Understanding the Advantages of Modularity in Neural Systems

John A. Bullinaria (j.a.bullinaria@cs.bham.ac.uk)
School of Computer Science, University of Birmingham,
Birmingham, B15 2TT, UK

Abstract

Understanding, or even defining, modularity in the human brain is not as straightforward as one might expect. It is natural to assume that modularity offers computational advantages, and that evolution by natural selection would translate those advantages into the kind of modular neural structures familiar to cognitive scientists. However, explicit simulations of the evolution of neural systems have shown that, in many cases, it is actually *non*-modular architectures that are most efficient. In this paper, I present a further series of simulations that reveal a crucial dependence on the details of the tasks that are being modelled, and the importance of taking into account physical brain constraints, such as the degree of neural connectivity. Eventually, we end up finding modularity emerging reliably from evolution across a range of neural processing tasks.

Introduction

Cognitive neuropsychology and fRMI research have made great progress in elucidating the structure of the human brain, and, although much controversy remains, it appears to involve some degree of modularity. However, the reasons for that modularity are not very well understood, and it is not clear how much of the modularity is innate and how much arises through learning, nor even how one should best define modularity (e.g., Jacobs, 1999; Elman et al., 1996). A natural assumption is that modular systems have some advantage over non-modular systems, and that either evolution or learning will therefore result in the emergence of modules. The big question for cognitive scientists then is: what are those advantages, and what mechanisms enable those advantages to translate into modular architectures? One promising approach has been to simulate or model the evolution of some appropriately simplified neural systems performing simplified cognitive tasks, and study the structures that emerge. In the next section of this paper I review some of the previous attempts to do this, and identify some of their shortcomings. I will then present a new series of simulations that clarify many of the relevant issues. I end with some discussion and conclusions.

Previous Explorations

The earliest systematic study in this area was the Rueckl, Cave & Kosslyn (1989) computational investigation of the separation of "what" and "where" processing in the human brain. They carried out a series of simulations of simple neural network models trained to perform what and where classifications from simplified 'retinal images', and found that a modular architecture was able to generate more efficient internal representations and learned more easily than fully distributed networks. It is debatable whether this could be the real reason for modularity in the human visual system (Mishkin, Ungerleider & Macko, 1983; Ungerleider & Haxby, 1994), but given the obvious potential for disruptive interference in the simultaneous processing of two independent tasks, it should be no surprise if, in general, dedicated modules for two tasks work better than a single homogeneous system. It is then easy to imagine how evolution by natural selection could enable that advantage to lead to the emergence of modularity.

The obvious next step was to simulate such an evolutionary process and watch the modularity emerge. Although such simulations did show that modularity could evolve in that way if learning and performance were based on the Sum-Squared Error (SSE) measure, they also showed that even better *non*-modular systems could emerge if based on the Cross Entropy (CE) error measure, thus throwing this whole approach into doubt (Bullinaria, 2001).

Other evolutionary neural network simulations involving the same what-where tasks (Di Ferdinando et al., 2001; Calabretta et al., 2003) have confirmed the increasingly wide-spread 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 have also elucidated further the emergence of modularity in the SSE case, but they didn't consider CE based learning.

In another series of evolutionary neural network simulations, Hüsken, Igel & Toussaint (2002) introduced finer grained measures of modularity and also 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 arose until limits were placed on dendritic distances and the output neurons corresponding to the two tasks were physically separated by sufficient distance. This was seen to be consistent with the consequences of a bias towards short connections discussed by Jacobs & Jordan (1992).

In a non-evolutionary study, Jacobs, Jordan & Barto (1991) explored the emergence of modules in a gated mixtures of experts network, but it is difficult to set up those systems in such a way that there is no inherent bias towards modularity. If one removes the bias towards modularity, and evolves the gating parameters, we end up with the same SSE versus CE differences as above (Bullinaria, 2002).

Three factors clearly need further work: the dependence on the learning algorithm (e.g., SSE or CE), the effect of physical constraints, and the dependence on the task (i.e., how general are the what-where results). The following three sections will address each of these issues.

Evolving the Learning Algorithm

The Bullinaria (2001, 2002) neural network simulations showed that modularity was advantageous for the simplified what-where problem if the SSE cost function was used for learning, as in the Rueckl et al. (1989) study, but not if the CE cost function was used. For either cost function there will be 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 question is: under which circumstances will the trade-off favor modularity?

There is a well known problem when using the SSE cost function in neural networks with sigmoidal outputs and binary targets. During learning, the gradient descent weight updates are proportional to the output sigmoid derivatives, which are 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, then it can be difficult, if not impossible, to correct them later. Attempts to evolve solutions to this problem for general single task binary mappings (Bullinaria, 2003) consistently resulted in the SSE learning algorithm *evolving into* the CE learning algorithm. The problematic sigmoid derivatives cancel out of the weight updates for the CE cost function, and there are also good theoretical reasons why the CE cost function is more appropriate for classification tasks anyway (Bishop, 2001). It is not surprising then, that the Bullinaria (2001) study found that the interference prone SSE case favored modularity, while the superior CE algorithm preferred the extra flexibility of non-modularity. The question remains: will non-modularity always be the preferred option? In the remainder of this paper I shall present a further series of simulations that explore this issue.

The general idea of evolving neural networks is now well established (e.g., Yao, 1999). One takes a whole population of individual neural networks, each specified by a series of innate parameters. Then at each generation, the least fit individuals are replaced by children produced from the fittest individuals (using appropriate forms of cross-over and mutation). Such repeated natural selection causes useful innate characteristics to proliferate in the population, and fitness levels improve towards some local optimum.

For our purposes, a standard feed-forward neural network

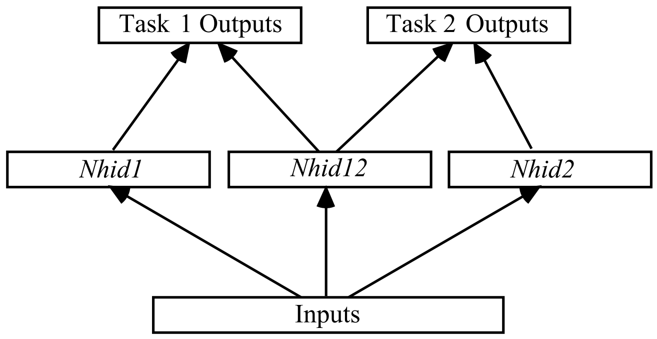

Figure 1: The simplified neural network architecture used to study the evolution of modularity.

as shown in Figure 1 is appropriate, with architecture parameters *Nhid1*, *Nhid12* and *Nhid2* that specify how many hidden units connect to each set of output units. If *Nhid12* tends to zero, we have a modular architecture, 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. The natural innate learning parameters here are the random initial weight distributions $[-l_L, +u_L]$ and 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). Previously, the learning algorithm has been fixed to be standard gradient descent learning using either the SSE or CE cost function (Bullinaria, 2001). Here we shall let the learning algorithm itself evolve too by using a cost function that can be SSE, CE, or anything in between:

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

The parameter μ is bounded to lie in the range [0, 1], so the extreme values of 0 and 1 correspond to the SSE and CE learning algorithm, while a value around 0.1 corresponds to the traditional 'sigmoid prime offset' approach for avoiding the SSE learning problem (Bullinaria, 2003). If we have a fixed total number of hidden units, that gives us two architecture and thirteen learning parameters to evolve.

We shall start with the same what-where training data set as used by Rueckl et al. (1989) and most subsequent studies, with nine 3×3 patterns that may appear in nine positions in a 5×5 input space. Fitness here corresponds to the number of training epochs required to correctly classify all 81 input patterns. The simulation results have been found to be extremely robust with respect to the evolutionary details. All the results presented here are for populations of 100 individuals. At each generation, half the children copy the innate parameters of a parent, and half have each parameter value chosen from the range spanned by their two parents, plus random Gaussian mutations that allow parameters outside that range. The initial population was started with random innate parameters, and the evolutionary process continued until all the parameters had clearly settled down.

Figure 2 shows the evolutionary simulation results for

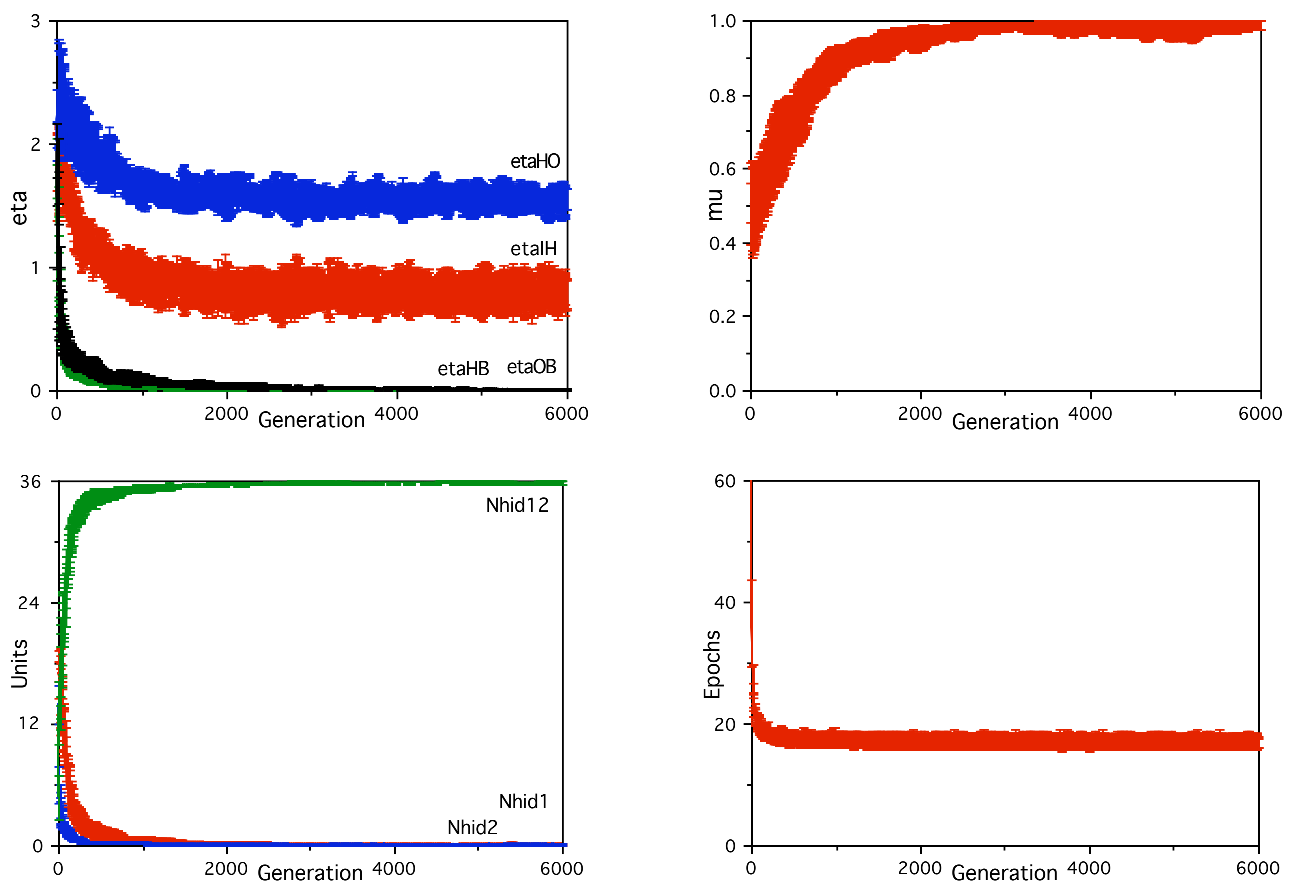

Figure 2: The evolution of the standard what-where neural network with 36 hidden units: the learning rates (top left), the CE versus SSE parameter μ (top right), the architecture parameters (bottom left), and epochs of training required (bottom right).

neural networks with 36 hidden units, with mean values and standard deviations over ten runs. The learning rates and initial weight distributions don't tell us much, apart from the fact that they differ somewhat from the kinds of parameters traditionally used in hand-built networks. The parameter μ ends up very close to 1, corresponding to a purely CE learning algorithm. The evolved architecture parameters correspond to totally non-modular networks. Together the evolved parameters result in the training data being learned in around 18 epochs, and provide a solid confirmation of the earlier results (Bullinaria, 2001) that the requirement for faster learning in this what-where task leads reliably to the emergence of *non-modular* neural architectures.

A further important consideration is the relation between the computational power of the neural network compared with the complexity of the task. It is easy to check this by repeating the above simulations with different total numbers of hidden units. Figure 3 shows how the evolved network architecture and performance varies with the computational power. On the left we see that the evolved architecture remains non-modular from the minimal network required to perform the given task (9 hidden units) to over a hundred times that size (1000 units). On the right we see how the required number of epochs of training varies with the computational power available. It seems that the optimality of non-modular architectures for this what-where task is quite robust with respect to network complexity too.

Physical Constraints on Neural Connectivity

In building cognitive models, one naturally needs to take into account the physical properties of the brain, in addition to the computations they are performing. Of particular relevance to us here is the fact that the significant volume occupied by neural connections (i.e. axons and dendrites) precludes full neural connectivity (Chklovskii et al., 2002; Sporns et al., 2004). Jacobs & Jordan (1992) and Bowers & Bullinaria (2005) have already looked at the emergence of restricted connectivity resulting from a bias towards short connections in models where the neurons have positions specified in a three dimensional space. In this section I will show that modularity will emerge simply by restricting the proportion of connections, without regard to the neuron positions and connection lengths. With a given pattern of connectivity, evolution will surely arrange the neurons and connections to minimize the volume of connections, but restrictions on the connectivity proportions alone is sufficient to lead to the evolution of modularity.

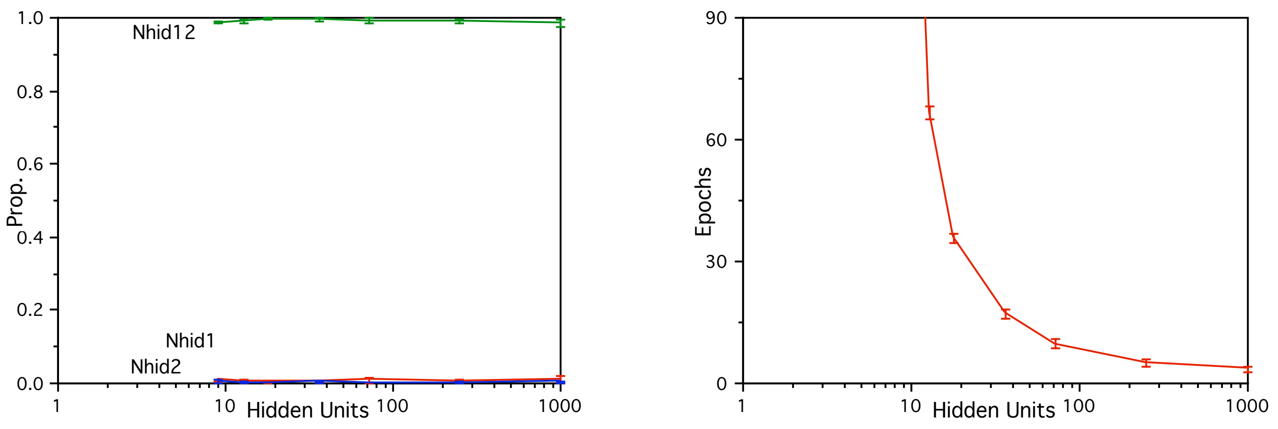

Figure 3: Dependence of the evolved what-where neural network results on the total number of hidden units. The architecture parameters as a proportion of the total number of hidden units (left), and the number of epochs of training required (right).

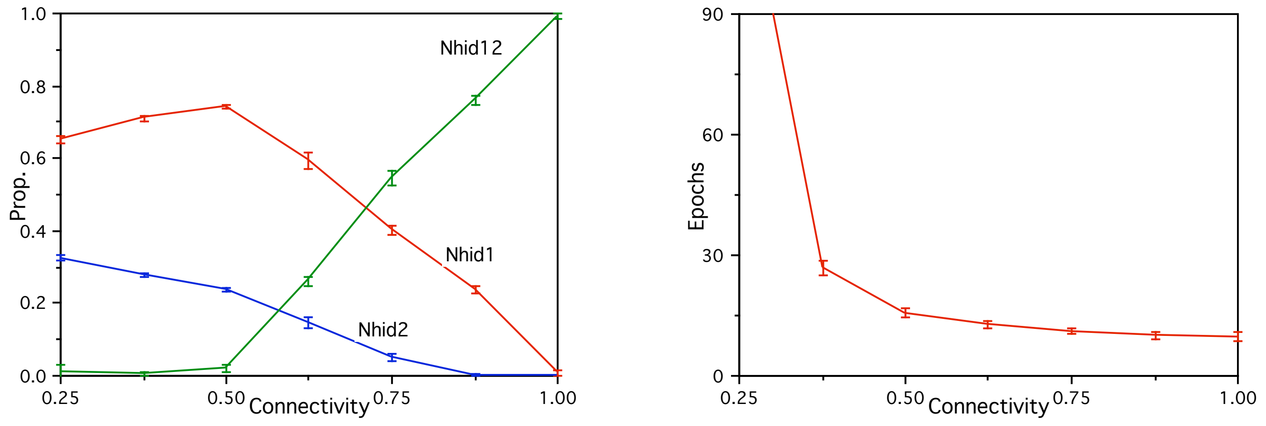

Figure 4: Dependence of the evolved what-where neural network results on the degree of connectivity between the network layers. The architecture parameters as proportions (left), and the number of epochs of training required (right).

The above simulations can easily be modified to test these ideas – one simply has to repeat them with the degree of connectivity between layers restricted to some fraction *f* of full connectivity. Figure 4 shows the architectures that emerge if we have 72 hidden units in total. As we reduce *f*, the number of hidden units shared by both output tasks, *Nhid12*, falls almost linearly until *f* reaches 0.5 and then stays around zero for all lower levels of connectivity. This means that a modular architecture makes the most efficient use of the available connections if they are limited to the extent that is found in real brains. 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 appropriate relative size of the two modules varies with the connectivity proportion.

More Realistic Learning Tasks

We have now established that modularity will only be an advantage for learning the what-where task when there are constraints on the proportion of neural connectivity, but it is not obvious that this will be true of all tasks. A particular worry is that learning a small set of input-output mappings for the 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 some continuous data distribution.

A typical problem humans are faced with is to classify in various ways new inputs drawn from some continuous space 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 are 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 could 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). We require our neural networks to learn the classification boundaries in our two dimensional input space for each output task, from a continuous stream of examples. Obviously, even for this

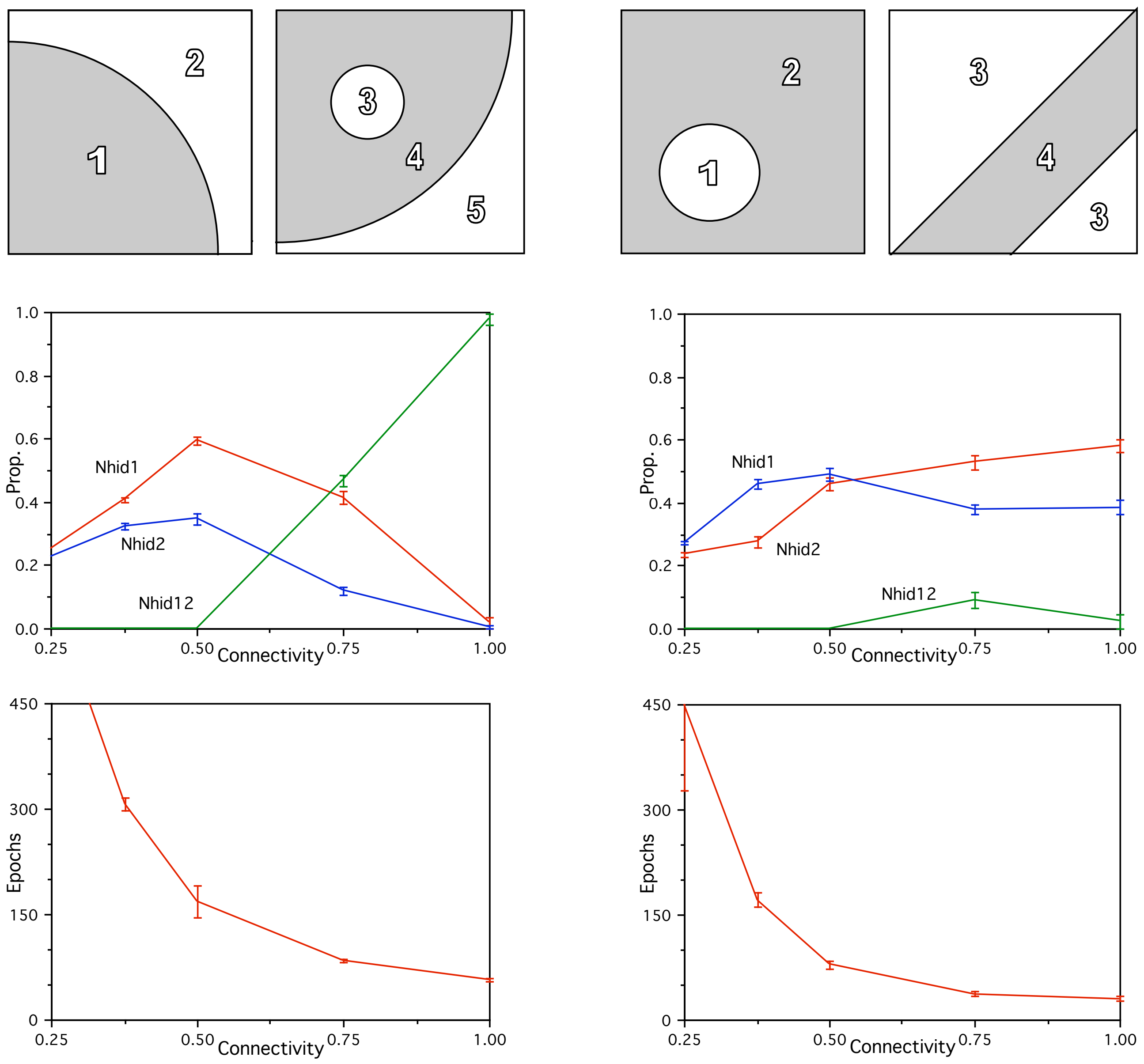

Figure 5: Results for two online generalization problems. The two pairs of classification boundaries (top), the architecture parameters as functions of connectivity proportion (middle), and the number of epochs of training required (bottom).

simplified set-up, there are an infinite number of possible tasks corresponding to different classification boundaries. The question is: will a separate module for each output task consistently work better or worse than a fully distributed network, or will the need for modularity be problem dependent? I attempted to answer that by re-running the above simulations with everything else the same except for the training data and the fitness measure. Here fitness is the ability to learn quickly to generalize, i.e. to produce the right outputs for each new item *before* training on it, rather than producing the right outputs for each item after many epochs of training on it. In practice, the infinite sequence of possible inputs was presented in blocks (or epochs) of 400 training items, and fitness was measured as the number of blocks required before a full block of items was classified correctly before training on each item.

It did not take many simulations to find the answer 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 problems shown in Figure 5 illustrate this. The case on the left has one two class task and one three class task. We see that for full neural connectivity, the optimal architecture is non-modular, and that as the degree of neural connectivity is reduced, the degree of modularity increases, as for the what-where case earlier. The case on the right has two two class tasks. In this case, a modular architecture is found to evolve for any degree of neural connectivity. As one would expect, for both cases, the average amount of training data required to reach a block of perfect performance increases as the connectivity, and hence computational power, is reduced.

A final complication is that we need to check that the evolution has not become stuck with an architecture worse than the global optimum. So, all the simulations were run again 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 fastest learning performance for each task.

Discussion and Conclusions

This paper began by reviewing the previous attempts to understand the advantages of modularity in neural systems, and how evolution could translate those advantages into brain structures. These studies were extended here by allowing the neural learning algorithms to evolve alongside the architectures, and by investigating more realistic learning tasks. We 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 computational power, and modularity does evolve. For artificial systems then, the need for modularity is problem dependent, and it is proving difficult to formulate general purpose heuristics to tell us when there is an advantage to modularity, and when there isn't.

Cognitive scientists, of course, are interested in biological brains, rather than artificial systems, and their models are further constrained by various physical properties. Once we incorporate the known physical constraints on the degree of neural connectivity (Chklovskii et al., 2002) into those brain models, modular architectures are found to have a clear advantage when it comes to learning efficiently, and simulated evolution does lead to the emergence of modular structures, for all the pairs of simplified tasks considered. Understanding how these modules actually come about in real brains is something that still requires more detailed simulations, including more realistic neural structures and connectivity patterns, as well as incorporation of the known stages in human brain evolution. I hope to report on such simulations in the near future.

References

Bishop, C.M. (2001). *Neural networks for pattern recognition.* Oxford, UK: Oxford University Press.

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.

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

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

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

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.

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.

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

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.

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

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.

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.

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

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