated modularity. However, it is known that "resource artefacts" are possible, whereby two tasks can depend differently on a particular resource in such a way that they give rise to certain kinds of double dissociation without modularity (e.g., Shallice, 1988, p. 232). Moreover, there also exist good arguments why the double dissociation inference may be unreliable more generally (e.g., Dunn & Kirsner, 1988; Van Orden, Pennington & Stone, 2001).

A few early neural network models did seem to show that fully distributed systems could also show double dissociation (e.g., Wood, 1978; Sartori, 1988), and this cast some doubt on the inference of modularity. However, the potential for double dissociation in connectionist systems with and without modularity has since been studied extensively (e.g., Plaut, 1995; Bullinaria & Chater, 1995), and those early connectionist double dissociations are now seen to be merely the result of small scale artifacts. Several later studies (e.g., Devlin, Gonnerman, Andersen & Seidenberg, 1998; Bullinaria, 2005) have shown how weak double dissociation can arise as a result of resource artifacts in fully distributed systems,

but it seems that strong double dissociation does require some form of modularity, though not necessarily in the strong (hard-wired, innate and informationally encapsulated) sense of Fodor (1983). Plaut (1995), for example, has shown that double dissociation can result from damage to different parts of a single neural network; Plaut (2002) has explored modality-specific specialization; and Shallice (1988, p. 249) lists a number of systems that could result in double dissociation without modularity in the conventional sense. Moreover, fMRI studies (Huettel, Song & McCarthy, 2004) now provide an alternative fundamental approach for understanding brain organization, and are refining what has been inferred from brain lesion studies. For example, they have led to the suggestion that the human language processing system actually consists of "a large number of relatively small but tightly clustered and interconnected modules with unique contributions" (Bookheimer, 2002, p. 152), and that many supposed "language regions" in the brain are not actually specific to language, but involve lower level processes that are used more widely (Bookheimer, 2002; Marcus, 2004, p. 129). The whole issue is further complicated by the existence of many different definitions of "modularity", and the fact that different definitions may be appropriate for different applications or different levels of abstraction (e.g., Geary & Huffman, 2002; Seok, 2006). Then, even if one accepts a suitably general definition of modularity, there is still plenty of scope for disagreement over how much of it is innate and how much arises through learning, and what are the cost-benefit trade-offs which affect that distinction (e.g., O'Leary, 1989; Elman et al., 1996; Jacobs, 1999; Geary & Huffman, 2002).

This paper attempts to make progress on the complex issue of modularity, while avoiding much of the existing controversy, by looking at it from an orthogonal direction. It is natural to assume that, if modular systems have some advantage over non-modular systems, then evolution and/or learning will result in the emergence of modules. The important question for cognitive scientists then is: what are those advantages, and what mechanisms enable those advantages to translate into modular architectures? If one starts by simulating appropriately simplified neural systems that have the ability to evolve (or, if they prefer, not evolve) a well defined form of modularity to process some appropriately simplified tasks, one then has a solid foundation from which to understand the emergence of modularity in neural systems more generally, and a starting point for improving the realism so that we have increasing confidence that

it is telling us useful things about the operation of real brains. The idea is that the better one understands the reasons for modularity, and the mechanisms that can result in its emergence, the better one will be able to understand the structures and functions we observe in the human brain. This paper begins such an endeavor by simulating, or modelling, the evolution of some appropriately simplified neural systems performing simplified cognitive tasks, and studying the structures that emerge.

The next two sections outline the issues underlying the learning of multiple tasks by neural networks, and review some of the previous computational studies of modularity in neural systems. This leads to the identification of a number of shortcomings of the earlier studies. The following sections then describe how one can model the evolution of neural network systems, and present a series of new simulations that clarify many of the relevant issues. The paper ends with some discussion and conclusions.

2. Learning Multiple Tasks in Neural Networks

A standard structure for simple feed-forward neural network models has now become established, with three layers of simplified neurons. The input layer activations represent the system's input (e.g., a simplified retinal image). These activations are passed via weighted connections to the hidden layer where each unit sums its inputs and passes the result through some form of transfer function (such as a sigmoid) to produce its own activation level. Finally, these activations are passed through a second layer of weighted connections to the output layer where they are again summed and transformed to produce the output activations (e.g., representing classifications of the input patterns). It is known that such a structure with sufficiently many hidden units can approximate any classification decision boundary arbitrarily well (Bishop, 1995, p. 130). The connection weights are usually learned by some form of gradient descent training algorithm (such as back-propagation) whereby the weights are iteratively adjusted to reduce some appropriate error measure so that the network produces increasingly accurate outputs for each input in a set of training data (Bishop, 1995, pp. 140-148). We do not need to concern ourselves here with how exactly such a learning process might be implemented in real brains, but we do have to assume that the connection strengths in the brain can somehow be adjusted in a similar manner to minimize such an error measure. We shall return to this caveat later.

In this context, any modularity can be defined in terms of the connectivity patterns between the network's hidden and output layers, with the "modules" consisting of disjoint subsets of hidden units that share a common pattern of connectivity. During training, a hidden unit that is being used to process information for two or more output units is likely to receive conflicting weight update contributions for the connections feeding into it, with a consequent degradation of performance relative to a network that has a separate set of hidden units for each output unit (Plaut & Hinton, 1987). Such an extreme version of modularity, with a set of hidden units (or module) for each output unit, is likely to be rather inefficient in terms of computational resources, and a competent learning algorithm should be able to deal appropriately with the conflicting weight update signals anyway. Nevertheless, splitting the hidden units up into a small number of disjoint sets, corresponding to distinct output tasks, may be an efficient option. However, it is also quite possible that placing such restrictions on the neural architecture could degrade the processing efficiency, particularly if a learning algorithm is employed that can find an even better pattern of connections itself. Moreover, it is well known that when one trains a neural network using standard gradient descent type algorithms, the processing at the hidden layer tends to become fully distributed – in other words, modules do not emerge spontaneously (e.g., Plaut, 1995; Bullinaria, 1997). To see whether modularity is advantageous in practice, one needs to run explicit simulations on a representative set of tasks and neural networks.

Note that, even if advantages of particular forms of modularity are established in this way, the question still remains as to whether that modularity should be innate or learned (e.g., O'Leary, 1989; Jacobs, 1999). The Nature-Nurture debate has progressed a long way in recent years (e.g., Elman et al., 1996), and old ideas about the interaction of learning and evolution (Baldwin, 1896) can now be confirmed explicitly through simulation (e.g., Hinton & Nowlan, 1987; Belew & Mitchell, 1996). For example, in suitably simplified systems, one can observe the genetic assimilation of learned characteristics without Lamarckian inheritance, see how appropriate innate values for neural network parameters and learning rates can evolve, and understand how individual differences across evolved populations are constrained (Bullinaria, 2003b). In a similar way, evolutionary simulations can also explore the trade-offs between having innate versus learned neural structures. This paper, however, will

concentrate on exploring *why* modules should emerge, rather than the considerably more difficult task of simulating *how*. Using simulated evolution is a good way to identify the best neural architectures for particular tasks, but it does not, of course, imply that evolution (rather than learning within a lifetime) is necessarily responsible for the corresponding structures in the human brain. More detailed simulations, involving neural architectures that can change during an individual's lifetime, will be required to check that. Ebbesson (1984), Quartz (1999) and Jacobs (1999) discuss many of the issues associated with this that are beyond the scope of this paper.

Finally, it is also worth noting that in larger scale systems and more complete brain models, the "distinct output tasks" considered in this paper will often correspond to separate components or sub-tasks of more complex higher level systems, and any associated modules may be re-used by many higher level tasks. Indeed, such module re-use is another likely reason for the emergence of modularity (e.g., Marcus, 2004, pp. 133-134; Reisinger, Stanley & Miikkulainen, 2004; Kashtan & Alon, 2005), but exploring that aspect of modularity is also beyond the scope of this study.

3. Previous Simulation Studies

The earliest systematic computational study in this area was the Rueckl, Cave & Kosslyn (1989) investigation into the separation of "what" and "where" processing in the human brain. This was based on the belief that visual perception involves two distinct cortical pathways (Mishkin, Ungerleider & Macko, 1983) – one running ventrally for identifying objects ("what"), and another running dorsally for determining their spatial locations ("where"). Alternative accounts of the distinction based on "perception and action" or "what and how" (e.g., Goodale & Milner, 1992; Milner & Goodale, 1995), or "planning and control" (e.g., Glover, 2003), or even "semantic and pragmatic" (e.g., Jeannerod, 1997), rather than "what and where" (Ungerleider & Haxby, 1994), have been suggested, but for present purposes this is not important. There is no doubt that the human brain and its functions are much more complicated than current models, and that various accidents of evolutionary history have influenced its structure. What is important here is that there are two simple and well defined tasks based on the same inputs, and simulations can explore what advantages a modular system has over a fully distributed

system. Eventually, of course, one will have to address the complications of real brains, but that is not the best starting point for this kind of study. Rather, one should start with the simplest system possible, that has the minimum number of potential confounding factors. Rueckl et al. (1989) took a set of 81 simplified "retinal images" (5 × 5 binary arrays) as inputs to a standard feed-forward neural network and trained it with a standard gradient descent based learning algorithm to classify each image as one of nine 3 × 3 binary patterns (i.e. "what"), in one of nine positions (i.e. "where"). By analyzing the performance of the trained networks, they demonstrated that modular networks were able to generate more efficient internal representations than fully distributed networks, and that they learned more easily how to perform the two tasks. The implication was that a process of evolution by natural selection, maybe also involving some form of lifetime learning, would result in a modular architecture, and hence answer the question of why modularity had arisen.

The obvious next step was to simulate evolutionary processes for the Rueckl et al. (1989) style networks and watch the modularity emerge. Although such simulations (of the form described in the following sections) did show that modularity could evolve if the learning and performance were based on the same Sum-Squared Error (SSE) measure as used by Rueckl et al. (1989), they also showed that even better *non*-modular systems could emerge if they were based on the Cross Entropy (CE) error measure (Hinton, 1989), thus throwing this whole approach into doubt (Bullinaria, 2001). Other evolutionary neural network simulations involving the same what-where tasks (Di Ferdinando, Calabretta & Parisi, 2001; Calabretta, Di Ferinando, Wagner & Parisi, 2003) confirmed the increasingly widespread belief that for complex tasks it is most efficient to have the neural architectures largely innate and the connection weights largely learned (e.g., Elman et al., 1996). These simulations also elucidated further the emergence of modularity in the SSE case, but they did not consider CE based learning.

In another series of evolutionary neural network simulations, Hüsken, Igel & Toussaint (2002) introduced finer grained measures of modularity and again found that the requirement for fast learning increased the selective pressure for modularity in the SSE case, but could not reproduce those results for the CE case. Most recently, Bowers & Bullinaria (2005) took a computational embryogeny approach to model the evolution of modularity at an even lower level of description, involving neural stem cells and

connections growing along simulated chemical gradients. In these simulations, no sign of modularity emerged until limits were placed on the connection lengths and the output neurons corresponding to the two tasks were physically separated by sufficient distances. This was consistent with the consequences of the bias towards short range neural connectivity discussed by Jacobs & Jordan (1992). However, a major problem with incorporating such physical constraints into the models, is that it is not always clear whether the physical structures have emerged as an efficient way to implement the best possible architectures, or whether the emergent architectures are simply the best that can be achieved with the available physical structures that have been constrained by other factors.

In a non-evolutionary study, again using the same simplified what-where task, Jacobs, Jordan & Barto (1991) explored task decomposition and modularity through learning in gated mixtures of experts networks. These systems are comprised of a set of separate "expert" networks/modules whose outputs are combined according to the outputs of a "gating network" that controls how the "experts" are used. This arrangement was argued to provide advantages in terms of learning speed, minimization of cross-talk (i.e. spatial interference), minimization of forgetting (i.e. temporal interference), and generalization. However, it is difficult to set up these systems in such a way that there is no inherent bias towards modularity. Moreover, it seems that if one does remove the bias towards modularity, and evolves all the learning and architecture parameters, the same SSE versus CE differences emerge as in the standard networks (Bullinaria, 2002). By restricting the neural architecture in this study to be the simplified feed-forward structure shown in Figure 1, one can be sure of avoiding any biases. If necessary, the issue of how any associated gating mechanisms might operate and be implemented in brains can be returned to once any advantage of modularity has been established.

The above brief review has identified three specific factors that clearly need further study:

1. The dependence on the learning algorithm. In particular, the cause of the observed differences in structures emerging for the SSE and CE based error measures, and how to choose which is the most appropriate to use.
2. The effect of physical constraints. In particular, the factors that affect neural connectivity other than computational efficiency.

3. The dependence on the task. In particular, to what extent do the results for the simplified what-where task used in all the earlier studies extend to more realistic tasks.

The remainder of this paper will address each of these issues.

4. Evolving Modular Neural Networks

The aim here is to use simulated evolution to establish the best neural architectures for particular applications. The general procedures for evolving neural networks are now well established (e.g., Yao, 1999; Cantû-Paz & Kamath, 2005). One takes a whole population of individual neural networks and allows them to learn, procreate and die in a manner approximating these processes in real biological systems. Each individual is "born" with a genotype, representing all the appropriate innate parameters, that is derived from the genotypes of its two parents. Then, throughout its "life", it learns from its environment how best to adjust its connection weights to perform most effectively. The fittest individuals will tend to live longest and produce the most children. Repeated natural selection of this form allows useful innate characteristics to proliferate in the population, and fitness levels improve towards some (possibly local) optimum.

There are two broad types of evolutionary simulation one might use: generational approaches in which the populations are updated one generation at a time (e.g., Yao, 1999), and more biologically inspired approaches with populations of competing learning individuals of all ages, each with the potential for dying or procreation at each stage. Previously, it seemed natural to use the biologically inspired approach for simulating brain evolution (Bullinaria, 2001). However, a detailed comparison of the two approaches for evolving good neural network learners revealed that, unlike the generational approach, the biologically inspired approach frequently converges to far from optimal configurations if poor choices of the various evolutionary parameter values are made, and that when one manages to avoid that problem, the resultant networks are very similar to those produced by the generational approach (Bullinaria, 2004). For this reason, a generational approach is used in this study. This also allows comparisons with the Bullinaria (2001) results to confirm the robustness of the results with respect to the evolutionary details.

The simulated genotype should represent all the parameters necessary to specify the neural network details, such as the architecture, the learning algorithm, the learning rates, the initial connection weights, and so on. In real biological evolution, all these parameters will be free to evolve. In simulations that are designed to explore particular issues, it makes more sense to fix some of these parameters, and thus avoid the complication of unforeseen interactions, and also speed up the simulations. For example, in a study designed to investigate the Baldwin effect (i.e. the interaction of learning and evolution and genetic assimilation) in control systems (Bullinaria, 2003b), it made sense to keep the architecture fixed and allow the learning rates and innate initial connection weights to evolve. Here it is more appropriate to allow the architecture to evolve, and have each individual start with random initial connection weights drawn from innately specified distributions. Then, since the optimal learning parameters are likely to depend on the architecture, these should be allowed to evolve along with the architecture.

Thus, at the beginning of each simulated generation, all the individual neural networks are "born" with random weights drawn from their own innately specified initial weight distributions, and each then learns from its own experience with the environment (i.e. training/testing data sets for the specified tasks). For biological populations, the ability of an individual to survive or reproduce will rely on a number of factors which will usually depend in a complicated manner on their performance over a range of related tasks (feeding, fighting, fleeing, and so on). For the purposes of the simplified models studied here, it is more appropriate to assume a simple relation between the main task performance measure and the procreation fitness. For each new generation, the children are produced from the fittest individuals using appropriate forms of cross-over and mutation.

The simplest starting point for studying modularity is to have one set of inputs and two tasks processed by the standard feed-forward neural network shown in Figure 1, where the arrows represent full connectivity between blocks of processing units, and architecture parameters *Nhid1*, *Nhid12* and *Nhid2* specify how many hidden units connect to each set of output units. The idea is that simulated evolution will find the values for these parameters that result in the best network performance. If *Nhid12* tends to zero, the architecture is totally modular, with modules consisting of a separate set of hidden units dedicated to each of the two tasks. If *Nhid1* and *Nhid2* both tend to zero, the architecture is totally non-

modular, with the processing of both tasks distributed across all the hidden units. For reasons that will become clear later, the total number of hidden units *Nhid* is kept fixed, leaving two free architecture parameters *Nhid1* + *Nhid12* and *Nhid2* + *Nhid12* corresponding to the total number of hidden units connecting to each output block. Then, since the best values for all the learning parameters are likely to depend on the architecture, and vice-versa, we also need to evolve the random initial weight distributions $[-l_L, +u_L]$ and gradient descent learning rates η_L for the four network components L (input to hidden weights *IH*, hidden unit biases *HB*, hidden to output weights *HO*, and output biases *OB*). The genotypes thus represent a total of 14 evolvable numerical parameters, each directly encoded as positive real numbers, with rounding applied to produce the integer hidden unit numbers. Ultimately, the simulations might also benefit from more biologically realistic encodings of the parameters, concepts such as recessive and dominant genes, learning and procreation costs, better inheritance and mutation details, different survival and procreation criteria, more restrictive mate selection regimes, offspring protection, more sophisticated learning algorithms and fitness functions, and so on, but for the purposes of this paper, the above simplified approach should be adequate.

5. Baseline Simulation Results

To avoid having to repeat the extensive internal representation analyses carried out by Rueckl et al. (1989), this study will begin with the same simplified "what-where" tasks that they used (with nine 3×3 patterns that may appear in nine positions in a 5×5 input space), and explore whether the advantages of modularity they observed will be sufficient to drive the evolution of modularity. Fitness here corresponds to the number of training epochs required to correctly classify all 81 input patterns. The simulation results are found to be extremely robust with respect to the details of the evolutionary processes, which are chosen here to produce clear results with the minimum drain on computational resources. All the results presented are for populations of 100 individuals, which was a trade-off between maintaining genetic diversity and running the simulations reasonably quickly. Each new generation is populated by children of the fittest half of the previous generation. There are many ways this can be done, but as long as population diversity is maintained, the details generally only affect the speed of evolution, rather than

what eventually emerges. Comparison of the baseline results obtained here with those from the biologically inspired evolution of Bullinaria (2001) provides explicit confirmation of the robustness with respect to the evolutionary details. Here, to speed the evolution, a fairly strong form of elitism was used, with half the children generated from just one parent (copying their parent's innate parameters), and the other half generated more conventionally from two parents (with each parameter value chosen randomly from the range spanned by their two parents, plus random Gaussian mutations that allow parameters outside that range). No learned information is carried between generations – at each generation, each individual starts with new random initial weights drawn from its own innate distribution.

Earlier studies (e.g., Bullinaria, 2003b) have shown empirically that the evolution can depend strongly on the initial conditions, i.e. the distribution of innate parameters across the initial population, and that the populations settle into near optimal states more quickly and reliably if they begin with a fairly wide distribution of initial parameters, rather than expecting mutations to carry the system from a relatively uniform state in which there is very little learning at all. Consequently, the initial populations were started with all the innate learning and initial weight parameters chosen randomly from ranges that spanned those values generally used in hand-crafted networks, namely [0, 4]. This ensured a good chance of starting from a range of reasonably competent individuals. The evolutionary process then continued until all the evolving parameters had clearly settled down.

For ease of comparison against earlier and later results, it is appropriate to begin the current study by establishing some baseline results for 32 hidden units, presenting averages and variances over ten independent evolutionary runs. Figure 2 shows how the learning rates and architecture evolve for the Rueckl et al. (1989) learning process, which uses the SSE gradient descent error function with the binary output targets offset to 0.1 and 0.9. As found previously, a purely modular architecture emerges, with a much higher proportion of the hidden units used for the harder "what" task. In Figure 3, the corresponding results for the CE cost function with binary output targets show that somewhat different learning rates are appropriate there, and that now a purely non-modular architecture emerges. The difference in evolved learning rates confirms the need to evolve good parameter values for each cost function, rather than attempting to perform comparisons using equal values for each case. The evolution

of the initial weight distributions do not tell us much, apart from the fact that, like the learning rates, they differ somewhat between network components, and from the values traditionally employed in hand-crafted networks. On the left of Figure 4 is shown the corresponding learning performances of the evolved individuals, with averages and variances over 32 learning runs for each individual from the 10 evolved populations after 6000 generations. The non-modular CE individuals are seen to perform significantly better than the modular SSE individuals.

6. Varying the Learning Algorithm

The baseline simulation results presented in the previous section constitute a confirmation of the main findings of Bullinaria (2001), and provide a check of their robustness with respect to the details of the evolutionary processes employed. A potential difficulty, however, lies in the use of non-binary output targets for the SSE approach, which is now known to be a less than optimal way to proceed (Bullinaria, 2003a). There are two common variations on the SSE theme that also need to be investigated: first using Pure SSE with the same binary targets as in the CE case, and second adding a so-called Sigmoid Prime Offset (SPO) of 0.1 into the output sigmoid derivative term in the weight update equations instead of offsetting the binary targets (Fahlman, 1988; Bullinaria, 2003a). Repeating all the evolutionary simulations using these two alternative approaches results in the evolved learning performances shown on the right of Figure 4. The Pure SSE case has much worse performance than SSE with offset targets, and again purely modular architectures emerges. For the SSE+SPO case, the performance is much better than for SSE with offset targets, but not as quite as good as the CE case. In this case, purely non-modular architectures emerge, as for the CE simulations before.

The importance of having the right neural network architecture can be confirmed by repeating all the evolutionary runs with the architecture fixed to be opposite (in the sense of purely modular versus purely non-modular) to that which emerged when it was free to evolve. The mean learning times and variances across 10,000 individual runs of each of the four learning algorithms, for evolved and opposite architectures, are shown on the left of Figure 5. The differences between learning algorithms are all statistically significant, and for each learning algorithm, the performance is significantly worse if the

wrong architecture is used. On the right of Figure 5 is shown a contour plot of how the performance degrades for the CE case away from the architecture optimum at the apex of the triangle.

The neural network simulations so far have shown that modularity is advantageous for the simplified what-where problem if the SSE with target offset cost function is used for learning, as in the Rueckl et al. (1989) study, or if a pure SSE learning algorithm is used. However, if the CE cost function is used, or SSE with an SPO, the performance is best with a purely non-modular architecture, and it remains better than the other learning algorithms even if a modular architecture is used. For each case there is a trade-off between employing modularity to reduce the cross-task interference, and the additional flexibility and free parameters arising from the full connectivity of non-modular architectures. The obvious question is: why does the trade-off favor modularity in some cases but not others?

There is actually a well known problem with using the SSE cost function in neural networks with sigmoidal outputs and binary targets, which is why the SSE learning algorithm variations exist. During learning, the gradient descent weight updates are proportional to the output sigmoid derivatives, which become close to zero near totally incorrect outputs, as well as for correct outputs. This means that if the weight updates from distinct training patterns interfere with each other in such a way as to cause incorrect outputs, as will tend to happen when learning multiple tasks in non-modular networks, correcting them later will be relatively difficult (Anand, Mehrotra, Mohan & Ranka, 1995). Attempts to evolve solutions to this problem for general single task binary mappings consistently resulted in the SSE learning algorithm *evolving into* the CE learning algorithm (Bullinaria, 2003a). For the CE cost function, the problematic sigmoid derivatives cancel out of the weight update equations, and there are also good theoretical reasons why CE is more appropriate for classification tasks anyway (Hinton, 1989; Bishop, 1995). It is understandable then, why the evolutionary simulations find that the interference prone SSE case favors modularity, while the superior CE algorithm is sufficiently interference free that it is able to make good use of the extra flexibility of non-modularity. Offsetting the output targets is aimed at keeping the network outputs away from the zero derivative regions, but it is only partially successful (Bullinaria, 2003a), so modularity is still preferred in that case. Adding a small SPO to the sigmoid derivatives is a more direct approach for preventing them from going to zero, and this is successful at preventing enough

of the interference problems to render useable the advantages of non-modularity, and comes close to reaching the performance levels of CE. The remainder of this paper will present a further series of evolutionary simulations to explore whether non-modularity is always the preferred option when the most efficient learning algorithms are employed.

7. Evolving the Learning Algorithm

In all the above simulations, the learning algorithm was fixed to be standard gradient descent learning using either the CE cost function or some variation of SSE. Now, to make sure that the best possible learning algorithm is used in all the subsequent simulations, the learning algorithm itself will also be evolved. This can be done by using a cost function that is an evolvable linear combination of SSE and CE, which (because the coefficients must remain non-negative and an overall scale factor can be absorbed into the evolvable learning rates) can always be written without loss of generality as

$$E = (1-\mu)E_{SSE} + \mu E_{CE}$$

with the parameter μ bounded to lie in the range [0, 1]. This parameterization is deliberately different to that of Bullinaria (2003a), to provide another check of the robustness of the results with respect to the implementational details. Although the SSE and CE cost functions have rather different mathematical forms, the gradient descent weight updates are proportional to their derivatives which differ only by the problematic output sigmoid derivative term $(1-o_j)o_j$, so the combined weight w_{ij} update equation for the connection between hidden unit i and output unit j can be written

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}} = -\eta h_i (t_j - o_j)\Big[(1-\mu)(1-o_j)o_j \; + \; \mu\Big]$$

where t_j is the target output, o_j is the actual output, h_i is the hidden unit activation, and η is the learning rate. The evolvable parameter μ has extreme values of 0 and 1 corresponding to the pure SSE and CE learning algorithms, and an intermediate value of around 0.1 corresponds to the traditional Sigmoid Prime Offset approach for avoiding the SSE learning problem (Fahlman, 1988; Bullinaria, 2003a). If we continue to keep the total number of hidden units fixed, that gives a total of two architecture and thirteen

learning parameters to evolve. We now simply have to repeat the above evolutionary simulations, using the same Rueckl et al. (1989) what-where training data, with the new cost function E and associated evolvable innate parameter μ.

Figure 6 shows the new simulation results for neural networks with 36 hidden units, with mean values and standard deviations over ten runs. The parameter μ takes on values very close to one, corresponding to a purely CE learning algorithm, and the evolved architecture parameters again correspond to purely non-modular networks. All the final evolved learning rates and initial weight distributions are indistinguishable from those found in the CE runs before. Together, the evolved parameters result in the training data being learned in around 18 epochs, and provide a solid confirmation of the earlier results that the requirement for faster learning on this what-where task leads reliably to the emergence of *non-modular* neural architectures.

8. Physical Constraints on Neural Connectivity

When building cognitive models, it is naturally important to take into account the physical properties of the brain, in addition to the computations it is performing. However, in understanding the reasons for particular brain structures, it is also important to distinguish between the physical properties that really do constrain those structures and the computations they perform, and those physical properties that could potentially be different if the associated computational advantages were sufficient to cause them to evolve. This is why one needs to explore the models both with and without the constraints applied.

Perhaps the most obvious physical brain property is its finite size, and so an important factor to consider is the dependence on the computational power of the neural network compared with the complexity of the task being learned. It has certainly been found elsewhere in neuropsychological modelling that neural networks can behave rather differently when they have barely enough hidden units to carry out the given tasks, compared to when they have plenty of spare resources (e.g., Bullinaria & Chater, 1995; Bullinaria, 2005). Moreover, Ballard (1986) has argued that limitations on the number of available neurons can lead to advantages for modularity for representing complex high dimensional spaces. It is reasonably straightforward to check this by repeating the above simulations with different

total numbers of hidden units. Figure 7 shows how the evolved network architecture and performance vary with the computational power. On the left we see that the evolved architectures remain non-modular from the minimal network required to perform the given task (9 hidden units) right up to over a hundred times that size (1000 units). On the right we see how the required number of epochs of training decreases as more computational power is made available. This is why the total number of hidden units has been fixed in all the evolutionary simulations – otherwise the evolution of faster learning would just keep on increasing it, and the evolutionary process would slow down (in real time on a non-parallel processor) due to the increased computations required, rather than settling down because an optimal configuration had emerged. Real brains, of course, are parallel processors and their sizes are constrained by various physical factors, such as growth costs, energy/oxygen consumption, heat dissipation, and such like. Kaas (2000) considers the design problems that are faced as brains get bigger or smaller, and discusses why brain sizes should vary so much, and have appropriate sizes that depend on the type and size of animal and its particular environment. It makes good sense, therefore, for the number of hidden units to be fixed in all the models at some appropriate value. The fact that the optimality of non-modular architectures for the what-where task is so robust with respect to network complexity, means that there is no need to be too concerned about fixing that simulated brain size here with uncertain accuracy.

A related physical constraint of importance here derives from the fact that the significant volume occupied by neural connections (i.e. axons and dendrites) precludes full neural connectivity (e.g., Chklovskii et al., 2002; Sporns et al., 2004). The relevant biological factors and their consequences for brain structure have been discussed by Stevens (1989), Kaas (2000), Changizi (2001) and Karbowski (2003), and Ringo (1991) has presented an explicit model which shows why the degree of neural connectivity should decrease as brain size is increased. Perhaps the most obvious approach to minimize the volume of connections would be to keep them as short as possible. Jacobs & Jordan (1992) and Bowers & Bullinaria (2005) have already looked at the emergence of restricted connectivity resulting from a bias towards short connections in neural network models where the neurons have positions specified in a three dimensional space. However, it is difficult to model the evolution of such details without introducing an inherent bias towards modularity, so instead we shall here explore whether

modularity will emerge simply from restrictions on the *proportion* of connections, without regard to the neuron positions and connection lengths. With a given optimal pattern of connectivity, evolution will surely arrange the neurons and connections to minimize the various costs of the connections (Chklovskii, 2004), but restrictions on the connectivity proportions alone could be sufficient to drive the evolution of modularity.

The above simulations can easily be extended to test these ideas – one simply has to repeat them with the degree of connectivity between layers restricted to be no more than some fraction f of full connectivity. The easiest way this might be implemented is to allow only a random subset of the possible connections between each block of units, and let the learning algorithm determine how to use those connections most effectively. For the architecture we have been evolving, shown in Figure 1, the total connectivity fraction for the hidden to output layer is

$$f = \frac{\left[(Nhid1 + Nhid12).Nout1 + (Nhid2 + Nhid12).Nout2 \right].f_{HO}}{Nhid.(Nout1 + Nout2)}$$

Assuming each hidden unit is connected to at least one output unit, and the total number of hidden units $Nhid = Nhid1 + Nhid12 + Nhid2$ is fixed, this can be reduced either by reducing the connectivity proportion f_{HO} between the blocks of units, or by reducing $Nhid12$ which corresponds to increasing the degree of modularity. The choice is effectively between randomly removing connections from anywhere, or systematically removing connections from hidden units that contribute to both output tasks. If a hidden unit can usefully contribute to both tasks, it is likely to be efficient to keep the corresponding connections, but if the two tasks are sufficiently different that they cause interference in the learning process for the common hidden units, it will be more efficient to remove connections so that each hidden unit only contributes to a single task. The architectures that we can expect to emerge from evolution will thus depend on the details of the tasks involved, but it seems likely that modularity will emerge for qualitatively different tasks, such as the simplified what-where tasks we have been studying.

Figure 8 shows the architectures that emerge when the above simplified what-where network is evolved with $Nhid = 72$ hidden units in total. As f is reduced, the number of hidden units shared by both

output tasks, *Nhid12*, falls almost linearly until *f* reaches one half, and then it stays close to zero for all lower levels of connectivity. Since the connectivity proportion in real brains is known to be considerably less than one half (Chklovskii et al., 2002), this means that a modular architecture will make the most efficient use of the available connections if they are limited to the extent that is found in brains. As one would predict, Figure 8 also shows that the number of epochs of training required rises sharply as the connectivity proportion reaches a level close to the minimum number of connections necessary to perform the given tasks. Increasing the total number of hidden units allows efficient processing at lower connectivity levels, but modularity remains the preferred architecture for connectivity proportions below one half. Throughout, *Nhid2*, corresponding to the easier "where" task, is lower than *Nhid1*, as was found in the modular SSE simulations (Bullinaria, 2001) and the original Rueckl et al. (1989) study, but the relative size of the two modules varies slightly with the connectivity proportion.

9. More Realistic Learning Tasks

Having established that modularity will only be an advantage for learning the what-where task when there are constraints upon the proportion of neural connectivity, there remains the obvious question of whether that will be true for all tasks. A particular concern is that learning a small set of input-output mappings for the simplified what-where task is very different to most realistic human cognitive tasks in which we are typically required to generalize from, and respond to, an unpredictable stream of inputs drawn from continuous data distributions. Moreover, it has been suggested elsewhere that modularity is crucial to obtain good generalization for some complex tasks, such as in the connectionist sentence production model of Chang (2002).

A typical type of task humans have to cope with is the classification in various ways of novel input data drawn from some continuous distribution, by learning to generalize from different examples they have experienced before. To keep things simple for simulation purposes, suppose we have just two continuous valued inputs that have been normalized to lie in the range [0, 1], and we need to perform two distinct classifications based on those input values. For example, the inputs might correspond to two crucial measurable characteristics of animals, and the two output tasks could be to classify them as being

good food (or not) and dangerous (or not). The neural networks are required to learn the classification boundaries in the two dimensional input space for each output task, from a continuous stream of examples. Obviously, even for this simplified set-up, there are still an infinite number of possible tasks corresponding to the different possible classification boundaries. What we need to establish is whether a separate module for each output task consistently works better or worse than a fully distributed network, rather than the advantage of modularity being problem dependent. One can attempt to answer that question by repeating all the above simulations with everything else the same except for the training data and the fitness measure. Here the fitness is the ability to learn quickly to generalize, i.e. to produce the right output for each new item *before* training on it, rather than producing the right output for each item after many epochs of training on it. In practice, it was convenient to present the infinite sequence of possible input patterns in blocks (or epochs) of 400 training items, and measure the fitness as the number of blocks required before a full block of items was classified correctly before training on each item. The precise details of this regime are not crucial; for example, increasing or decreasing the block size by a factor of two does not produce qualitatively different results.

It did not require many evolutionary simulations to determine that the advantage of modularity *is* problem dependent, and that the advantage depends on many factors, in particular, the overlap of the two classification tasks, the relative difficulties of the two tasks, the complexity of the decision boundaries, and the number of classes. The two simple cases shown in Figure 9 are representative of the two patterns of results that emerge for networks with 200 hidden units. The case on the left involves one two class task and one three class task. For full neural connectivity, the optimal architecture that emerges is non-modular, and as the degree of neural connectivity is reduced, the degree of modularity increases, as we found for the what-where case earlier. The case on the right consists of two two class tasks. Here, a modular architecture is found to evolve for any degree of neural connectivity. As one would expect from the earlier results, for both cases the average amount of training data required to reach a block of perfect performance decreases as the connectivity, and hence the computational power, is increased. Since the evolved architectures for both cases are consistently modular below a connectivity proportion of one half, and the proportions in real brains are considerably lower than that (overall, at least), it seems that the

evolutionary simulations do provide a consistent understanding of why modularity should emerge.

A final complication is the need to check again that the evolutionary simulations are not converging on architectures that actually perform worse than the other possibilities. To do this, a representative subset of the simulations were repeated with the architecture constrained to be modular, and again with it constrained to be non-modular. These runs confirmed that the evolved architectures did indeed produce the best performance for each task.

10. Discussion and Conclusions

This paper began by reviewing the previous attempts to understand the advantages of modularity in neural systems, which evolution, or evolution plus lifetime learning, might be expected to translate into the brain structures familiar to cognitive scientists. There was a clear need for further exploration into the effects of: the choice of learning algorithm, the type of task, and the incorporation of physical constraints. An approach was described for simulating the evolution of artificial neural networks in which modularity could be defined as "specialized sub-sets of hidden units" in standard feed-forward networks, and this was used to confirm the main results of the previous studies. Those simulations were then extended by allowing the neural learning algorithms to evolve alongside the architectures, and by investigating more realistic learning tasks. It was found that for many tasks there is no learning advantage for modularity because the reduction in cross-task interference that modularity provides is out-weighed by the extra computational power allowed by full connectivity. For other tasks, the problem of interference is more important than the computational power, and modularity does evolve. For artificial systems then, the usefulness of modularity is application dependent, and it seems difficult to formulate general purpose heuristics that tell us when modularity is likely to be an advantage.

Cognitive scientists more interested in understanding biological brains, than building artificial systems, will need to have their models further constrained by various physical properties. Two particular aspects were investigated: limits on the total number of neurons, and limits on the degree of neural connectivity. The earlier results were found to be robust with respect to the number of neurons, but for connectivity proportions less than one half, modular architectures were found to have a clear advantage in

terms of learning efficiency, and simulated evolution led to the emergence of modular structures for all the pairs of simplified tasks considered. Since the degree of neural connectivity in real brains is considerably less than one half, this implies a clear reason why modular architectures should exist in the brain, and ties in with existing suggestions in the literature that increasing modularity is a practical solution to the difficulties of building larger brains (e.g., Kaas, 2000). Of course, the computations carried out by real brains are far removed from the simplified pairs of tasks simulated in this paper, and the patterns of connectivity will inevitably need to be correspondingly more complex, quite possibly with *small world network* structure (Watts & Strogatz, 1998; Sporns et al., 2004). It will certainly be instructive in the future to extend the simulations of this paper to investigate exactly what types of neural configurations emerge to deal with more complex information processing tasks.

Interestingly, the existing simulation results (e.g., Figures 8 and 9) show that the learning performance increases with the degree of connectivity, and so if this were free to evolve too, we should expect it to increase towards full connectivity if that were physically possible. Chklovskii et al. (2002) have considered some of the ways evolution might have optimized neural circuits (to minimize conduction delays, signal attenuation, and connection lengths) and concluded that approximately 60% of the space should be taken up by wiring (i.e. axons and dendrites), which is close to the proportion actually found. This in turn will place hard constraints on the connectivity proportions, and it is not clear if any amount of evolutionary pressure would be able to overcome them (Chklovskii, 2004), except perhaps in the simplest possible brain structures. In fact, further simulations indicate that, as the number of hidden units increases, the performance versus connectivity relation becomes increasingly flat, so for structures with human brain-like numbers of neurons, the evolutionary pressure towards increased connectivity may be relatively low anyway, at least for connectivity levels in the range where non-modular architectures might be advantageous.

It is important to remember that the simplified gradient descent learning algorithms employed in this study are unlikely to bear much resemblance to the learning mechanisms of real brains. Past comparisons of the resultant neural representations (e.g., Plaut & Shallice, 1993) have indicated that the results obtained from such simplified algorithms *do* give a good indication of what can be expected to emerge

from more biologically plausible learning mechanisms. However, it is not clear that such comparisons can be relied upon when factors such as learning speed are so important. Indeed, two aspects of the results presented in this paper show how crucial the choice of learning algorithm is; namely the dependence of the evolved architecture on learning algorithm seen in Sections 5 and 6, and the learning time histograms in Figure 5 indicating that the choice of learning algorithm is more important than having the right architecture. This learning algorithm dependence would have been problematic if the study had stopped there. The obvious argument then would have been that evolution will surely drive the brain towards the most efficient learning algorithm possible (as we saw in Section 7), and that would not have problems with cross task interference, and hence there would be no computational advantage for modularity, and no understanding of why it should exist in the brain. Going on and finding (as in Section 9) that modularity is advantageous for some, but not all, more realistic complex tasks, even with the most efficient artificial learning algorithm, then raises the question of what properties of real tasks will require modularity for real learning algorithms. Then, given the simplifications involved with the current models, we would have ended up with no useful answers at all. Fortunately, the issue of neural connectivity proportions has brought us to the conclusion that modularity evolves for *all* the tasks and learning algorithms studied, thus rendering the lack of biological plausibility less problematic.

The overall conclusion then seems to be that, despite the doubts raised by Bullinaria (2001), the reasons for the emergence of modularity in brains *can* be understood simply in terms of learning advantages in neural systems, *if* the appropriate physical constraints are taken into account. Understanding exactly how these modules actually come about through evolution and/or lifetime learning in real brains is something that still requires more detailed simulations. Moreover, having one good reason for why modularity should emerge does not preclude there being other reasons for some, or all, aspects of modularity in the brain. Indeed, there are numerous other reasons why modularity might be advantageous, for both artificial systems and biological brains, such as to facilitate problem decomposition and the re-use of common processing sub-systems across multiple tasks (e.g., Reisinger, Stanley & Miikkulainen, 2004; Kashtan & Alon, 2005), to allow improved generalization in complex multi-component tasks (e.g., Chang, 2002), to improve robustness and evolvability (e.g., Wagner, 1996),

and to cope better with changing environments (e.g., Lipson, Pollack & Suh, 2002; Kashtan & Alon, 2005). Nevertheless, the simulations presented in this paper have demonstrated how modularity can also be an advantage in much simpler systems, and it is likely that the evolutionary neural network framework presented here can also provide a sound foundation for investigating the emergence of modularity in more complex systems too. Those future models will inevitably have to include more realistic neural structures, connectivity patterns, growth mechanisms, and learning algorithms, as well as the incorporation of the relevant known stages in human brain evolution. They will clearly also need to involve classification tasks much more complex than those considered here, as well as all the other forms of information processing known to be carried out by brains. Finally, they will need to establish when modularity is *not* an advantage, and how all the "modules" should interact. It is hoped that progress in these directions will be reported in the near future.

Acknowledgements

A preliminary version of this work was presented at the 28th Annual Conference of the Cognitive Science Society in July 2006, and appeared in its proceedings. Three anonymous reviewers are thanked for their helpful comments on an earlier version of this paper.

References

Anand, R., Mehrotra, K., Mohan, C.K. & Ranka, S. (1995). Efficient classification for multiclass problems using modular neural networks. *IEEE Transactions on Neural Networks*, **6**, 117-124.

Baldwin, J.M. (1896). A new factor in evolution. *The American Naturalist*, **30**, 441-451.

Ballard, D.H. (1986). Cortical connections and parallel processing: Structure and function. *Behavioral and Brain Sciences*, **9**, 67-120.

Belew, R.K. & Mitchell, M. (Eds) (1996). *Adaptive individuals in evolving populations*. Reading, MA: Addison-Wesley.

Bishop, C.M. (1995). *Neural networks for pattern recognition*. Oxford, UK: Oxford University Press.

Bookheimer, S. (2002). Functional MRI of language: New approaches to understanding the cortical organization of semantic processing. *Annual Review of Neuroscience*, **25**, 151-188.

Bowers, C.P. & Bullinaria, J.A. (2005). Embryological modelling of the evolution of neural architecture. In A. Cangelosi, G. Bugmann & R. Borisyuk (Eds), *Modeling Language, Cognition and Action*, 375-384. Singapore: World Scientific.

Bullinaria, J.A. (1997). Analysing the internal representations of trained neural networks. In A. Browne (Ed.), *Neural Network Analysis, Architectures and Applications*, 3-26. Bristol, UK: Institute of Physics.

Bullinaria, J.A. (2001). Simulating the evolution of modular neural systems. In *Proceedings of the Twenty-third Annual Conference of the Cognitive Science Society,* 146-151. Mahwah, NJ: Lawrence Erlbaum.

Bullinaria, J.A. (2002). The evolution of gated sub-networks and modularity in the human brain. In J.A. Bullinaria & W. Lowe (Eds), *Connectionist Models of Cognition and Perception*, 27-39. Singapore: World Scientific.

Bullinaria, J.A. (2003a). Evolving efficient learning algorithms for binary mappings. *Neural Networks*, **16**, 793-800.

Bullinaria, J.A. (2003b). From biological models to the evolution of robot control systems. *Philosophical*

Transactions of the Royal Society of London A, **361**, 2145-2164.

Bullinaria, J.A. (2004). Generational versus steady-state evolution for optimizing neural network learning. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2004)*, 2297-2302. Piscataway, NJ: IEEE.

Bullinaria, J.A. (2005). Connectionist neuropsychology. In G. Houghton (Ed.), *Connectionist Models in Cognitive Psychology*, 83-111. Brighton, UK: Psychology Press.

Bullinaria, J.A. & Chater N. (1995). Connectionist modelling: Implications for cognitive neuropsychology. *Language and Cognitive Processes*, **10**, 227-264.

Calabretta, R., Di Ferinando, A., Wagner, G.P. & Parisi, D. (2003). What does it take to evolve behaviorally complex organisms? *BioSystems*, **69**, 245-262.

Cantû-Paz, E. & Kamath, C. (2005). An empirical comparison of combinations of evolutionary algorithms and neural networks for classification problems. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics,* **35**, 915-927.

Caramazza, A. (1986). On drawing inferences about the structure of normal cognitive systems from analysis of patterns of impaired performance: The case of single-patient studies. *Brain and Cognition*, **5**, 41-66.

Chang, F. (2002). Symbolically speaking: A connectionist model of sentence production, *Cognitive Science*, **26**, 609-651.

Changizi, M.A. (2001). Principles underlying mammalian neocortical scaling. *Biological Cybernetics*, **84**, 207-215.

Chklovskii, D.B. (2004). Exact solution for the optimal neuronal layout problem. *Neural Computation*, **16**, 2067-2078.

Chklovskii, D.B., Schikorski, T. & Stevens, C.F. (2002). Wiring optimization in cortical circuits. *Neuron*, **34**, 341-347.

Devlin, J.T., Gonnerman, L.M., Andersen, E.S. & Seidenberg, M.S. (1998). Category-specific semantic deficits in focal and widespread brain damage: A computational account. *Journal of Cognitive*

Neuroscience, **10**, 77-94.

Di Ferdinando, A., Calabretta, R, & Parisi. D. (2001). Evolving modular architectures for neural networks. In R.F. French & J.P. Sougne (Eds), *Connectionist Models of Learning, Development and Evolution*, 253-262. London: Springer-Verlag.

Dunn, J.C. & Kirsner, K. (1988). Discovering functionally independent mental processes: The principle of reversed association. *Psychological Review,* **95**, 91-101.

Ebbesson, S.O.E. (1984). Evolution and ontogeny of neural circuits. *Behavioral and Brain Sciences*, **7**, 321-366.

Elman, J.L., Bates, E.A., Johnson, M.H., Karmiloff-Smith, A., Parisi, D. & Plunkett, K. (1996). *Rethinking innateness: A connectionist perspective on development*. Cambridge, MA: MIT Press.

Fahlman, S.E. (1988). Faster learning variations of back propagation: An empirical study. In D. Touretzky, G.E. Hinton & T.J. Sejnowski (Eds), *Proceedings of the 1988 Connectionist Models Summer School*, 38-51. San Mateo, CA: Morgan Kaufmann.

Fodor, J.A. (1983). *The Modularity of the Mind*. Cambridge, MA: MIT Press.

Geary, D.C. & Huffman, K.J. (2002). Brain and cognitive evolution: Forms of modularity and functions of mind. *Psychological Bulletin*, **128**, 667-698.

Glover, S. (2004). Separate visual representations in the planning and control of action. *Behavioral and Brain Sciences*, 27, 3-24.

Goodale, M. A., & Milner, A. (1992). Separate visual pathways for perception and action. *Trends in Neuroscience,* **15**, 20-25.

Hinton, G.E. (1989). Connectionist learning procedures. *Artificial Intelligence*, **40**, 185-234.

Hinton, G.E. & Nowlan, S.J. (1987). How learning can guide evolution. *Complex Systems*, **1**, 495-502.

Hüsken, M., Igel, C. & Toussaint, M. (2002). Task-dependent evolution of modularity in neural networks. *Connection Science*, **14**, 219-229.

Huettel, S.A., Song, A.W. & McCarthy, G. (2004). *Functional magnetic resonance imaging*. Sunderland, MA: Sinauer Associates Incorporated.

Jacobs, R.A. (1999). Computational studies of the development of functionally specialized neural modules. *Trends in Cognitive Science*, **3**, 31-38.

Jacobs, R.A. & Jordan, M.I. (1992). Computational consequences of a bias toward short connections. *Journal of Cognitive Neuroscience,* **4**, 323-336.

Jacobs, R.A., Jordan, M.I. & Barto, A.G. (1991). Task decomposition through competition in modular connectionist architecture: The what and where vision tasks. *Cognitive Science,* **15**, 219-250.

Jeannerod, M. (1997). *The cognitive neuroscience of action*. Oxford, UK: Blackwell.

Kaas, J.H. (2000). Why is brain size so important: Design problems and solutions as neo-cortex gets bigger or smaller. *Brain and Mind*, **1**, 7-23.

Karbowski, J. (2003). How does connectivity between cortical areas depend on brain size? Implications for efficient computation. *Journal of Computational Neuroscience*, **15**, 347-356.

Kashtan, N & Alon, U. (2005). Spontaneous evolution of modularity and network motifs. *Proceedings of the National Academy of Sciences*, **102**, 13773-13778.

Lipson, H., Pollack, J.B. & Suh, N.P. (2002). On the origin of modular variation. *Evolution*, **56**, 1549-1556.

Marcus, G. (2004). *The birth of the mind*. New York, NY: Basic Books.

Milner, A.D., & Goodale, M.A. (1995). *The visual brain in action*. Oxford, UK: Oxford University Press.

Mishkin, M., Ungerleider, L.G. & Macko, K.A. (1983). Object vision and spatial vision: Two cortical pathways. *Trends in Neurosciences*, **6**, 414-417.

O'Leary, D.D.M. (1989). Do cortical areas emerge from a protocortex? *Trends in Neurosciences*, **12**, 400-406.

Plaut, D.C. (1995). Double dissociation without modularity: Evidence from connectionist neuropsychology. *Journal of Clinical and Experimental Neuropsychology*, **17**, 291-321.

Plaut, D.C. (2002). Graded modality-specific specialization in semantics: A computational account of optic aphasia. *Cognitive Neuropsychology*, **19**, 603-639.

Plaut, D.C. & Hinton, G.E. (1987). Learning sets of filters using back-propagation. *Computer Speech and*

Language, **2**, 35-61.

Plaut, D.C. & Shallice, T. (1993). Deep dyslexia: A case study of connectionist neuropsychology. *Cognitive Neuropsychology*, **10,** 377-500.

Quartz, S.R. (1999). The constructivist brain. *Trends in Cognitive Sciences*, **3**, 48-57.

Reisinger, J., Stanley, K.O. & Miikkulainen, R. (2004). Evolving reusable neural modules. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2004)*, 69-81. New York, NY: Springer-Verlag.

Ringo, J.L. (1991). Neuronal interconnection as a function of brain size. *Brain Behavior and Evolution*, **38**, 1-6.

Rueckl, J.G., Cave, K.R. & Kosslyn, S.M. (1989). Why are "what" and "where" processed by separate cortical visual systems? A computational investigation. *Journal of Cognitive Neuroscience*, **1**, 171-186.

Sartori, G. (1988). From neuropsychological data to theory and vice versa. In G. Denes, P. Bisiacchi, C. Semenza & E. Andrewsky (Eds), *Perspectives in Cognitive Neuropsychology*. London, UK: Erlbaum.

Seok, B. (2006). Diversity and unity of modularity. *Cognitive Science*, **30**, 347-380.

Shallice, T. (1988). *From neuropsychology to mental structure*. Cambridge, UK: Cambridge University Press.

Sporns, O., Chialvo, D.R., Kaiser, M. & Hilgetag, C.C. (2004). Organization, development and function of complex brain networks. *Trends in Cognitive Sciences,* **8***,* 418-425.

Stevens, C.F. (1989). How cortical interconnectedness varies with network size. *Neural Computation*, **1**, 473-479.

Teuber, H.L. (1955). Physiological psychology. *Annual Review of Psychology*, **6**, 267-296.

Ungerleider, L.G. & Haxby, J.V. (1994). 'What' and 'where' in the human brain. *Current Opinion in Neurobiology*, **4**, 157-165.

Van Orden, G.C., Pennington, B.F. & Stone, G.O. (2001). What do double dissociations prove? *Cognitive*

Science, **25**, 111-172.

Wagner, G.P. (1996). Homologues, natural kinds, and the evolution of modularity. *American Zoologist,* **36**, 36-43.

Watts, D.J. & Strogatz, S.H. (1998). Collective dynamics of 'small world' networks. *Nature*, **393**, 440-442.

Wood, C.C. (1978). Variations on a theme of Lashley: Lesion experiments on the neural model of Anderson, Silverstein, Ritz & Jones. *Psychological Review,* **85,** 582-591.

Yao, X. (1999). Evolving artificial neural networks. *Proceedings of the IEEE,* **87**, 1423-1447.

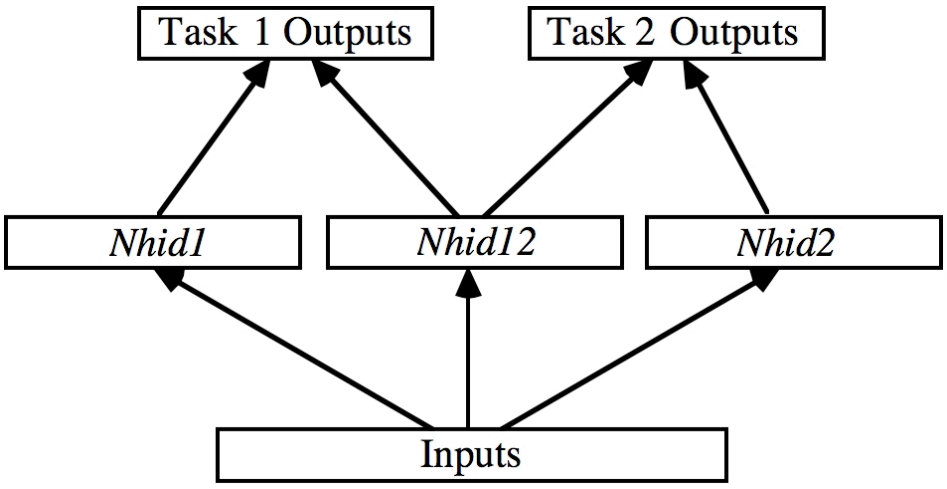

Figure 1: The simplified neural network architecture used to study the evolution of modularity. Arrows represent full connectivity between blocks of processing units.

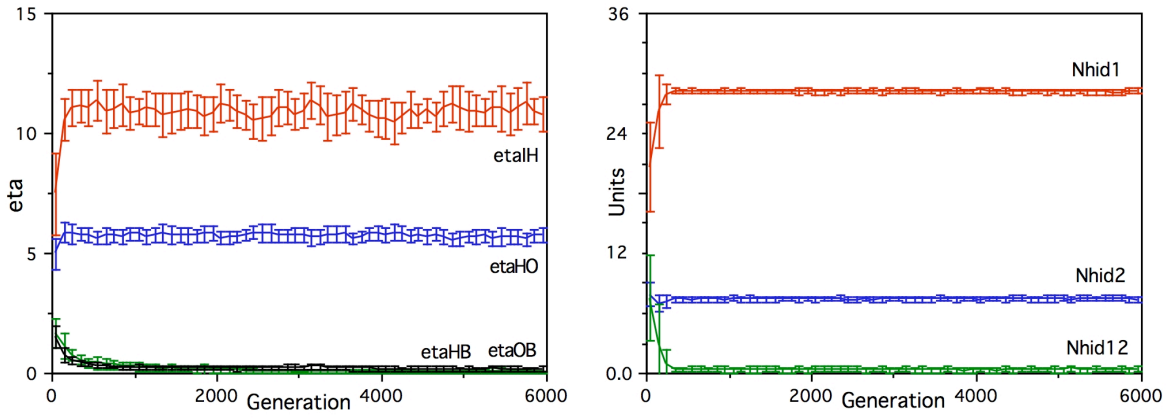

Figure 2: Evolution of the learning rates (left) and architecture parameters (right) when using the Sum-Squared Error cost function with offset targets (SSE). A pure modular architecture emerges.

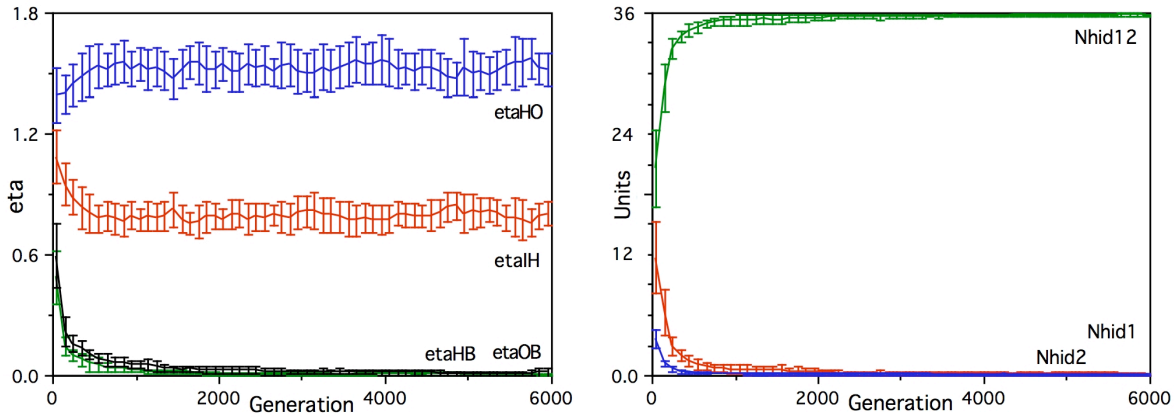

Figure 3: Evolution of the learning rates (left) and architecture parameters (right) when using the Cross Entropy cost function (CE). A fully distributed architecture emerges.

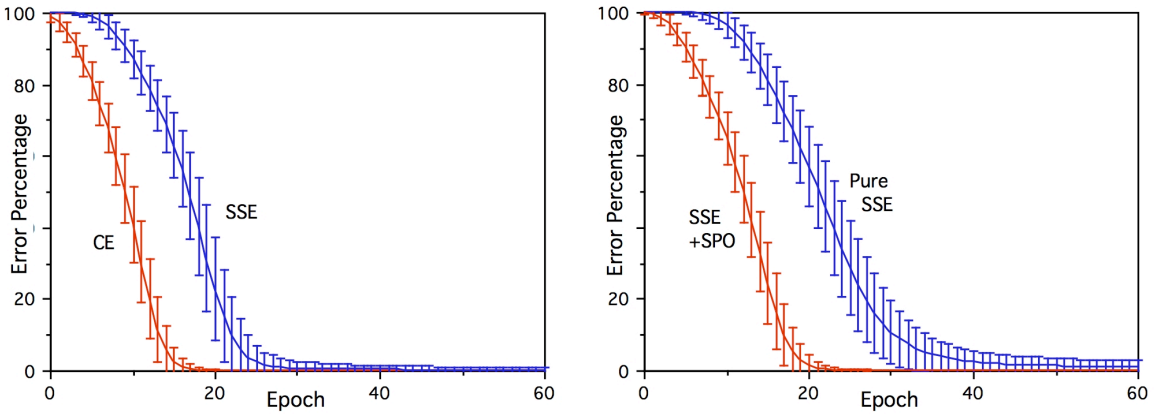

Figure 4: Learning performance of the evolved populations: baseline SSE and CE systems (left) and two variations of the SSE learning algorithm (right). Modularity emerges for the SSE and Pure SSE cases, while non-modular architectures evolve for the better performing CE and SSE+SPO cases.

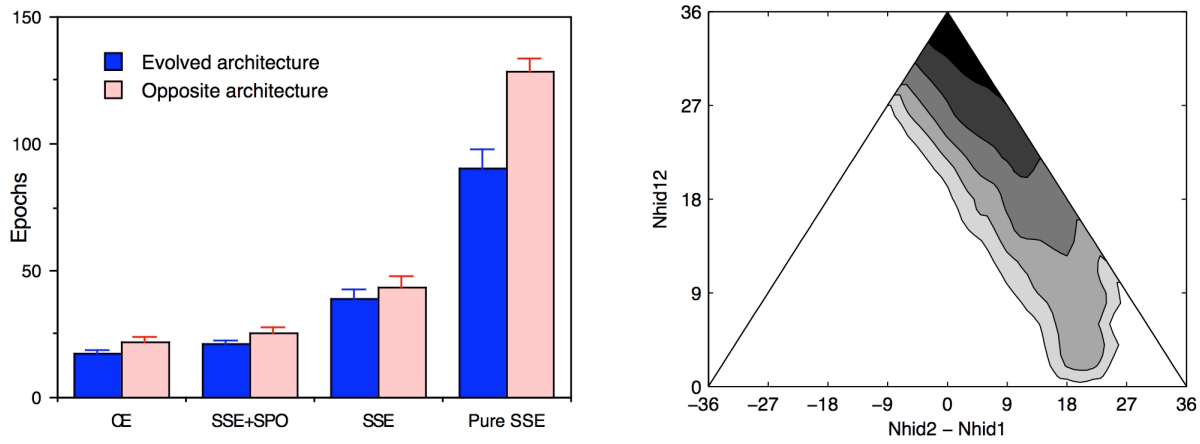

Figure 5: Learning times for the various learning algorithms with evolved and opposite architectures (left), and the variation of learning times with architecture for the CE learning algorithm (right).

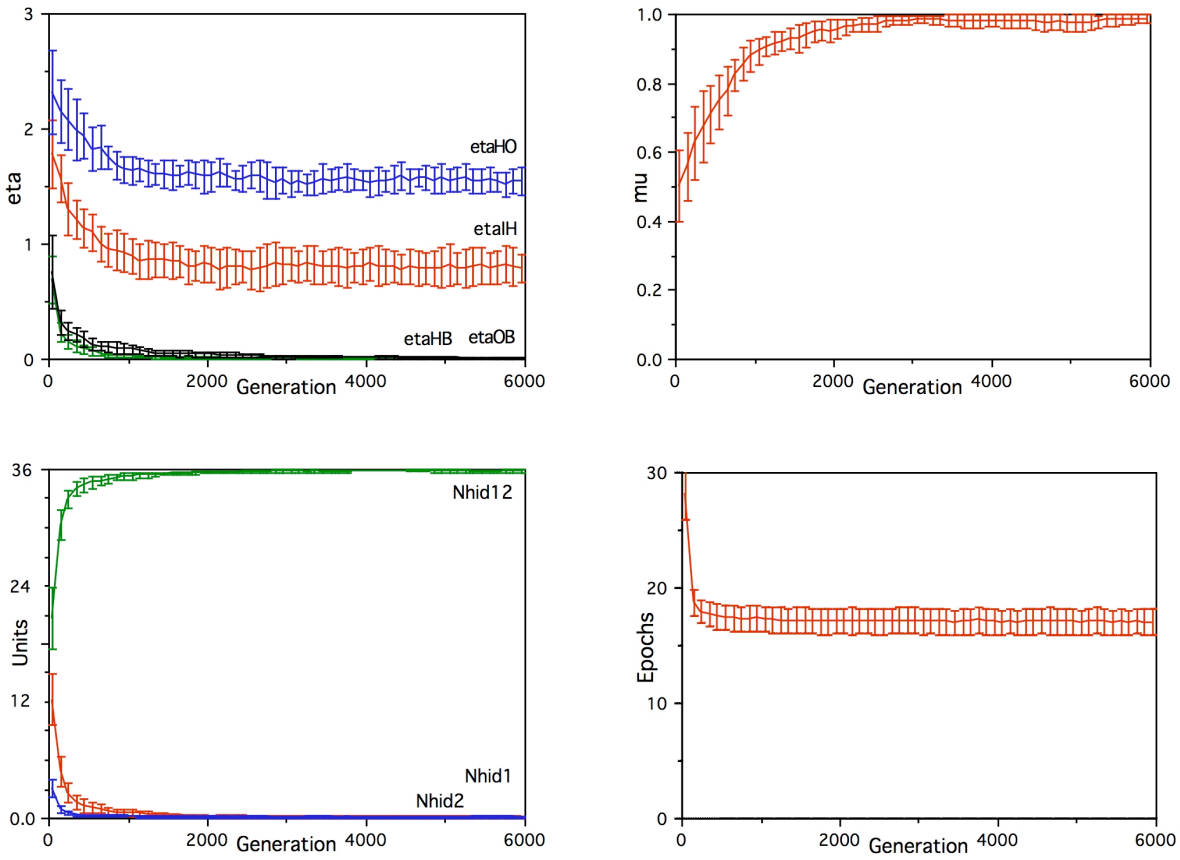

Figure 6: Evolution of the standard what-where neural network with 36 hidden units and an evolvable learning algorithm: the learning rates (top left), the CE versus SSE parameter μ (top right), the architecture parameters (bottom left), and the epochs of training required (bottom right). A pure CE learning algorithm and non-modular architecture emerge.

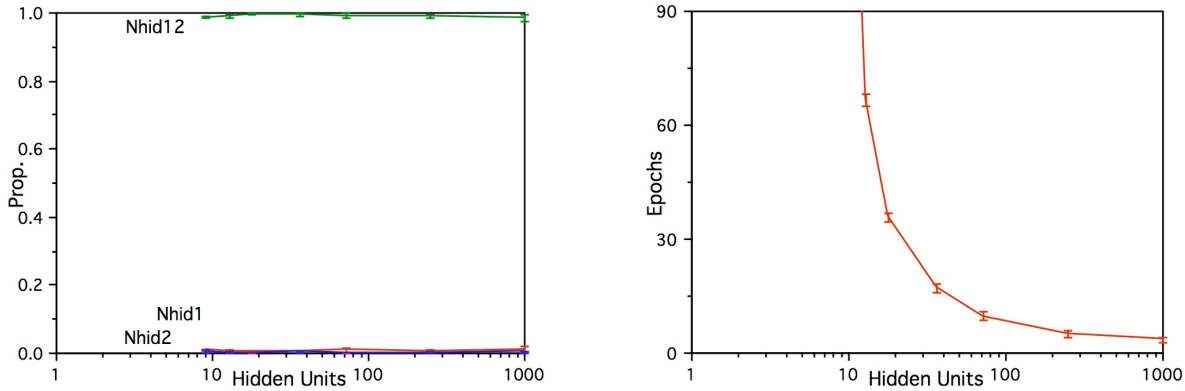

Figure 7: Dependence of the evolved what-where neural network results on the total number of hidden units. The architecture parameters as a proportion of the total number of hidden units (left), and the number of epochs of training required (right). Non-modularity emerges for any number of hidden units.

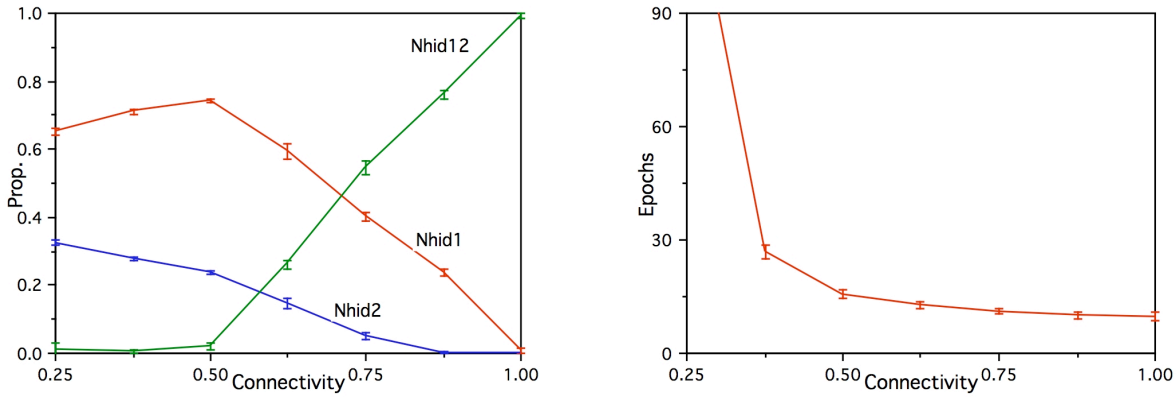

Figure 8: Dependence of the evolved what-where neural network results on the degree of connectivity between the network layers. The architecture parameters as proportions (left), and the number of epochs of training required (right). Modularity emerges for degrees of connectivity below one half.

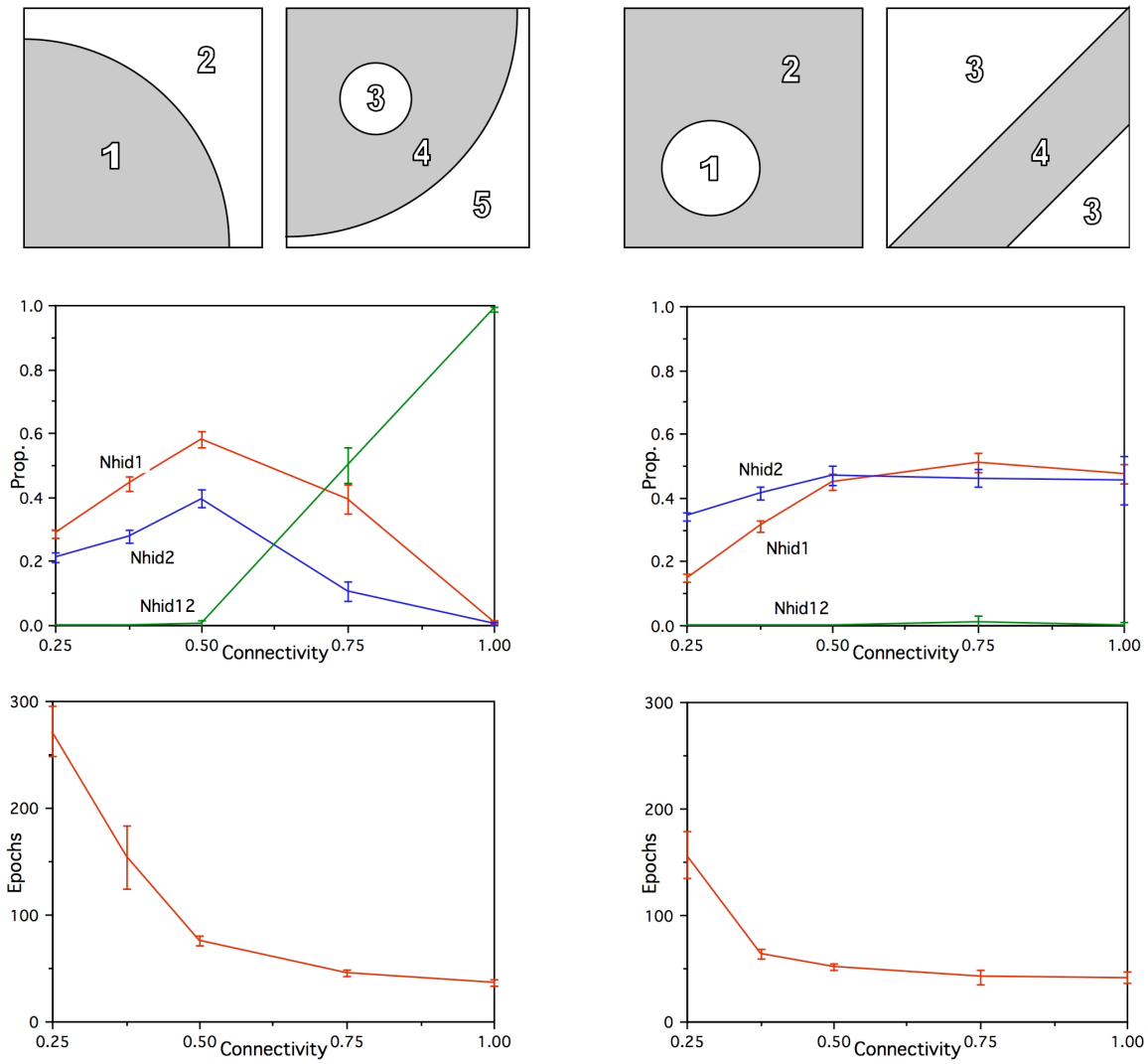

Figure 9: Evolved neural network results for two online generalization problems. The two pairs of classification boundaries (top), the architecture parameters as functions of connectivity proportion (middle), and the number of epochs of training required (bottom). For full connectivity, the evolved architecture is task dependent, but modularity emerges consistently when the degree of connectivity is below one half.