The Importance of Neurophysiological Constraints for Modelling the Emergence of Modularity

John A. Bullinaria

School of Computer Science, University of Birmingham

Birmingham, B15 2TT, UK

Abstract: This paper is concerned with the large scale structure of the human brain, how that relates to human behaviour, and how certain types of modularity could have emerged through evolution as a result of a combination of fitness advantages and neurophysiological constraints. It is argued that computational modelling involving evolutionary neural network simulations is the best way to explore such issues, and a series of simulation results are presented and discussed. The incorporation of known neurophysiological constraints into the models is seen to be of crucial importance.

Introduction

Cognitive neuropsychology and fMRI studies have provided a great deal of information about large scale brain structure. Numerous neuropsychological studies of brain damaged patients have examined patterns of deficits and observed double dissociations (Teuber, 1955) that have been taken to provide evidence of modularity (Shallice, 1988). More recently, the modular structures inferred in that way have been further refined by fMRI studies (e.g., Huettel, Song, & McCarthy, 2004; Binder, et al., 2005), and many large scale "modules" are now thought to consist of collections of lower level modules, some of which are actually used more widely (e.g., Duncan & Owen, 2000; Duncan, 2001; Bookheimer, 2002; Marcus 2004). Despite this large body of research, there remain many controversies concerning modules and modularity.

First, there is disagreement over how to define modules. The hard-wired, innate and informationally encapsulated modules of Fodor (1983), are, for example, rather different from the modules in the cognitive models of Coltheart et al. (1993). What Bullinaria (2005a) would call a module, Plaut (1995) would claim is not a module. The range of different definitions has been discussed by Seok (2006), and it is becoming clear that it is quite appropriate to use different definitions across different applications and different levels of description (e.g., Geary & Huffman, 2002, Seok, 2006).

Next, there is disagreement whether particular data implies modularity or not. For example, doubt has been cast on the inference of modularity from double dissociation (e.g., Dunn & Kirsner, 1988; Van Orden, Pennington & Stone, 2001; Bullinaria, 2005a; Plunkett & Bandelow, 2006), while other earlier claims have been shown to be mere artifacts of the models involved (Bullinaria & Chater, 1995).

Finally, even when it is agreed that certain types of modularity exist, there remains disagreement over the extent to which it is innate rather than learned during the individual's lifetime, and what cost-benefit trade-offs affect that distinction (e.g., O'Leary, 1989; Elman et al., 1996; Jacobs, 1999; Geary & Huffman, 2002). It is known that the costs of learning can drive the evolution of behaviours which reduce that cost, and eventually result in the learned behaviours being assimilated into the genotype - a process often referred to as the Baldwin Effect (Baldwin, 1896; Bullinaria, 2003). It may well be that modular processing is first learned and only later becomes assimilated, or partially assimilated, into the genotype by such a process. On the other hand, if the tasks to be performed change too quickly for evolution to track them, or if the necessary neural structures are too complex for easy genetic encoding, then the need to learn them will persist.

One promising approach for making progress on the issue of brain modularity, while avoiding much of the existing controversy, is to consider from a theoretical point of view what are the possible advantages and disadvantages of modularity. This may then inform why and how and when it may have emerged in brains as a result of evolution. A reliable approach for testing such ideas is to build computational (neural network) models that perform simplified behavioural tasks, and study the various trade-offs inherent in them. Such

models can be built "by hand", but simulating the evolution of such systems is often a more productive approach (Bullinaria, 2005b). Natural selection will presumably have resulted in the most advantageous structures for biological systems, and what emerges from the evolutionary models can be compared with what is believed to be known about real brains. Of course, the evolution of brains is constrained by various biological practicalities (Allman, 1998; Striedter, 2005), and realistic models need to reflect this. Interestingly, evolutionary simulation results show that imposing known physical/neurophysiological constraints on the models can have a drastic effect on what emerges from them. Moreover, such models can also begin to explore the interaction of learning and evolution, and thus explicitly address the nature/nurture debate with regard to brain structures.

The remainder of this paper is structured as follows: In the next section, the main advantages of neural modularity are reviewed. Then an approach is described for simulating the evolution of neural networks that must carry out pairs of simple tasks, and it is shown how this can be applied to studying the evolution of modularity in such systems, with particular reference to the accommodation of known neurophysiological constraints on the emergent structures. Empirical results are presented from a series of computational experiments that are designed to explore the circumstances under which modularity will emerge. The paper ends with some discussion and conclusions.

Advantages of Modularity

The literature already suggests numerous (overlapping) reasons while modularity may be advantageous for information processing systems such as brains. In fact, it can be regarded as a general design principle for both vertebrates and invertebrates (e.g., Leise, 1990). Perhaps the most obvious advantage of modularity corresponds to the familiar idea of breaking a problem into identifiable sub-tasks and making effective use of common processing sub-systems across multiple problems (e.g., Marcus, 2004; Reisinger, Stanley & Miikkulainen, 2004; Kashtan & Alon, 2005). This is certainly standard practice when writing complex computer programs, and Chang (2002) has presented a specific cognitive model in which modularity is shown to allow improved generalization in complex multi-

component language processing tasks. In fact, numerous different types of modular neural networks and their associated successful applications were reviewed some time ago by Caelli, Guan & Wen (1999). Such forms of modularity might have evolved in brains specifically to deal with particular existing tasks. It is also possible that new ways were discovered to use existing modules that gave individuals abilities they did not have before, and that these provided such an evolutionary advantage that they were quickly adopted into the whole population. In fact, it has been argued that modularity will prove particularly advantageous when adaptation to changing environments is required (e.g., Lipson, Pollack & Suh, 2002; Kashtan & Alon, 2005), especially since the opportunity to employ useful modules from a rich existing source will invariably be more efficient than generating a new ability from scratch. That modularity can act to improve robustness and evolvability is certainly not a new idea (e.g., Wagner, 1996).

Another way of thinking about brain modularity is in terms of low level neural processes. It seems intuitively obvious that attempting to carry out two distinct tasks using the same set of neurons and connections will be less efficient than having separate modules for the two tasks. This intuition was explored in a series of simple neural network simulations by Rueckl, Cave, & Kosslyn (1989), in which standard Multi-Layer Perceptron neural networks were trained using gradient descent to perform simplified visual object recognition. This involved carrying out separate "what" and "where" classification tasks on images presented as a 5×5 input grid, as shown in Figure 1. They found that disruptive interference between connection weight updates for the two tasks did result in poorer learning performance for a single network compared to a network with separate modules (blocks of hidden units) for the two tasks. The problem with such a simulation approach, however, is that the learning algorithms used were far from biologically plausible, and it is possible that brains have evolved better learning algorithms that do not suffer such disruptive interference. In fact, it was later shown that a slightly different learning algorithm (equally biologically implausible) did perform significantly better, did not suffer such disruptive interference, and performed better when the neural architecture was non-modular (Bullinaria, 2001). This naturally cast doubt on the whole idea that low level disruptive interference was relevant for modularity.

In an attempt to clarify this matter, an extensive follow-up study was carried out to explore further the trade-off between minimizing interference and restricting the allowed neural architecture (Bullinaria, 2007b). First, by using simulated evolution to optimize the neural architecture for various fixed gradient descent learning algorithms, for the same simplified what-where task studied by Rueckl et al. (1989), it was confirmed that learning algorithms prone to significant learning interference between independent tasks tended to lead to modular architectures, whereas learning algorithms that did not suffer such interference led to non-modular architectures emerging. Then, if the learning algorithm was allowed to evolve alongside the architecture, the learning algorithm least prone to cross-task interference consistently emerged, along with non-modular architectures, and it was shown that this set-up did indeed provide the best performance on the chosen tasks.

The problem remaining was that the simplified what-where learning task, used in virtually all the earlier studies, was far removed from the kind of tasks that are faced by real biological individuals. Rather than learning a small set of input-output mappings by repeated exposure to them, they typically experience steady streams of input patterns drawn from continuous distributions, and they need to learn classification boundaries that allow them to generalize in order to respond correctly to inputs they have never seen before. To explore more realistic learning scenarios of this type, a series of artificial data sets were generated based on various decision boundaries in a two dimensional input space normalized to a unit square. Some examples are shown in Figure 2. Such inputs might correspond to observable features of other animals (shapes, sizes, etc.), and the output classes could represent important properties of them (dangerous, edible, etc.). The neural networks experience a stream of random data points from the input space, and their fitness corresponds to how well they perform the right classifications *before* learning from that experience. Evolving the neural network learning algorithms to perform well on such generalization tasks invariably led to the better (less interference prone) learning algorithm emerging, but the associated architectures were now more variable (Bullinaria, 2007b). There was a trade-off between leaving the learning algorithm more freedom to use all the available network connections, versus minimizing the task interference by placing restrictions on how the connections were

used. Whether that trade-off favoured modularity or non-modularity proved to be rather task dependent, affected by numerous factors, such as the number of classes that need to be learned, the complexity of the various classification boundaries, and the relative difficulties of the two tasks. For example, Task Pair A in Figure 2 leads to non-modular architectures, while Task Pair B usually results in modular architectures.

It seemed then that the emergence of modularity depended both on the power of the learning algorithm and on the properties of the tasks, but the simulations did provide an explanation of why low level neural modules should evolve. The third factor explored by Bullinaria (2007b), after the learning algorithm and task type, related to the fact that there are numerous neurophysiological constraints placed upon biological brains that were absent in the previous models. The remainder of this paper will describe the approach that was used to investigate these factors, and expand upon the analysis and discussion of what emerged from those simulations.

Simulating the Evolution of Modularity

The idea of applying simulated evolution by natural selection to neural networks is now well established (e.g., Yao, 1999; Cantû-Paz & Kamath, 2005). In principle, all aspects of the neural networks can be evolved (the architecture, the properties of the neurons, the connection weights, the learning algorithm, the data preprocessing, and so on), but for modelling brains it is most natural to have an innate learning algorithm to adjust the connection weights during the individual's lifetime, rather than having them evolved (Elman, et al., 1996). The evolution will generate appropriate neural architectures, learning algorithms, and initial weight distributions, that allow the learning to take place most effectively. There are still many different ways this can be done, as discussed by Bullinaria (2007a), all involving a population of neural networks, each specified by an innate genotype that represents the various evolvable parameters. However, the key results concerning modularity appear to be quite robust with respect to the details of the evolutionary process, as demonstrated by the identical outcomes from the generational approach of Bullinaria (2007b) in which the populations are simulated one generation at a time, and the more biologically

inspired approach of Bullinaria (2001) involving populations of individuals of different ages with competition, procreation and deaths each simulated year. Since it seems not to be crucial which evolutionary process is used, this paper will adopt the computationally simpler approach of Bullinaria (2007b).

The neural networks used are standard Multi-Layer Perceptrons (MLPs) with a single hidden layer, sigmoidal processing units, and trained using a gradient descent learning algorithm with a cost function E that is an evolvable linear combination of Sum-Squared Error (SSE) and Cross Entropy (CE). This allows the evolutionary process to choose between SSE as used in the study of Rueckl et al. (1989), and CE which is now known to be more appropriate for classification tasks (Bishop, 1995). The weight update equation for the connection weight w_{ij} between hidden unit i with activation h_i and output unit j with activation o_j and target t_j then takes the form

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

in which $\mu \in [0, 1]$ specifies the learning cost function, with $\mu = 0$ being pure SSE, and $\mu = 1$ being pure CE. The intermediate value of $\mu \sim 0.1$ is equivalent to using SSE with a Sigmoid Prime Offset, which was suggested by Fahlman (1988) as a way of resolving some of the learning difficulties inherent in using pure SSE. Note that it has already been established that having independent evolvable learning rates η_L for the distinct network components L (input to hidden weights *IH*, hidden unit biases *HB*, hidden to output weights *HO*, and output biases *OB*) can lead to massive improvements in performance over a single learning rate across the whole network (Bullinaria, 2005b). Similarly, it proves advantageous to evolve different uniform ranges $[-l_L, +u_L]$ from which to draw the random initial weights for each component L. In total, then, the network initialization and learning process is specified by 13 evolvable parameters (four η_L, four l_L, four u_L and μ), and the evolutionary process is expected to optimize these to give the best possible learning performance on the given tasks.

In this context, modularity can be defined in terms of the pattern of connectivity in the network, ranging from fully modular with separate sets of hidden units for each task, to fully distributed with all hidden units used for all tasks. For two tasks, this can conveniently be

parameterized as shown in Figure 3, with the total number of hidden units *Nhid* partitioned into *Nhid1* that only participate in task 1, *Nhid2* that only participate in task 2, and *Nhid12* that participate in both. If *Nhid* is fixed, that leaves two parameters which need to be evolved to specify the neural architecture, in addition to the 13 learning parameters.

Having specified each neural network by its 15 innate parameters, a fitness measure for that network must be determined, to which the simulated natural selection process can be applied. For current purposes, fitness can be conveniently measured in terms of the number of training epochs required to reach the first full epoch of perfect performance (Bullinaria, 2007a,b). For the what-where task (of Figure 1), an epoch corresponds to the full set of 81 training items, and performance is measured on that training set. For the online generalization tasks (of Figure 2), an epoch is defined as a block of 400 training items and performance is measured on those items *before* training on them. In both cases, the random sequence of training patterns and online learning will introduce a stochastic element into the fitness measurement. This will combine with the various other sources of randomness (such as the choice of initial populations, the selection of mates for reproduction, the crossover and mutation processes, and the initial weights of each new individual) to give a certain variability across runs, as will be seen in the simulation results plotted later.

Each simulated evolution run starts with a population of 100 networks with randomly allocated innate parameters. The learning and initial weight parameters are taken from ranges that span those values generally used in hand-crafted networks, namely [0, 4], and the learning algorithm and architecture parameters are taken from their full valid ranges. Each network starts with random weights drawn from its innately specified ranges, and trains until it reaches its first epoch of perfect performance, which determines its fitness. The next generation is then made up of the fittest half of the population, plus their children. These children inherit innate parameters taken randomly from the ranges spanned by their two parents (representing crossover), with random adjustments applied from a Gaussian distribution (representing mutation). The whole new generation is then initialized and the learning and selection process repeated. Over many generations the innate parameters are optimized and the individual performances improve (Bullinaria, 2007a,b).

As noted above, the neural architectures that emerge from simulated evolution of this type depend on the pair of tasks that need to be performed (Bullinaria, 2007b). The aim of this paper is to consider further how known neurophysiological constraints will affect what evolves. There are two obvious factors that need to be considered: First, the fact that real brain sizes are constrained by numerous factors such as growth costs, energy/oxygen consumption, heat dissipation, and such like. Second, that space and other requirements preclude full connectivity within the brain. The next two sections present results from a series of simulations that explore these issues.

The Effect of Brain Size

In the artificial neural networks studied here, it is the number of hidden units that corresponds to brain, or brain region, size. If all the learning parameters are optimized, the more hidden units employed, the fewer epochs of training are generally required. However, most simulations have the hidden units processed serially on conventional computers, rather than in parallel as in real brains, so it usually works out more computationally efficient to keep their numbers as low as possible. The relevant lower bound on the number of hidden units will depend mainly on the complexity of the task at hand. Brains have rather different processing overheads, with their sizes constrained by various physical factors, such as growth costs, energy/oxygen consumption, heat dissipation, and such like. The various "design problems" that are encountered as brains get bigger or smaller have been discussed by Kaas (2000), and it is not surprising that brain sizes should vary so much across species when the range of demands placed on them by different animals and environments are taken into account. Striedter (2005, ch. 4 & 5) provides a good review of evolutionary changes in the sizes of brains and brain regions.

The problem that faces brain modelers here is the enormous difficulty in reliably matching the scales of the simplified neurons, representations, learning algorithms, tasks, and so on, in the models against the corresponding components found in real brains. What can, *and should*, be done, however, is to run the models many times, with a wide range of numbers of hidden units, to determine how the precise numbers affect the results. In fact,

doing this in previous neuropsychological modelling studies has revealed serious modelling artifacts, with neural networks behaving rather differently when they have barely enough hidden units to carry out the given tasks, compared to when they have brain-like magnitudes of resources (e.g., Bullinaria & Chater, 1995; Bullinaria, 2005a). Moreover, it has also been suggested that restrictions on the number of available neurons can lead to advantages for modularity for representing complex high dimensional spaces (Ballard, 1986), and that is obviously of direct relevance to the current study.

The way to explore this issue is clearly to run the simulations described in the previous section with a wide range of different numbers of hidden units. Figure 4 shows the results obtained for the What-Where problem, with Task 1 being What and Task 2 being Where. In the top-left graph it is seen that the more efficient CE learning algorithm emerges across the full range of values. However, the evolved learning rates do vary considerably with the number of hidden units, as seen in the top-right graph. This confirms the importance of allowing the learning rates for the various network components to vary independently, and of evolving them rather than attempting to set them by hand. The bottom-left graph shows that the evolved architectures remain totally non-modular from near the minimal network size required to perform the given task (9 hidden units) right up to over a hundred times that size (1000 units). Finally, the average evolved network fitness (inverse number of training epochs) is seen in the bottom-right graph to increase approximately logarithmically with the number of hidden units. This is why the total number of hidden units itself has not been allowed to evolve in the simulations, because if it were, it would just keep on increasing to whatever limit was imposed on it. From a practical point of view, this would slow down the simulations unnecessarily (in real time on a non-parallel processor) because of the increased computations required, rather than having the process settle down as the optimal configuration emerges.

For the two generalization tasks of Figure 2, the corresponding dependences of the evolved network architectures on the number of hidden units are shown in Figure 5. The results here are much noisier than for the What-Where task, particularly for the very small networks. As noted before, Task Pair A leads to non-modular architectures evolving, and

there is broad independence of the total number of hidden units. For Task Pair B, modular architectures evolve reliably for large numbers of hidden units (200 or more), but for smaller numbers there is an increasing tendency for varying degrees of non-modularity to occur.

It is possible to check the quality of the evolved architectures by generating contour plots of average fitness as a function of architecture, using the evolved learning parameters (Bullinaria, 2001). Figure 6 shows such plots for Task Pairs A and B for 18 hidden units in total. The apex of each triangle corresponds to a fully distributed architecture, with $Nhid12 \sim Nhid$, while the base represents full modularity, with $Nhid12 \sim 0$. The contours represent epochs of training, running from the lowest value up to 1.3 times that lowest value, with darker shading indicating higher fitness. Even though the evolutionary runs result in variable results for such low numbers of hidden units, these plots show clear non-modularity for Task Pair A and clear modularity for Task Pair B, as is found in the evolutionary runs for higher numbers of hidden units. As has already been observed in other studies (e.g., Bullinaria & Chater, 1995; Bullinaria, 2005a), to achieve reliable results it is best to avoid using networks that have close to the minimal number of hidden units required to perform the given tasks.

The fact that in each case, for reasonably large networks, the emergent architectures are independent of network size means that it is possible to set the number of hidden units at some convenient value and not be too concerned that there remains considerable uncertainty in how exactly a network size maps onto a real brain region size.

The Effect of Neural Connectivity Levels

An important factor related to the size of the brain, or brain region, is that of the neural connectivity levels within in it. As the number of neurons grows, so does the number of potential connections between them, but those connections would take up valuable space in the brain, corresponding to the volume that would need to be occupied by workable axons and dendrites. Moreover, the longer range connections required to fully connect large brains would also reduce the speed of neural computation. Beyond a certain size, full connectivity becomes an inefficient use of resources, as Ringo (1991) has demonstrated with an explicit model. Such biological factors and their consequences for brain structure have also been

discussed in some detail by Stevens (1989), Kaas (2000), Changizi (2001) and Karbowski (2003), with the general conclusion that the degree of connectedness should fall as the brain size increases. The ways by which evolution might have optimized neural circuits in the light of these issues (i.e. to minimize conduction delays, signal attenuation, connection volumes, and so on) has been considered by Chklovskii et al. (2002), and they concluded that approximately 60% of the space should be taken up by the wiring (i.e. axons and dendrites), which is close to the proportion actually found. It is clear that the issue of neural connectivity imposes important constraints on real brain structures, and should therefore be taken into account in any reliable models of their emergence.

The most obvious way to minimize the volume of connections and increase the rate of information flow would be to keep the connections as short as possible. Jacobs & Jordan (1992) considered how such a bias towards short connections in neural network models would affect the neural architecture, and found that it led to a tendency to decompose tasks into sub-tasks, which is clearly of direct relevance to the issue of modularity. Similar architecture issues were studied in the computational embryogeny approach of Bowers & Bullinaria (2005) in which processes were evolved that allowed neurons to grow from stem cells to connect up input and output neurons and learn to perform simple mappings between them. The problem with such models is that there is always a danger of identifying factors as physical constraints when they are actually unconstrained physical solutions to other problems, and this could introduce a bias into what the models subsequently tell us. For example, it could be that short connections emerged as the best way to implement modularity that has evolved because of its computational advantages, rather than as a consequence of the physical space taken up by the connections, in which case short connections should not be built into models as a constraint on what can evolve.

The safest way to proceed is to assume nothing about the physical brain structures other than the fact that full neural connectivity is not possible. The idea is to see what architectural changes will emerge simply by restricting the *proportion* of connections, without regard to any physical properties like the neuron locations and connection lengths. The natural assumption is that, for any constrained level of connectivity, evolution will result in the

various physical costs being minimized by a suitable neuronal layout. Various aspects of this assumption have already been studied in some detail: Mitchison (1991) considered how different branching patterns affect the minimization of cortical wiring. Cherniak (1995) looked at neural component placement with a similar aim. Zhang & Sejnowski (2000) developed models that explained the empirical universal scaling law between gray and white matter. Chklovskii (2004) argued that under certain biologically plausible constraints the optimal neuronal layout problem has an exact solution that is in broad agreement with the layouts found in real brains. Striedter (2005, ch. 7) provides a good overview of the current beliefs concerning the evolution of neuronal connectivity.

It is reasonably straightforward to extend the evolutionary neural network approach described above to test the effect of restricted neural connectivity levels. The only extra feature needed is a restriction on the degree of connectivity between layers to some fraction f of full connectivity, where full connectivity means that each node of each layer is connected to each node of the next layer. One way this reduced connectivity can be achieved is by randomly removing a fraction $1–f$ of connections from the fully connected network. Within the block structure of Figure 3, full connectivity means $Nhid12 = Nhid$ and $Nhid1 = Nhid2 = 0$, where $Nhid$ is the total number of hidden units. Here, the total connectivity levels can also be reduced more systematically by increasing the module sizes $Nhid1$ and/or $Nhid2$ at the expense of $Nhid12$, or by reducing the total number of connected neurons $Nhid1 + Nhid2 + Nhid12$ below the maximum allowed number $Nhid$. The evolutionary process will determine the best way to reduce the connectivity level, by being free to adjust the innate degree of connectivity f_{HO} between the blocks of hidden and output units (achieved by randomly removing a fraction $1–f_{HO}$ of allowed connections) as well as the existing architecture parameters $Nhid1$, $Nhid2$ and $Nhid12$. The total proportion of connectivity between the hidden and output layers is then easily calculated to be

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

The idea is to run a number of simulations to explore how the outcomes vary for different

values of f. In each simulation, f is set to be a particular fixed value, and the architecture parameters are allowed to evolve as before, but now the equation needs to be inverted to compute the value of f_{HO} that gives the right total connectivity level f.

The results of doing this for the What-Where task with 72 hidden units are shown in Figure 7. The learning algorithm parameter μ and the learning rates η_L remain fairly steady as the level of connectivity is reduced, at least until around $f \sim 0.25$ where there remain so few connections left that the network is unable to perform the task reliably. The fitness, measured as the inverse of the number of epochs of training required, falls steadily as the connectivity is reduced. The interesting discovery is that the number of hidden units shared by both output tasks, *Nhid12*, falls almost linearly with f until it reaches one half, when it stays close to zero for all lower levels of connectivity. Modules emerge for each of the two output tasks with their relative sizes in proportion to the difficulty of the tasks, with a slight floor effect as the connectivity level gets close to the minimum required for the tasks. That smaller modules are needed for the linearly separable Where task, compared to the non-linearly separable What task, was found in the earlier studies too (Rueckl et al., 1989; Bullinaria, 2001). The minimum workable connectivity level can be decreased by increasing the total number of hidden units, but modularity still emerges as the preferred architecture for all connectivity fractions below one half. When connections need to be removed to achieve a lower value of f, the choice is effectively between randomly removing them from anywhere, versus systematically removing them from hidden units that contribute to both tasks to increase the modularity, and the simulated evolution shows that increasing the modularity is the better option. Connectivity levels in human brains are certainly considerably below one half, and even in much simpler organisms they are likely to be below that level (Chklovskii et al., 2002). It seems then, that a reason has been found why modularity should emerge in simple neural structures performing multiple tasks.

Naturally, it would not be wise to base such a conclusion on one particular problem. The robustness of the result needs to be tested on other pairs of tasks. Figure 8 shows the corresponding results for the Generalization Task Pair A shown in Figure 2, for 200 hidden units. For full connectivity, a purely non-modular architecture evolves, as for the What-

Where tasks, and the result of restricting the connectivity level follows a similar pattern. The learning parameters are less tightly constrained in this case, indicating that they have less influence on the performance. The magnitudes of the evolved learning rates are also rather different for this task, as one might expect given the different numbers of input and output units and the training data distributions, but they are again fairly steady across the range of connectivity levels. Most importantly, the degree of modularity again falls almost linearly from full to half connectivity, and the architecture remains purely modular for all lower connectivity levels.

It can be seen more clearly how the performance on this task depends on the network architecture in the fitness versus architecture contour plots of Figure 9, generated in the same way as those of Figure 6. Even though the fitness levels are averaged over 500 individuals each, the plots are still rather noisy, but the patterns are easy to see, and the evolutionary pressure is clearly enough for the preferred architectures to emerge. The top-left graph is for the evolved networks with full connectivity, with the best performance clearly around the fully non-modular region. The top-right graph is for networks with a connectivity level of $f = 0.75$, and the peak performance area has now shifted to the intermediate regions near the centre of the triangle. The bottom-left graph is for networks with a connectivity level of $f = 0.5$, and the peak performance area is now clearly in the modular region. As a consistency check, the bottom-right graph shows the performance of the evolved $f = 0.5$ networks when f_{HO} is left unconstrained at 1 for all architectures. In this case, even though all the other network parameters have been optimized to deal with restricted connectivity, as soon as the restriction to keep f at 0.5 for all values of *Nhid12* is removed, the advantage of non-modularity returns. It seems that the increase of modularity as the connectivity level reduces is a robust effect.

Finally, for completeness, it is worth checking how reduced connectivity levels affect networks evolved to deal with task pairs for which modularity consistently emerges even when full connectivity is possible. Figure 10 shows such results for the Generalization Task Pair B shown in Figure 2, for 200 hidden units. Again the learning parameters are fairly stable across connectivity levels and the fitness increases with connectivity. In this case the

architecture simply remains modular throughout.

Interestingly, below connectivity levels of 0.5, the networks evolved to perform the generalization tasks employ a different connection reduction strategy to those for the What-Where learning task. Rather than removing random connections by reducing f_{HO}, they instead use the architecture parameters to reduce their usage of the available pool of hidden units by having some of them not connected to any output units. This is seen more clearly in Figure 11. Given the small number of output units driven by each module hidden unit in these tasks, this strategy makes good sense. That the simulated evolution automatically determines such sensible strategies, that may easily have been missed by human modelers, is another reason why the evolutionary approach shows such good promise in this area.

Discussion and Conclusions

This paper has shown how simulations of the evolution of simple neural networks, that must learn to perform pairs of simple tasks, can be used to explore the factors that might lead to the emergence of modularity in brains. It noted that particular attention needs to be paid to two important high level physical constraints that affect real brains, namely the restrictions placed on brain size and the associated levels of neural connectivity However, to understand the reasons for particular brain structures, it is always 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 advantages were sufficient to cause them to evolve. For this reason, models across the spectrum were studied, ranging from the kind of unconstrained systems often used when building the best possible artificial systems, to those with levels of constraint approaching those found in biological systems.

It is clear from the plots of fitness in Figures 4, 7, 8 and 10 that there is a fitness advantage for larger networks and higher levels of connectivity. However, as discussed above, there are strong biological reasons which make that infeasible. Once those biological factors are incorporated into the models, modular architectures emerge automatically as the best approach for dealing with those constraints. This confirms empirically earlier

suggestions that this would happen (e.g., Kaas, 2000).

The tasks and neural networks that have been discussed in this paper are, of course, far removed from the levels of complexity seen in the higher organisms alive today, but they may be more directly relevant to the low level processing requirements of simpler organisms at earlier stages of evolutionary history. There remains a good deal of controversy in the field of early nervous system evolution (e.g., Holland, 2003), but having demonstrated how simple low level modules can emerge, it is reasonable to conjecture that evolution will find ways of using them as building blocks for creating higher level modules, and eventually whole brains as they exist today (e.g., Alon, 2003). Certainly, the kinds of "distinct output tasks" used in the simulations presented in this paper will almost certainly correspond to separate components or sub-tasks of more complex higher level systems. Exploring how the prior existence of simple low level modules of the type considered here will facilitate (or not facilitate) the emergence of more complex modules (with their higher level advantages) would clearly be a fruitful area of future research.

Having established that modularity emerges as the result of restricted connectivity levels, the next question to ask is how are those restricted connectivity levels actually implemented in biological brains. Chklovskii (2004) has considered the issue of optimal neuronal layout in some detail. Even highly modular architectures will still need long range connections between modules. It seems most probable that something like "small world" or scale free networks (Watts & Strogatz, 1998) will prove to be the most efficient organizational structure, as has been discussed by Sporns et al. (2000), Buzsáki, et al. (2004), and Sporns, et al. (2004). Moreover, it has been suggested that the trade-off between computational requirements and physical constraints may be responsible for the diversity of interneurons in the mammalian cortex (Chklovskii, et al., 2002; Buzsáki, et al., 2004). Such neural structures, of course, go way beyond the simple architecture of Figure 3, and more complex neural representation schemes will be required to extend the evolutionary neural network approach in that direction.

Perhaps the best way to make further progress in this area would be to allow increasingly general neural structures to evolve. Bowers & Bullinaria (2005) started looking at this by

evolving systems that grew neural structures from stem cells, with connections growing along chemical gradients. Approaches such as this will obviously require significantly more than the two evolvable parameters needed to specify the architecture in Figure 3, and that will make the evolutionary process considerably more demanding, but it will allow an almost unlimited range of neural structures to emerge. It will also allow appropriate structures to emerge in response to the training data, rather than being fixed innately, and this could begin to address questions about what degree of modularity should be innate and how much should be learned within a lifetime. Taking the evolving neural network approach in this direction will be difficult, but the results presented in this paper suggest that it could be an extremely profitable way to make further progress in understanding brain structures.

References

Allman, J.M. (1998). *Evolving Brains*. New York, NY: Scientific American Library/W.H. Freeman.

Alon, U. (2003). Biological networks: The tinkerer as an engineer. *Science*, **301**, 1866-1867.

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.

Binder, J.R., Medler, D.A., Desai, R., Conant, L.L. & Liebenthal E. (2005). Some neurophysiological constraints on models of word naming. *NeuroImage*, **27**, 677-693.

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. (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 Associates.

Bullinaria, J.A. (2003). 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. (2005a). Connectionist neuropsychology. In G. Houghton (Ed.), *Connectionist Models in Cognitive Psychology*, 83-111. Hove, UK: Psychology Press.

Bullinaria, J.A. (2005b). Evolving neural networks: Is it really worth the effort? In: *Proceedings of the European Symposium on Artificial Neural Networks*, 267-272. Evere, Belgium: d-side.

Bullinaria, J.A. (2007a). Using evolution to improve neural network learning: Pitfalls and solutions. *Neural Computing & Applications*, **16**, 209-226.

Bullinaria, J.A. (2007b). Understanding the emergence of modularity in neural systems. *Cognitive Science*, **31**, 673-695.

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

Buzsáki, G. , Geisler, C., Henze, D.A. & Wang, X.-J. (2004). Interneuron diversity series: Circuit complexity and axon wiring economy of cortical interneurons, *Trends in Neurosciences*, **27**, 186-193.

Caelli, T., Guan, L. & Wen, W. (1999). Modularity in neural computing. *Proceedings of the IEEE,* **87**, 1497-1518.

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.

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.

Cherniak, C. (1995). Neural component placement. *Trends in Neurosciences*, **18**, 522-527.

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.

Coltheart, M., Curtis, B., Atkins, P. & Haller, M. (1993). Models of reading aloud: Dual-route and parallel-distributed-processing approaches. *Psychological Review*, **100**, 589-608.

Duncan, J. (2001). An adaptive coding model of neural function in prefrontal cortex. *Nature Reviews: Neuroscience*, **2**, 820-829.

Duncan, J. & Owen, A.M. (2000). Common regions of the human frontal lobe recruited by diverse cognitive demands. *Trends in Neurosciences*, **23**, 475-483.

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

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.

Holland, N.D. (2003). Early central nervous system evolution: An era of skin brains? *Nature Reviews: Neuroscience*, **4**, 617-627.

Huettel, S.A., Song, A.W. & McCarthy, G. (2004). *Functional Magnetic Resonance Imaging*. Sunderland, MA: Sinauer Associates.

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.

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.

Leise, E. (1990). Modular construction of nervous systems: A basic principle of design for invertebrates and vertebrates. *Brain Research Review*, **15**, 1-23.

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.

Mitchison, G. (1991). Neuronal branching patterns and the economy of cortical wiring. *Philosophical Transactions of the Royal Society of London* B, **245**, 151-158.

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.

Plunkett, K & Bandelow, S. (2006). Stochastic approaches to understanding dissociations in inflectional morphology. *Brain and Language*, **98**, 194-209.

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.

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.

Sporns, O., Tononi, G. & Edelman, G.M. (2000). Theoretical neuroanatomy: Relating anatomical and functional connectivity in graphs and cortical connection matrices. *Cerebral Cortex*, **10**, 127-141.

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

Striedter, G.F. (2005). *Principles of Brain Evolution.* Sunderland, MA: Sinauer Associates.

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

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.

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

Zhang, K. & Sejnowski, T.J. (2000). A universal scaling law between gray matter and white matter of cerebral cortex. *Proceedings of the National Academy of Sciences*, **97**, 5621-5626.

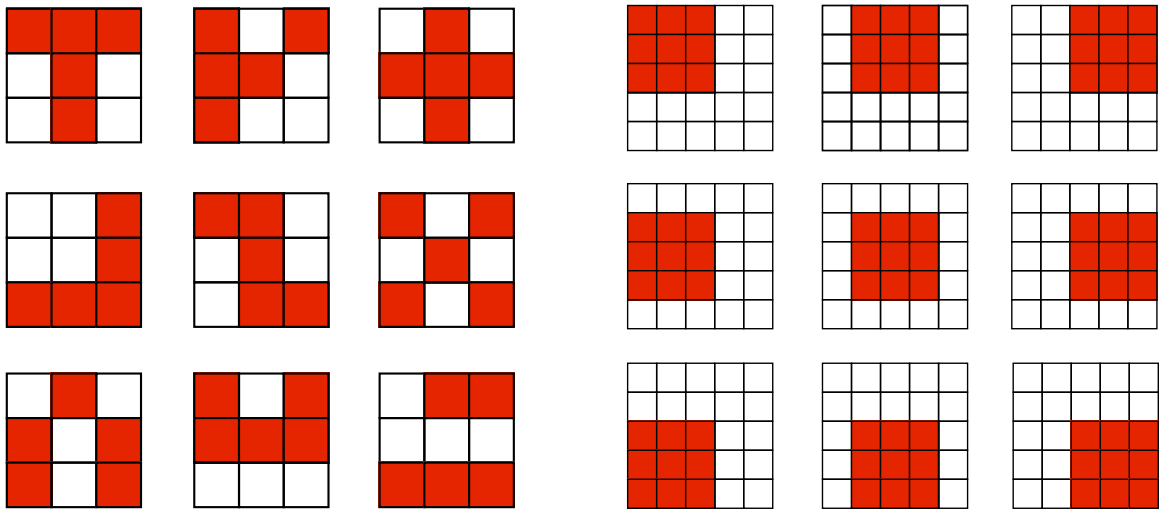

Figure 1: The simplified What-Where training data studied by Ruekl et al. (1989) and most subsequent studies. There are nine distinct 3×3 images (What) that may appear in any of nine positions (Where) in the 5×5 input grid.

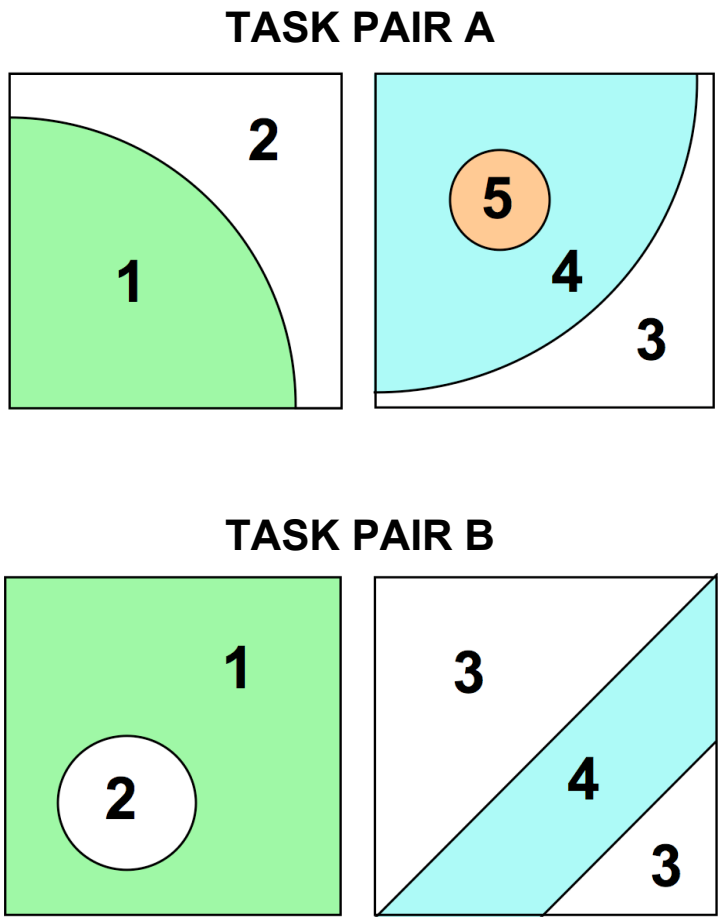

Figure 2: Two representative pairs of classification generalization tasks based on a two dimensional input space. The upper pair (A) has one two-class task and one three-class task, both with circular boundaries. The lower pair (B) has two two-class tasks, one with circular boundary and one with linear boundary.

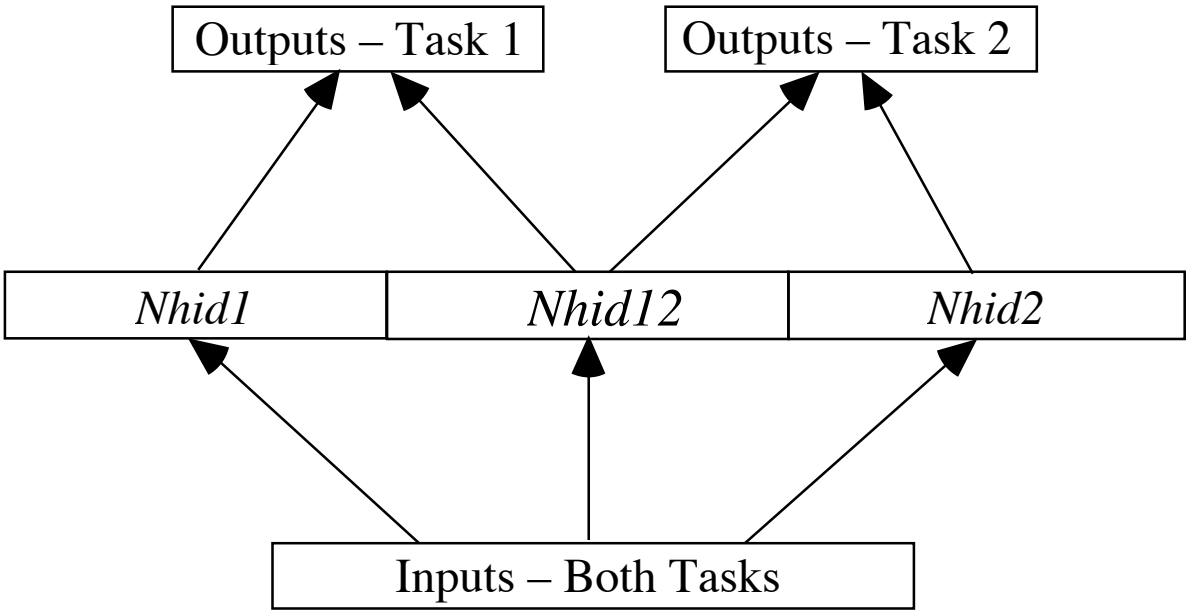

Figure 3: The neural network architecture designed to study the evolution of modularity. Rectangles represent sets of neurons and arrows represent full connectivity. There is a common set of input units, a set of output units for each task, and a set of hidden units partitioned according to which output set the neurons connect to.

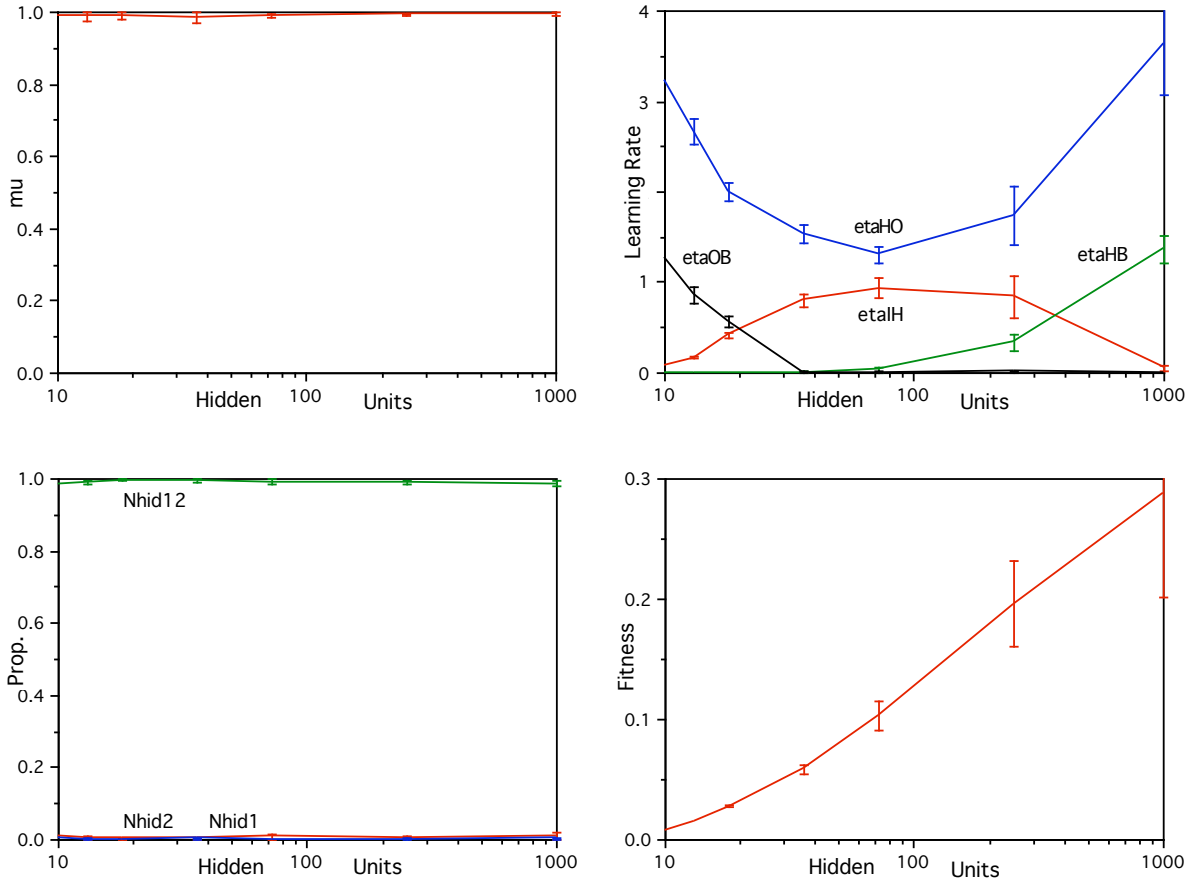

Figure 4: Dependence of the evolved What-Where networks on their number of hidden units. The learning cost function parameter μ (top-left), learning rates η_L (top-right), architecture parameters (bottom-left), and average fitness (bottom-right).

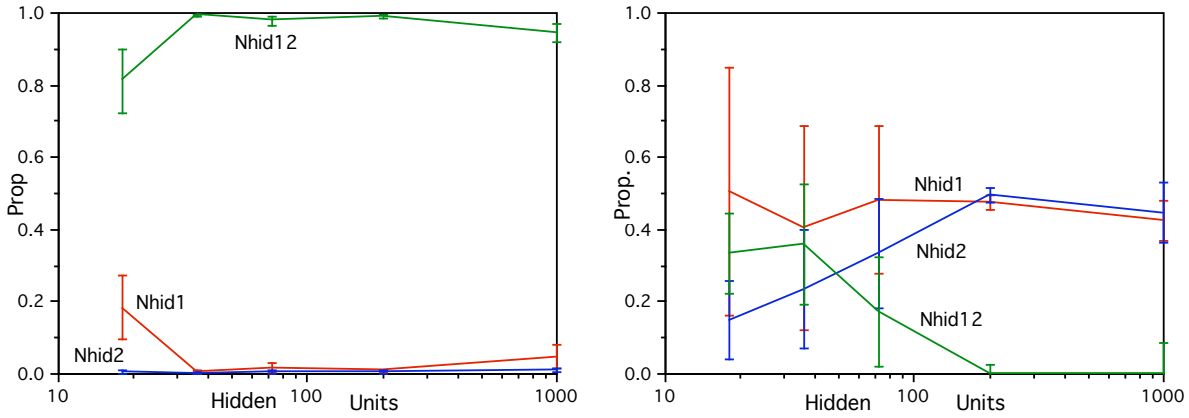

Figure 5: Dependence of the evolved generalization task network architectures on their total number of hidden units: Task Pair A (left) and Task Pair B (right).

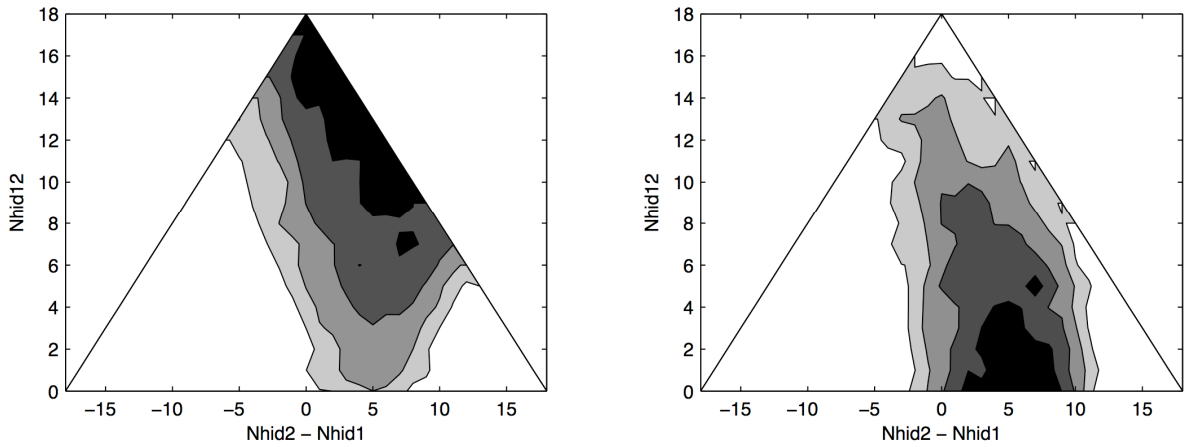

Figure 6: Contour plots of network learning performance as a function of architecture for 18 hidden units in total, with higher fitness shown darker. Task Pair A (left) and Task Pair B (right).

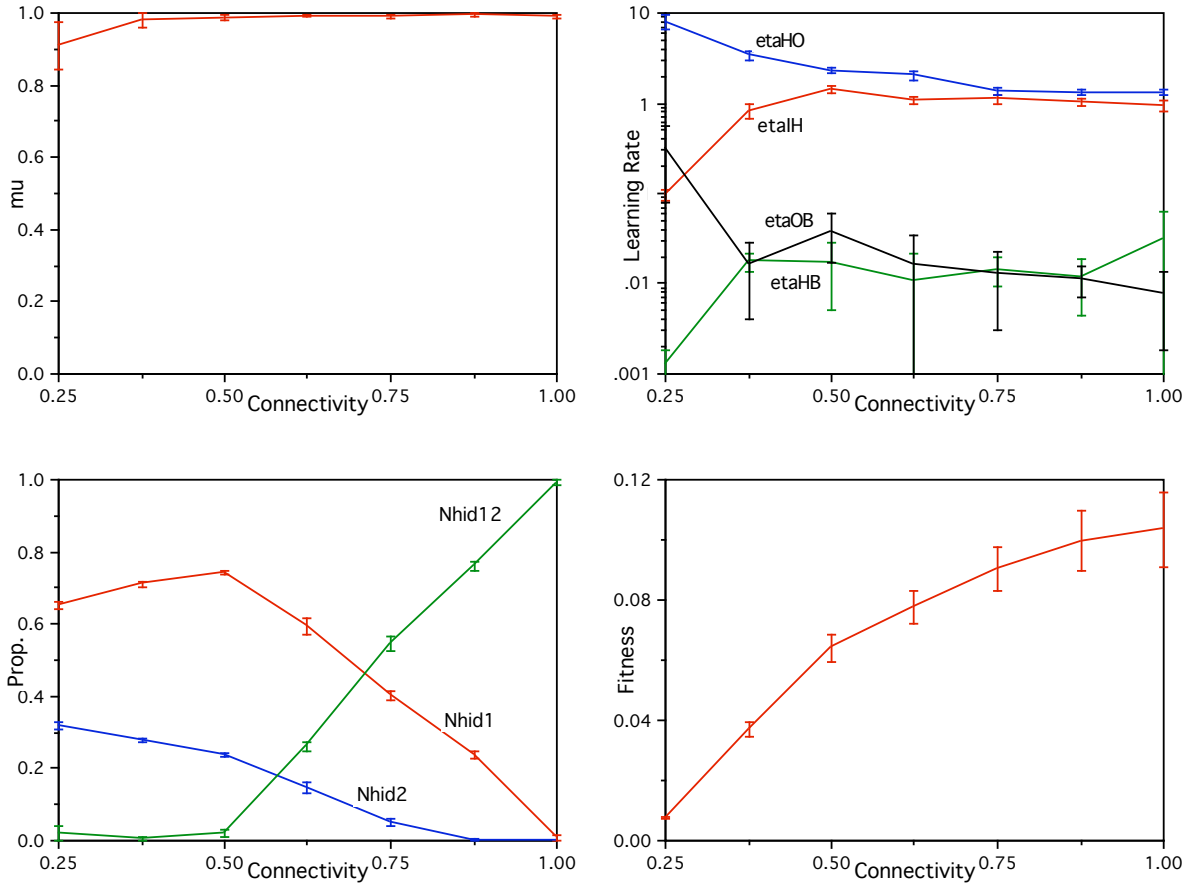

Figure 7: Dependence of the evolved What-Where networks on the fraction f of full neural connectivity. The learning cost function parameter μ (top-left), learning rates η_L (top-right), architecture parameters (bottom-left), and average fitness (bottom-right).

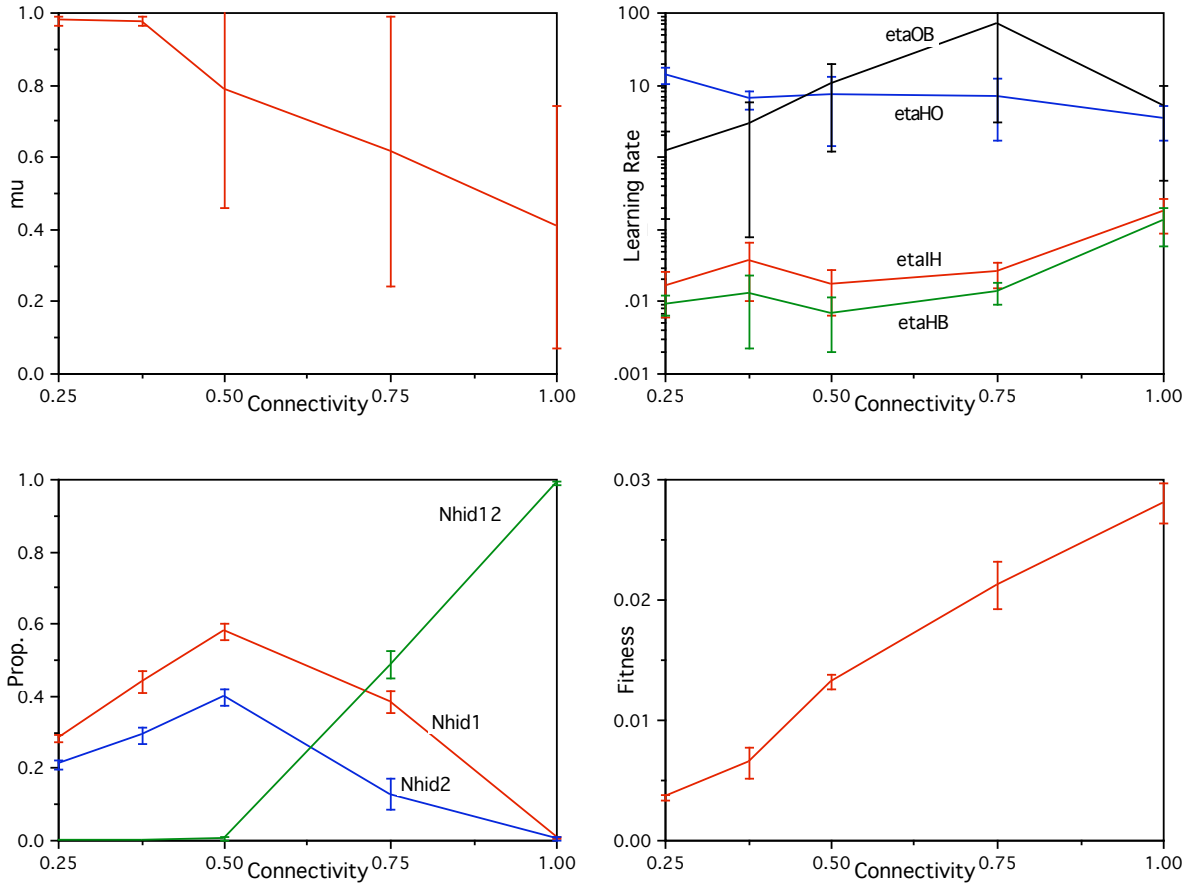

Figure 8: Dependence of the evolved Generalization Task Pair A networks on the fraction f of full neural connectivity. The learning cost function parameter μ (top-left), learning rates η_L (top-right), architecture parameters (bottom-left), and average fitness (bottom-right).

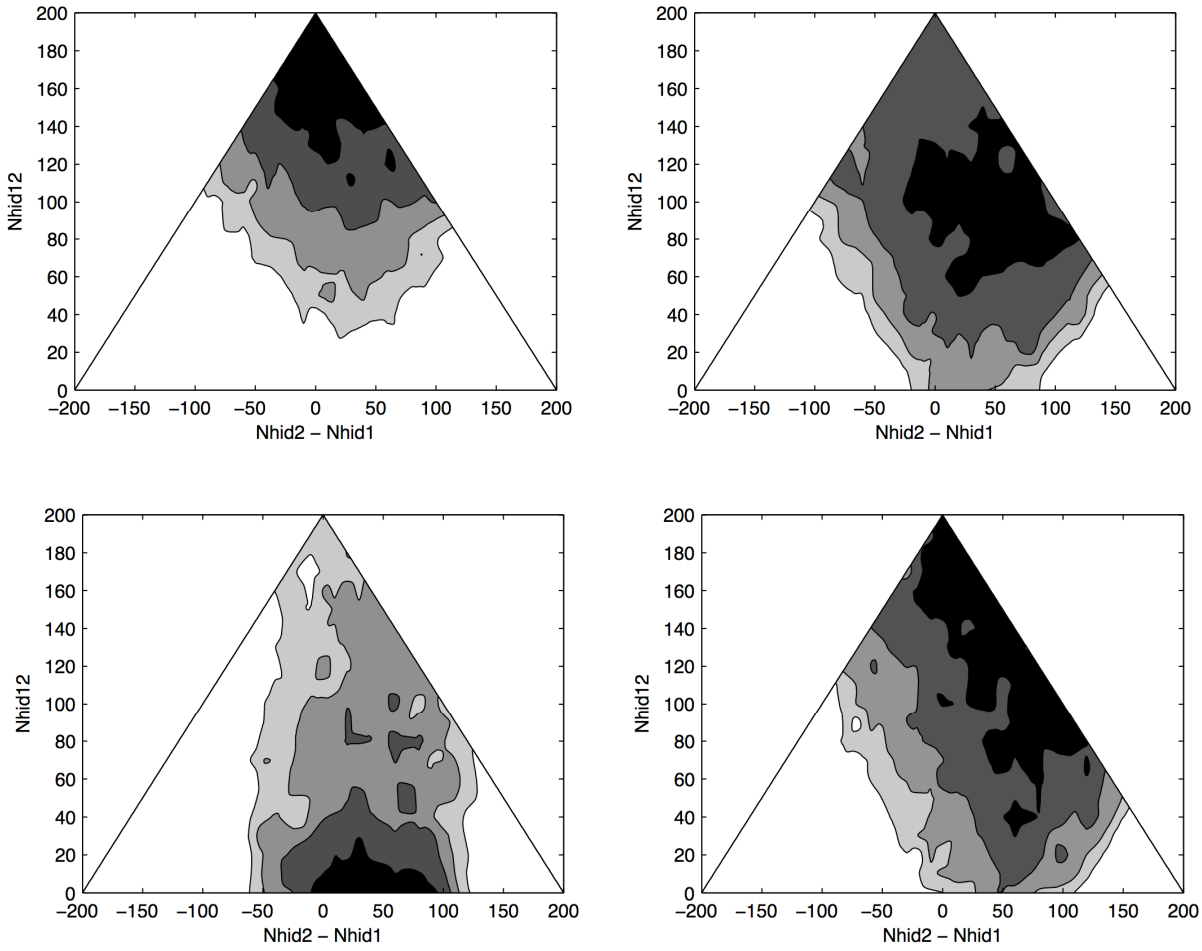

Figure 9: Contour plots of network learning performance as a function of architecture, for Task Pair A and 200 hidden units, with higher fitness shown darker. The evolved full connectivity networks (top-left), 0.75 connectivity networks (top-right), 0.5 connectivity networks (bottom-left), and 0.5 connectivity networks without the connectivity level enforced (bottom-right).

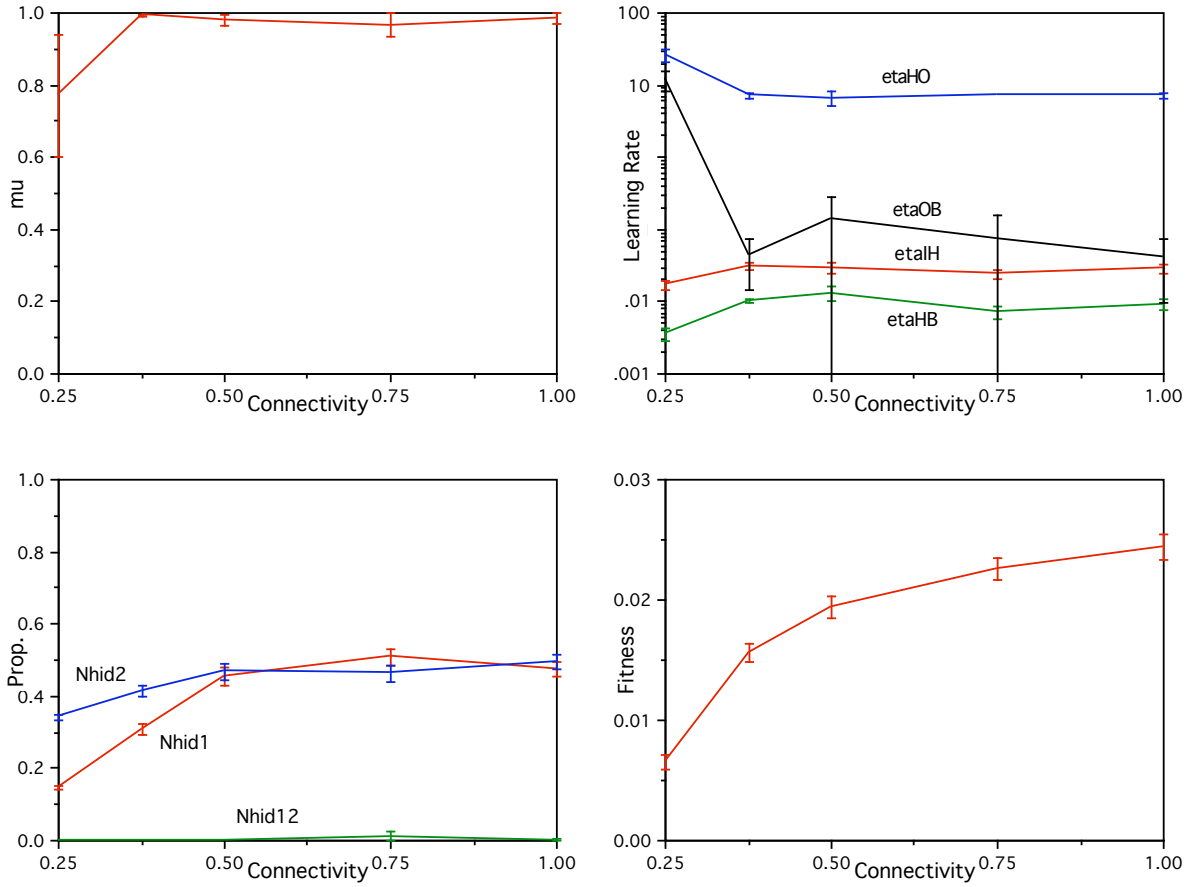

Figure 10: Dependence of the evolved Generalization Task Pair B networks on the fraction f of full neural connectivity. The learning cost function parameter μ (top-left), learning rates η_L (top-right), architecture parameters (bottom-left), and average fitness (bottom-right).

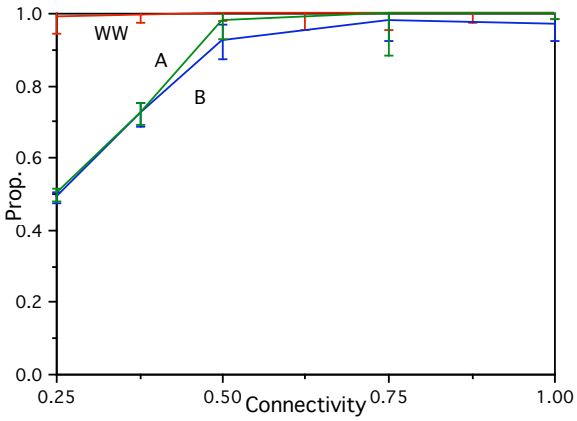

Figure 11: Proportional usage of the available hidden units as the connectivity level is restricted. The evolved What-Where task networks (WW) are seen to behave rather differently to those evolved for the Generalization Task Pairs (A and B).