Evolution and Devolved Action: towards the evolution of systems.

C. MacLeod, D. McMinn, A. B. Reddipogu, N. F. Capanni, G. M. Maxwell.
School of Electronic and Electrical Engineering,
The Robert Gordon University, Aberdeen.

Introduction.

The Artificial Neural Networks group at the Robert Gordon University has, over the last six years, built up considerable knowledge and practical experience in Evolutionary Artificial Neural Networks. This experience started with the PhD project by C MacLeod [1] and continued with the work of D McMinn [2] which is due for submission in June 2001. These are being followed up by the work of research students A B Reddipogu and N F Capanni.

Initial work concentrated on Neural Networks that could grow to fulfil their function. C MacLeod, in his thesis on this topic, proposed a model of a robotic control system to be used as a vehicle for further research. This model formed the basis of D McMinn's project. The robotic control system [2, 3, 4] has a defined modular structure that enables the researcher to create networks for particular tasks and so allow the robot to function. McMinn has used this structure successfully as a basis to develop Evolutionary ANNs implementing Central Pattern Generators and Reflexes for robot locomotion.

It has become apparent, over the course of these projects, that a network that can evolve into a modular structure without the need for designed partitioning would be the next step forward for the group's research. This should allow the network to develop naturally and in an open-ended way without the need to artificially constrain it. Such an approach needs an evolutionary algorithm that can automatically and naturally evolve a "system": that is, a modular network rather than a fully interconnected homogenous structure. No acceptable Genetic or Evolutionary Techniques are currently available to do this. The group therefore needed to look to nature and discover the reasons why natural systems allowed such modularity to evolve and how it might be exploited in the course of future work.

It was felt that this work would be a culmination of the group's research to date, both furthering the work on robotic control systems and also including ideas from previous practical work, by MacLeod and McMinn, in terms of Evolutionary Networks and Incremental Evolution (Embryological Algorithms) [1]. This is because such a network will need to evolve and to "grow" from a simple to complex structure. The research and ideas that lay the foundations for this work are contained in this report. The report also considers two related aspects (neural learning and neural function), but as explained above, it is the layout topology or wiring which is felt to be the most important topic. After all, complex systems may be built up from simple elements, such as transistors or gates (or indeed, at the most basic level, atoms), providing that they are interconnected in the correct manner.

Therefore, to summarise: We are concerned with discovering an evolutionary method capable of evolving *systems*. To establish a method capable of this, we must look

again at the action of biological evolution and compare it with current artificial evolutionary methods.

PART A – A critical review of biological evolution.

A.1 Artificial Evolutionary Methods.

ANNs are usually directly coded into a GA [5], if such is to be used for topology evolution, each node or connection being part of the chromosome. However, the entire human genome does not contain enough space to directly code even a small part of an actual biological brain (let alone the rest of the body). So we must look again at the action of real genetics, as this must hold a clue to understanding the complexity of the biological system.

A.2 Biological genetic action.

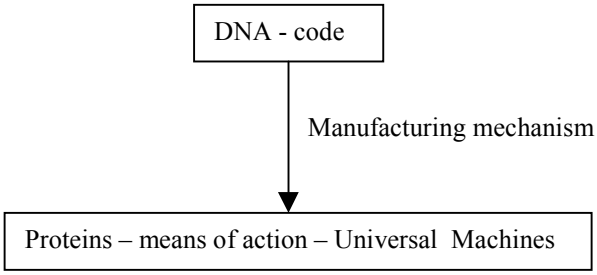
DNA is a code arranged in the form of a long string. Each position in the string has one of four possible values; these correspond to the four nucleotide bases. This code is transcribed by a rather complex machinery to amino acid. Three bases code for one amino acid (the three bases are known as a codon). This would allow a total of $4^3 = 64$ possible amino acids to be specified. However the code has redundancy built in (which reduces errors) and so only 20 amino acids are actually coded.

The proteins are made of these amino acids, joined together. Thus the DNA, by coding a string of amino acids, codes proteins (essentially, one protein = one gene). All that the genetic material does is code proteins and nothing else. The proteins are made by the amino acids joining together via peptide bonds (the joining is done automatically by the machinery which transcribes DNA to Protein). There is no defining line between a polypeptide and a protein. The genome of a simple organism such as e. coli can code about 4000 proteins. A human genome can code some 30,000 and 60,000 is considered the practical limit (due to problems with mutation) [6].

A.3 Proteins are the really clever part of genetics.

Every action in biology is mediated by proteins; they are the *Universal Machines* which make all biological processes possible. If other chemicals are required, they must be made with the aid of proteins. This vital point is summarised in figure 1.

Fig 1, Relationship between DNA and proteins.



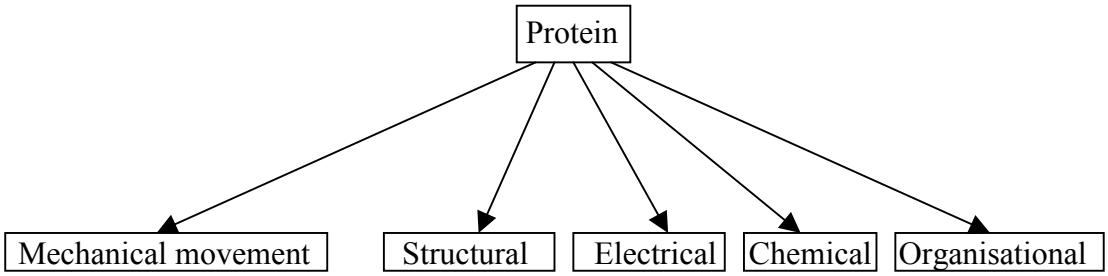
Because DNA cannot directly control the organism, we may say that the system exhibits a *Devolved Action*. All control proceeds via the proteins (the contrast with the artificial version is obvious).

Therefore, the system has two functional components. Firstly a code which can be mutated and exchanged through breeding. Secondly the machines (ie the proteins) which the code specifies (perhaps *part* is a better description than machine. This is because proteins can self assemble into more complex structures. However, it is worth remembering that there is much more to proteins than just self-assembly).

Examples of proteins:
Actin - Protein which operates mechanically to produce muscle contraction.
Haemoglobin – Carries oxygen supply around the body.
Enzymes – Catalyse chemical reactions.

So, these substances can perform almost any task. Figure 2 shows a tentative classification.

Fig 2, Proteins as Universal Machines.



It is astonishing that nature has one group of substances which can perform all these functions. The classification also shows up some of the limitations of organisms; there are some things which proteins cannot achieve.

One could, in fact, imagine a great map of all possible genotypes (DNA codes) to all possible phenotypes (resulting protein group structures, some of which would be organisms). Once such a process is understood, it opens up the possibility of "biological engines"- artificial biological structures engineered via DNA manipulation to become, for example, motors or other mechanisms. Imagine an engine running on grass in the same way a cow does. Such bio-engines might more resemble bagpipes than internal combustion engines.

All this throws up a problem for the designer of artificial systems: *The biological system is not directly coded as in the current Artificial Evolutionary Algorithms - what is coded are the universal machines which can assemble themselves or other parts into a system.* There is presently no way of synthesising such universal machines. Therefore, any system we design will be less adaptable than nature's machines (although it is arguable that a similar effect might be achieved by a code being supplied to a *Universal Manufacturing Machine*; such a system would not be able to interlock as discussed below). In the sections that follow we will consider how some of the most important of these machines operate.

Of course, it might be argued that even the process described above is not truly mimicking nature. After all, why not start with the most basic components, which occur naturally in the system - Electrons, Protons and Neutrons? These self assemble under the control of thermodynamic forces into molecules. One can argue, indeed,

from this point of view, that the current search in physics for a "theory of everything", is no more that the search for the initial rules governing the way fundamental automata assemble. However, it should be borne in mind that, apart from life, evidence of organised complex dynamical structures in nature is hard to find. Most structures form into "lumps" like familiar rocks. What complex structures do exist - crystal and dendritic mineral structures, for example, are the produce of simple automata rather than complex processes like those present in life. DNA and life can therefore be seen as having rather unique properties (perhaps because of their complexity). A point further underlined by the absence of life (or even evolution) based on any other chemical system, produced either in nature or the laboratory.

A.4 The importance of interlocking.

The proteins themselves can effect the DNA by binding to it and stopping it manufacturing other proteins. They can also effect each other and work together to create much more complexity in the system (see descriptions of signal peptides and adhesion molecules, below) [7]. Such behaviour may be termed *interlocking*, in that the different parts of the whole have evolved to function as a system.

Interlocking is a vital part of the organisational ability inherent in biological organisms and allows them to exhibit complexity which would be impossible through separately functional components.

Interlocking is puzzling because it is difficult to understand how one machine can evolve separately from another and still show inter-dependence on it. The explanation lies in the fact that proteins are all made from combinations of the same 20 amino acids. So eventually natural mutation will produce one protein which effects another in a beneficial way. In other words, the evolutionary search distance between proteins, which can effect each other, is not too great. This is dramatically illustrated in the genetic disease called sickle cell anaemia. In this disorder, there is a mutation in the coding for haemoglobin. This mutation causes the protein to form long thread-like structures within the cell and this causes problems with its function. However, it shows that a small mutation can change a protein from one machine (which carries oxygen) to another (a self assembled structural protein). So the crux of the matter is: *Because proteins are built from the same limited number of units, they can interact and relatively small variations can completely change their function* (and occasionally into something useful) [8].

A.5 The power of the universal machine – Protozoa.

Such is the power of this system that it can design 'intelligence' without the need for neural circuitry. For example, consider the Protozoa; these are single celled animals which display almost unbelievable complexity and apparent intelligence (defined loosely). All this is achieved by protein chemistry. They show that with a powerful organisational system, perhaps the individual components need not be complex 'processing units' such as neurons [9].

A.6 The perfect artificial evolutionary system is not achievable with current technology.

We can now see that the most general Artificial Evolutionary System would consist of a mutatable code, which specifies general universal machines (and the process which turns code into machine).

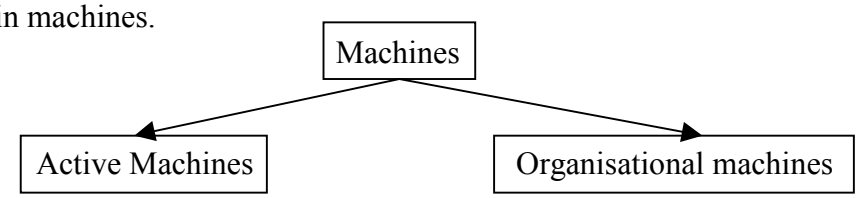
The general universal machine itself would be a machine capable of forming any mechanism or structure, capable of fully interlocking, with a small evolutionary distance between itself and others, and finally, capable of forming associations which can direct their own organisation.

If we apply selection (natural or artificial) to such a system, we could eventually evolve any mechanism. Although this is currently impossible, a method will be suggested in which a more restricted set will prove useful.

A.7 Active Machines and Organisational Machines.

First, let us make an important distinction between two types of protein. Those which form the active processing machinery and those which direct the construction of such, figure 3.

Fig 3, Protein machines.



If we can tackle the organisational part of the system, perhaps its power may be used (as demonstrated by the protozoa) to build complex structures without the need for a completely universal (active) machine.

A.8 Organisational proteins.

Presently, we do not understand all the mechanisms of organisation within an organism. However, what is known is powerful enough to solve many of the organisational problems we are concerned with. There are several different types of organisational protein identified. These operate in quite different ways [10].

One example, signal peptides, act like keys, allowing proteins which have the signal peptide chain attached to enter different organelles of the cell. Generally, once the protein has entered, the signal peptide (which forms only a short part of the total protein chain), is discarded. Proteins get distributed around the cell by the normal thermodynamic forces which act on all chemicals within the cell [11].

A.9 Cellular structure.

Why do all known, large organisms maintain a cellular structure? The answer to this question lies in the fact that the mechanism of chemical distribution in the cell is driven by thermodynamic forces as just described. The molecules are buffeted by other constituents of the cell and eventually, because of the small physical volume

involved, will find their way to the area where they are required (for example, be admitted by a gate to their signal peptide into another section of the cell)[12].

For billions of years organisms existed as single cells before multicellular structure developed. With the organisation of earth's biochemical system (which may be only one of many possible), multicellular organisation is necessary to achieve sizes greater than about 1mm.

A.10 Gradients.

The chemical signals which allowed signalling inside cells were adapted to allow signals to propagate from cell to cell. In fact, all multicellular organisms have extensive intercellular signalling built into their chemistry.

Probably the most important of these intercellular signals are *gradients*. If one cell emits a chemical signal, it reduces in concentration as it travels out from the cell which emitted it. Cells have a built-in threshold to these chemical gradients and can switch off or on the expression of proteins according to whether they are receiving enough of the signal or not (once a protein is switched on, it can maintain itself by positive feedback, that is it supports its own production. It can also block other proteins by binding to their DNA sites). This mechanism plays a pivotal role in the differentiation of cells in the foetus and in their eventual position in the body - and so the formation of a patterned organism. For example, cells migrate along gradients (so they control the eventual position of cells) and gradients can cause the cell to switch on the expression of proteins which might change it into (say) a muscle cell (so gradients control cell function). The symmetrical nature of gradients is the reason why animal bodies display radial or axial symmetry. It also explains the fractal nature of some biological systems (because, as each chemical gradient triggers, it produces divisions at successively smaller levels, so producing a self-similar pattern at successive dimensions) [13, 14, 15].

A.11 Neurons.

The neuron developed to allow long distance signalling. This is important because, for the reasons listed above, there are limits to intercellular signalling, particularly in terms of speed (and gradients like other chemical signals, which are not re-triggered in each cell, only operate over a distance of less than a couple of millimetres).

So Neurons developed with long cellular processes which can stretch over a much larger distance. Once they had this property, they could also develop a more specialised signalling apparatus. This is important if any organism is to react to its environment quickly (chemical signals are slow), with the obvious evolutionary advantage this would give the organism. In fact, neurons are one of the most ancient of all cells. They also demonstrate another type organisational protein, those which control cellular adhesion. These cause some cells to be able to adhere to each other and not to others. This appears, along with gradients and cellular migration, to be a major factor in neural organisation [16]. Neurons also use gradients to direct their long processes to their destinations (axon growth factors), something which, if we could harness it, could direct the wiring of three dimensional integrated circuits.

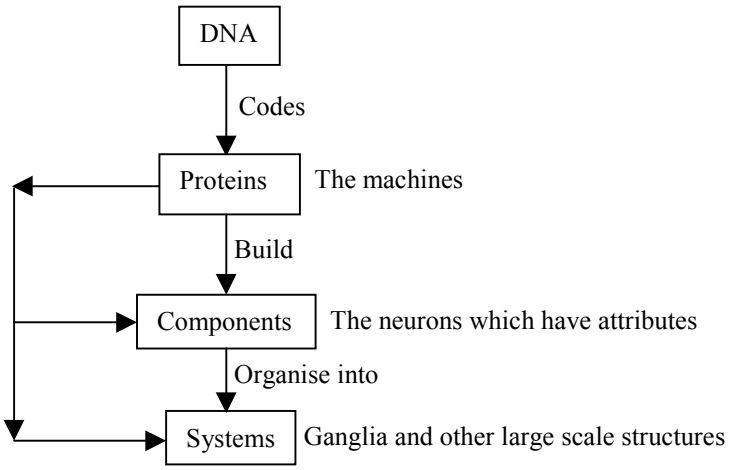
PART B – A New approach to ANNs.

B.1 An Evolutionary Artificial Neural Network.

Now, all this may have left the feeling that we know what goes into an evolutionary system but we couldn't replicate it because we can not replicate a universal machine (at least at present). However, while a full system is currently beyond our capabilities, we can identify, for a neuron, which parameters are important and so limit the case to something more manageable. However, to do this, we need a cunning scheme to replicate some of the diversity which nature provides.

We should recognise, firstly, that what the universal machines build are cells and what they give to these cells are a series of properties or attributes which aid self-organisation. For example, attributes include the ability to adhere to other cells or the ability to migrate along gradients. One way of looking at this, which helps us to simplify the problem, is the hierarchy shown in figure 4.

Fig 4, DNA and proteins as part of a systems structure.



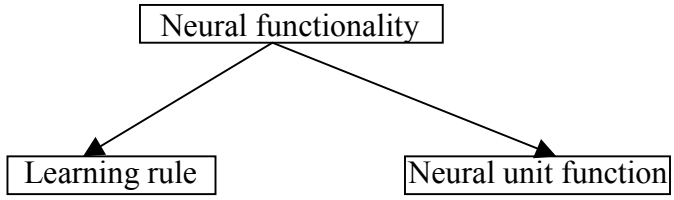
Obviously, it isn't quite as simple as this because proteins outside cells (in the cellular or developmental environment) help organise components and also play an important part in the functioning of the system. This is shown by the side arrows in the diagram. However, this idea does give us more to work with than the most basic systems outlined earlier.

The problem is best split into two different parts. Firstly, the function of the neuron and secondly its connections (this is where the modularity of the network arises). These two parts, in a way, are roughly equivalent to the distinction between active and organisational machines. It is to the second of these that we will mainly apply our evolutionary ideas.

B.2 Components necessary to make a general neural system.

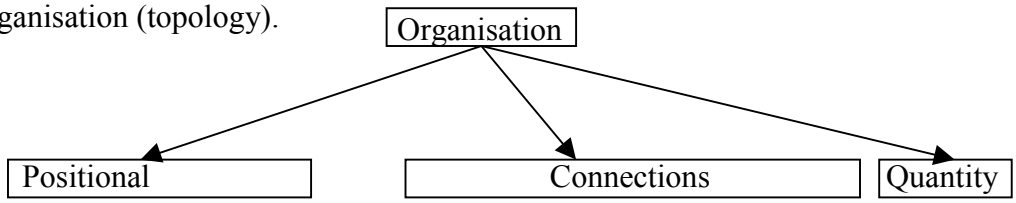
First let us consider the functionality of the neuron, what is required here? Well, there are really just two components, figure 5.

Fig 5.



Secondly, we need to consider the organisational mechanism of the network. We can identify the most important aspects of this from nature; these are shown in figure 6.

Fig 6, Network organisation (topology).



The importance of these will be become obvious as we proceed.

B.3 Functionality.

Let us consider the two components of the function of the neuron.

B.3.1 Unit function.

Contrary to popular belief, all neurons do not have the same unit functionality. Actually, some operate in quite different ways from each other. In fact we do not fully understand the operation of some of the more exotic types. What is needed is an evolutionary system which can evolve any reasonable neural function. This would mimic the universal machines within the cell which can arrange themselves to produce a wide range of electrical behaviours.

Although we could simply choose a combination of operators genetically [17], we can also model any function using an infinite Power Series (for example a Taylor Series):

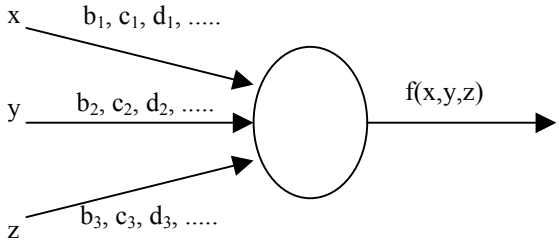
$$y = a + bx + cx^2 + ... + \delta_n x^{n-1}$$

This is the basic series which is given in most text books, However, it is extendable to any number of variables (and hence any number of dimensions)- for example in 3 dimensions:

$$f(x, y, z) = a + b_1 x + b_2 y + b_3 z + c_1 x^2 + ...etc$$

If this were an ANN x, y and z would be the three inputs and b_n, c_n and d_n would be the respective weights as shown in figure 7.

Fig 7, a polynomial ANN.

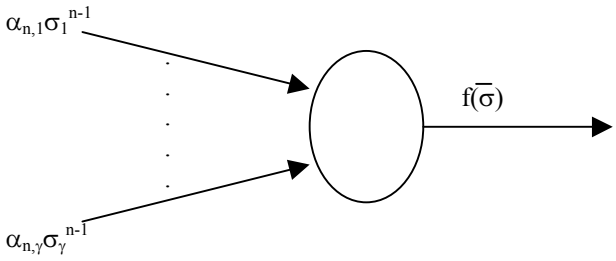


Or, more generally, for γ variables and a χ order series.

$$f(\bar{\sigma}) = \sum_{n=1}^{\chi} \sum_{m=1}^{\gamma} \alpha_{n,m} \sigma_m^{n-1}$$

Shown in figure 8.

Fig 8, a more generalised neuron.



As an example, we could approximate a circle of radius $\sqrt{a}$ with a two variable second order Power Series (coefficients $\bar{b} = 0$).

$$f(x, y) = x^2 + y^2 - a$$

We could also generalise the series more by adding fractional powers $\sigma_m^{1/n-1}$ or products of inputs $\sigma_m^n \sigma_j^i$, etc. Actually, the normal McCulloch-Pitts neuron is nothing more than a first order series. If the term *a* is included, it corresponds to bias in the network. It should be obvious when one considers that the separator in a normal neuron is a straight line that a first order power series is simply a straight line approximation to a function. A first order network may approximate any function provided there are enough neurons in the system. The coefficient vectors a, b, c, etc are weights which must be trained. So we have a system which can emulate almost any function of the neuron (which has a continuous response). Now let us consider time response.

We could model any time response with another Power Series:

$$f(t) = a + bt + ct^2 + ...etc$$

but there is no evidence that the temporal properties of neurons are this complex. So it may be easier to simply make a evolvable (or trainable) time decay:

$$f(t) = ae^{bt} \qquad (a, b \in \mathbb{R})$$

By applying a threshold function to this it is easy to produce a good approximation to "spiky" neurons. Usually a would be 1 and b negative, and so our general neuron has this function:

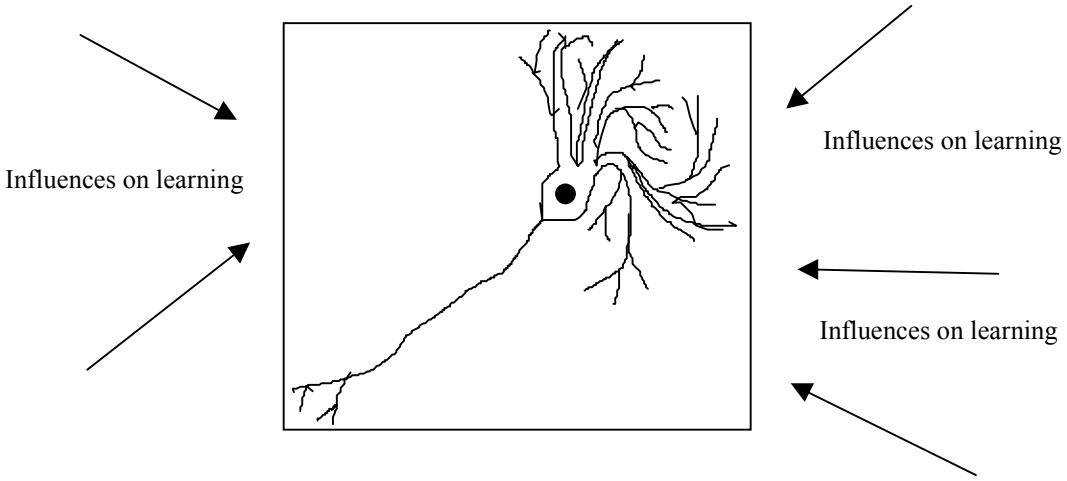
$$f(neuron) = T(\bar{i})f(t)$$

Where T($\bar{i}$) is a nth order Power Series of the input vector $\bar{i}$. A squashing function may be applied to this if necessary and we could also add a refractory (rest) period [18].

B.3.2 Learning.

How could such a system learn? Consider a biological neuron; what are its possible learning mechanisms? We can find the answer to this problem by drawing the isolated neuron - a "neuron in a box", figure 9, and writing down all the *possible* parameters which could allow it to learn (After all, the neuron is only connected to other neurons. So the only possible mechanisms involve either signals transmitted from other units or chemical signals from the surrounding medium).

Fig 9 Isolated "neuron in a box"



Some examples might be:

a) Biochemical. The neuron is bathed in a 'soup' of intercellular fluid. Hormonal and other stimuli can affect this soup. For example, its constituents might change in the presence of hunger, pain, fright, the urge to mate, etc (or, in a simple system, simply a good / bad signal). The signal would effect whole regions, not individual synapses. Let us call this component B.
b) Hebbian (or Anti-Hebbian). A synapse gets strengthened through use. The more activity it has, the stronger it becomes. This contribution effects each synapse individually and is therefore a matrix. Let us call this component $\overline{H}$.

c) Synchronous (or Anti). A synapse gets strengthened when it is active at the same time as others (and weakened if it is not). Like the Hebbian contribution, this is therefore a matrix. Let us call this component $\overline{S}$.
d) Mediated. One synapse (or a group of them) controls the strength of an other.

$$M_{i,j} = ax + by + ...etc$$

Where a, b, etc are constants defined by evolution (or learned), x, y etc are other inputs to the neuron. This effect may be difficult to achieve in practical terms, and since there is no evidence for it from biology, it may be best left out of a model.

The mathematical forms of these learning mechanisms have not been expanded because they are reasonably obvious. Of course the programmer may choose to add one of the traditional learning methods to (or instead of) these.

The total learning contribution to the weight matrix could be expressed as.

$$\overline{W^+} = \overline{W} + \eta(aB(b\overline{H} + c\overline{S}))$$

W^+ = updated weight matrix. η = learning rate. a, b, c = sensitivity to individual learning types. W = old weight matrix.

It is fairly obvious that these mechanisms, which are biologically realistic, are highly dependent on network topology. After all, a synchronous or mediated learning strategy will only work with neurons which are placed in the correct positions within networks – it is easy to see that in other positions they could have no effect or cause the network to deviate away from the required response. They are therefore probably only suitable for use with networks whose topology is defined using evolutionary mechanisms. This topology dependence may be the reason why it has been so difficult to decide on biologically feasible learning mechanisms for ANNs.

B.4 Organisation

It has already been explained that, of the three elements of network specification, (learning, neural function and connection pattern), the connection pattern or topology of the network would appear to be particularly important. One can perhaps see this if one considers that all machinery, whether artificial or natural, is made from smaller elements, indeed, in the ultimate analysis from atoms themselves. It is merely a matter of selecting and placing (and in the case of neurons, connecting) these elements to produce the response required.

In the case of neural networks (and also electronic systems is general) an important aspect of the system is its modularity. In the case of the robotic system presented in D McMinn's thesis, the network has been artificially partitioned into Reflexes, CPG and Biological Oscillator. When one attempts to evolve this system as a fully interconnected homogeneous network, the result is failure - or at least considerable difficulty. Therefore, the standard genetic algorithm is not suitable and we must look to a new paradigm – one which has the evolution of a modular system integrated into it.

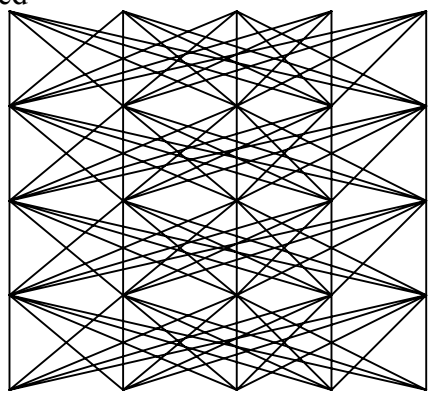
Before embarking on an exploration of the methods capable of doing this, let us first pause to consider *why* a fully connected network seems incapable of producing the results we require. Work needs to be done on the theoretical basis of this question as there seems to be a theorem missing from systems theory which adequately explains it. However three possible avenues of enquiry might be:

a) The fully connected networks currently used are too small.
Almost all networks are limited by orthodoxy to three layers. Kolmogorov's theorem is usually cited to support this view. However, this assumes that the problem is a single data domain solvable with straight line separators. In practice however, the problem may involve multiple data domains or multiple data transformations and so the simple three layer model may not be adequate.

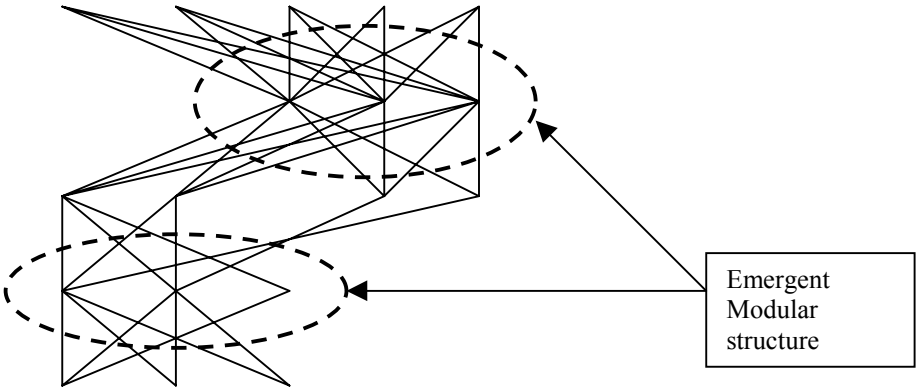
In theory a homogeneous "deep" network like that shown in figure 10 should, through its learning process, become a modular network as shown - the deleted weights becoming very small. That such networks are not often experimented with may be one reason why we don't observe this phenomena. Another could be that BP is not a suitable algorithm for training such a system (although a GA should do it).

Fig 10, a "deep" network.

i) Fully connected



ii) After it has learned (deleted connections have small weights)



b) The networks have to process separate datasets.

To illustrate this point let us take an example. Suppose that we have inputs from a complex visual system which requires several layers of processing (for example feature detection) and also inputs from an equally complex auditory system. If we tried to process both of these in the same network it would cause difficulty because the two independent datasets are not separated and can interfere with one another. the same argument might also apply to complex operations even within the same set. So separate discrete networks would certainly make this processing task easier.

c) A single large network represents too large a search space for the evolutionary algorithm.
Several smaller networks would represent a smaller search space, but in McMinn's networks the size was small and the single large network still did not converge well, which indicates that this explanation may be false.

B.4.1 Organisation - methods
Let us now turn our attention to some practical methods of introducing modularity into the network. At the simplest level, there are two basic approaches to this. One could adopt a method based on developmental biology, using computer models of cellular differentiation, migration and pattern formation to fashion a network. Alternatively, we could take an engineering standpoint and try and extract the essence of what is required to produce a modular network and code this from a purely pragmatic point of view. However, whichever path one chooses, there are certain obvious aspects that the algorithm will have to accommodate:

1. Position.	In biologically inspired algorithms, the neurons must be placed in the correct positions in the "evolution space" or body.
2. Number.	At each position the algorithm will have to decide the number of neurons to be present in each module and the number of modules.
3. Function.	What the units actually do in each position.
4. Connections.	Connections must be established both locally between neurons in the same module and globally between different modules.

Although these ideas may be implicit in all the solutions resulting from the different ideas described below, some may be easier to implement and more successful than others. Outlined are five methods for creating modular networks, these are:

i) Using models of biological neural development.
ii) Using a set of production rules.
iii) Using fractals or similar mathematical models of natural patterns.
iv) Adapting traditional evolutionary techniques such as the Genetic Algorithm.
v) Using Direct Modular Growth on a network.

Let us consider each of these in turn.

B.4.2 Modelling Biology
In biology, the formation of patterns in the organism is not particularly well understood. There are however certain generalisations we can make which are probably sufficient in themselves to allow us to develop an Artificial Neural Network that can evolve modularity [20]. The processes involved are differentiation,

proliferation, migration, and patterning in normal cells. On top of these, there is cell orientation and connection formation in the case of neurons. The order in which these occur may be different in different cell types. Let us consider these aspects separately.

In the new embryo, the cells have no set function. Differentiation is the process whereby these "stem cells" become specialised to be bone, muscle, neurons or one of the many other cell types in the body. When the fertilised egg cell attaches itself to the womb lining, it may be a chemical signal from the mother that starts development. It is thought that a chemical gradient starts cell differentiation. Cells further away from the point of attachment in the womb may be below the threshold level, whereas those closer are above it. This triggers a change in the cells closest to the womb lining that is the start of the road to development. Large aggregates of cells differentiate first, forming gross structures; these in turn start producing other gradients which allow differentiation to take place on a smaller scale and so on. The *birthdays* of the cells (when they differentiated) are important in this respect, as earlier cells tend to form the larger structures and later cells the finer. As an example, in the limb bud of an animal, developmental genes progressively get switched on and form finer and finer structure, starting with the form of the limb and working down eventually to the digits.

Proliferation is tied up with differentiation. Obviously there needs to be enough cells to build the structures of the organism and since development starts from a single cell, the embryo's cells are continuously dividing and so that the cell mass becomes larger.

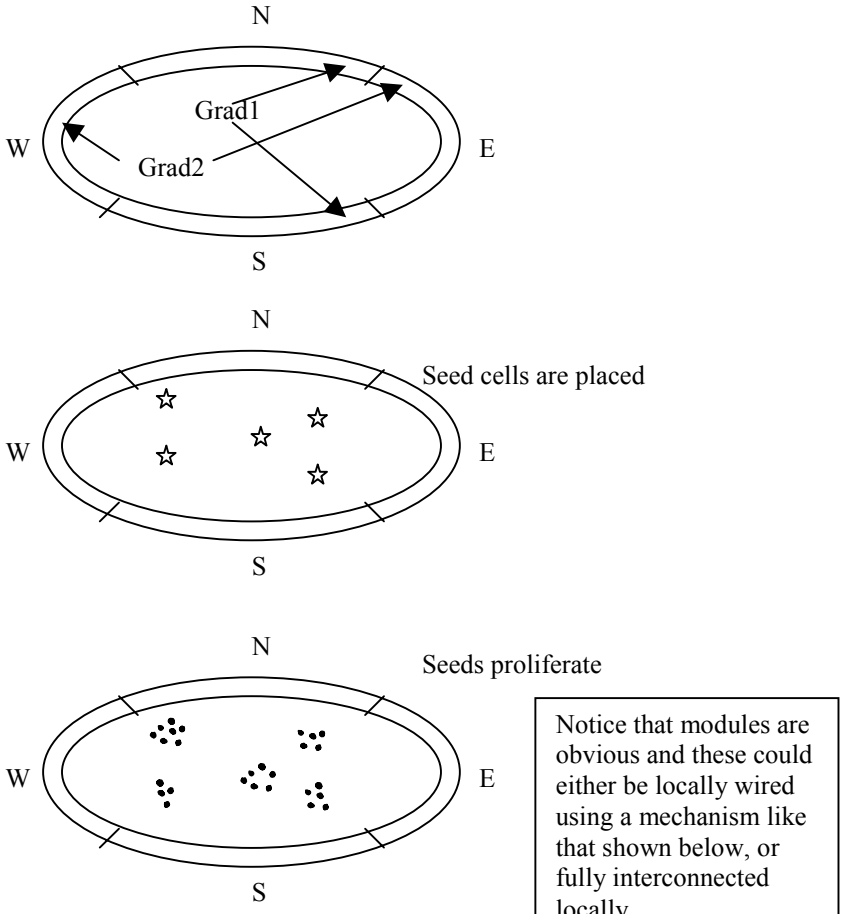
Migration is important in some cell types and in particular neurons. Basically the idea is that cells may proliferate and differentiate far away from the point where they are finally going to be used and so they travel or migrate to their final homes. Exactly why this evolved is not particularly clear, but it may be an accident of evolution. Neurons originate in the same layer as the skin (they may have originally been skin cells that became specialised during the course of evolution) and they have to move if other tissue is to become enervated. Migration is mediated by chemical attractors and cellular adhesion.

Patterning is the result of the previous processes. Cells can form patterns in situ by proliferation and differentiation – this is what happens with bone and muscle. As different cells form, they switch on new developmental genes, which form their own gradients, and spark the next level of differentiation and pattern detail. This mechanism results in symmetrical structures (because gradients are involved) and fractal structures (because larger patterns cause smaller ones to form in more and more detail). Whether migration is involved or not, these three mechanisms have the result that: *The right type of tissue is at the right place and in the right quantity.*

From the point of view of a computer model, we can simplify the picture to say that we need to place the correct quantity of cells in the correct place. This may be done if artificial gradients are set up which allow "seed cells" to be placed, figure 11. These may be assigned numbers which signify their attraction towards two perpendicular poles, therefore locating themselves in 2D space (3D space is obviously just as easy). The seeds then proliferate to form the correct number of neurons. One has got to make

sure that the genetic coding for this is such that when mutations or crossover occur they conserve positional and proliferation information.

Figure 11, Artificial Gradients can be set up in an evolution space.



Cell orientation and the formation of connections are particular to neurons as they have to form "wiring". Experiments show that axons form along chemical pathways ensuring that one part of the brain is connected, as appropriate, to another. The mechanism could be mimicked by allowing the neurons to grow connections from only one side, these would then progress outwards until they connected with another neuron which was compatible with the first (compatibility values could be attached to neurons). Or alternatively, all neurons (and the sensory inputs and motor outputs of the network) may be labelled with markers which specify attractors for the growth of connections from the placed neurons. Those which are closer or have stronger markers 'win' and connections are made towards them.

In schemes like this, numbers are attached to each neuron type which allow the units to find their own connection patterns. This idea is equivalent to proteins getting expressed in individual neuron types hence allowing them to self-organise. The organisational machinery is contained within the individual neurons and not in a universal genotype. Action is therefore devolved.

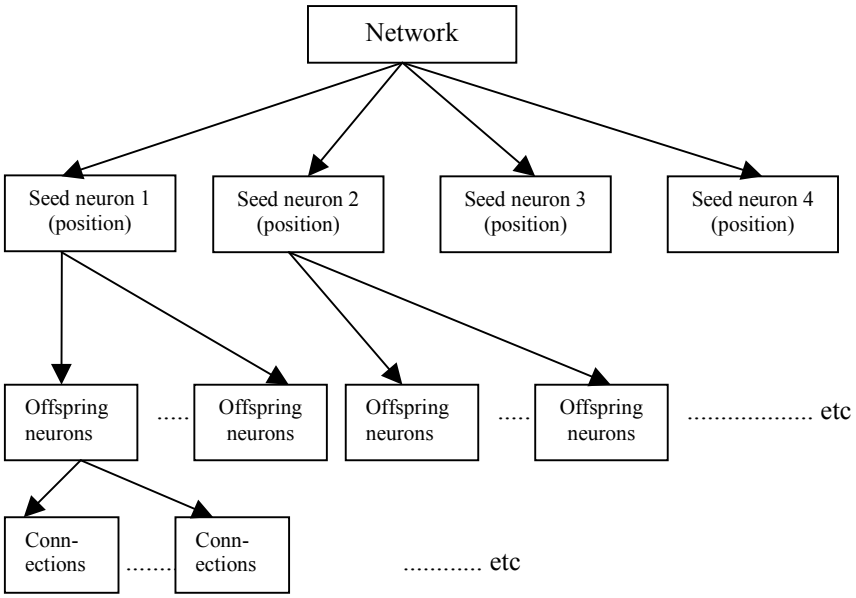
Of course this example is used simply to illustrate the issues involved in implementing a scheme based on nature. One could include much more complex

details from developmental biology, right up to creating a sophisticated model which simulated the effect of known genes (for example the hox genes), or axon growth factor. However, the basic idea remains the same and it is doubtful if this level of detail confers much advantage over a simpler algorithm that captures the essence of the system.

B.4.3 Production Rules and Trees

Another system which captures the biological approach, but at a more abstract level, is to use a set of Production Rules to configure the network. Either the rules, or the network that they produce, can be conveniently written down in the form of a tree - following the tree from top to bottom produces the network. A typical tree for encoding a network shown in figure 11 is shown in figure 12.

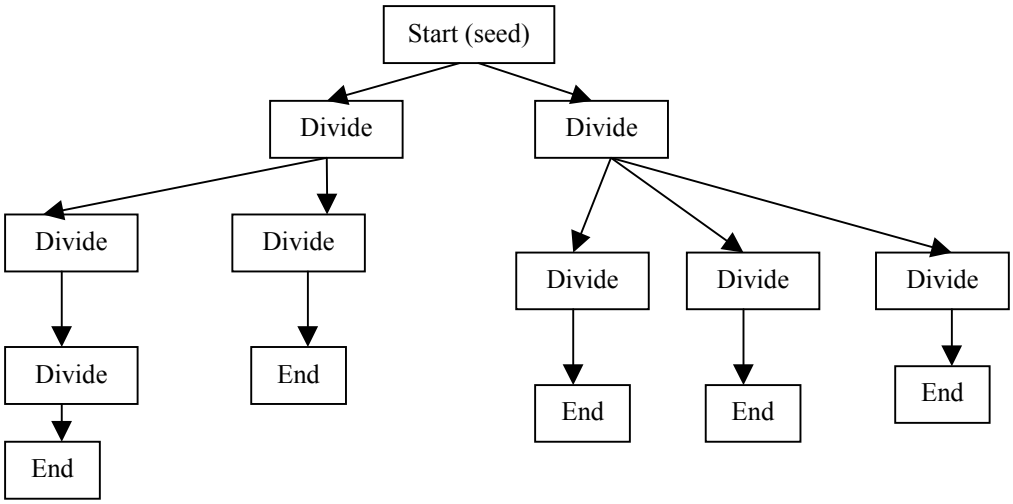
Figure 12, A tree encoding a network.



Of course this is really just an alternative way of representing the biological system which was illustrated in the previous example. The tree comes under the control of the Evolutionary Algorithm - which is basically the production rules for creating it. They might include, for example, neuron proliferation (turning one neuron into many), migration and so on. Arranging the system like this however has the advantage that one can allow mutation and recombination (crossover) in the G.P. fashion by swapping or mutating individual branches of the tree. In this way important sections of the network can be reused without having to be re-evolved.

This is just one of many possible trees which could be used to encode the network and many different production rules which could be used. The alternative to encoding the network is to encode the production rules themselves as the tree [19], figure 13, the rules in this case being produced by the Evolutionary Algorithm.

Figure 13, Production rules.



The connections may be part of the production rules or evolved through the hierarchy established by the rules themselves. The arguments about recombination and mutation outlined for the previous case apply here also.

B.4.4 Fractals

Many of the patterns produced by living organisms are self-similar at different scales. This phenomenon can be seen clearly in many plants - good examples are ferns. The study of such structures has become popular and important in recent years and some aspects of them (although not all) are tied up with non-linear systems and in particular so-called chaotic dynamics. Mappings of the stability of such systems produce the beautiful "Mandelbrot Set" and others. Other self-similar structures can be produced by the repeated application of a formation rule, by geometric transformations or by self-organisation in automata. However they are produced, such structures are often referred to as "Fractals" [21]. The word Fractal is widely misused and applies to many different types of system - here we are taking a loose meaning, understanding it to mean a mathematical model which forms approximately self-similar systems that resemble biological structures.

The importance of fractals is that they are often the mathematical form resulting from the action of genetics. They display symmetry (as does the biological nervous system) and are generally "lumpy" (modular). The idea that fractals could be used to define neural net topology has been suggested before [22], but researchers have yet to take it seriously enough to produce working systems. One of the main reasons for this is that it would require the design or discovery of a fractal system suited for the job (currently available fractals are not).

Not all types of fractal would be suitable for use in the definition of neural nets. There are three important attributes which the fractal should have. Firstly, it must be constructed from lines, because these will be the connections in the network. Secondly, it must be able to grow, from a simple to a complex structure, the later stages building on the earlier. Finally, it must have "nodes" at which neurons or modules can be placed.

There are three types of fractal (and perhaps several others) which almost immediately fit the bill. These are outlined below.

i) Dawkin's Biomorphs.
Biomorphs made their appearance in the book 'The Blind Watchmaker', by Richard Dawkins [23]. In the book, Dawkins explained the operation of evolution in animal populations. To aid his explanation of the evolutionary process, he developed a simple computer program which produced branching structures which he named Biomorphs. Each time the program ran, the Biomorph grew; that is, it added another layer to its structure, figure 14. Dawkins encoded a sequence of numbers which represented the structure of the Biomorph as a string and which can be mutated - hence the method is inherently genetic.

Fig 14, Typical Biomorph development.

In the original program, the Biomorph was represented by nine genes. The genes influence parameters such as the height and width of the structure. The Biomorphs always grew by branching into two segments (as a biological cell divides in two, during reproduction).

The process has several important properties:
- It represented a structured search and produced a structured result
- It allowed growth from simple to complex structures
- It was based on sound biology (although the final algorithm is not a direct model of nature).
- It appeared to be adaptable to neural nets

The system, as presented by Dawkins, is not in itself suitable for use with neural networks. However, it may be possible to use a similar algorithm as the basis of a system suitable for use with ANNs.

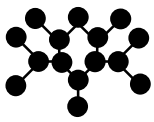
(ii) "L" (Lindenmayer) systems.
These are often seen in the context of producing beautiful plant-like patterns, particularly ferns and trees. They are generated through the repeated application of rules. Like the Biomorphs described above they can "grow" from simple to complex and can display line features. Other similar fractals include Pythagoris and Mandelbrot trees.

(iii) Cellular Automata.
Certain types of Cellular Automata can also be configured to produce similar structures to fractals, both symmetrical and non-symmetrical. They can form branch-like dendritic structures in the same way as the other fractals discussed. However these are built up from the self-assembling properties of the automata rather than from a mathematical formula, transformations or growth rules like the other structures. In the case of the automata, systems can be envisaged where the neurons are the

automata themselves, or alternatively, where the dendritic chains simply form the "wiring" of the network. Both "L" systems and automata are illustrated in Stewart's book [21].

The three examples given above are simply used to illustrate the possibility of the fractal use in ANN definition, very little real work has been done in this area. As mentioned, none of the systems, as they presently stand, are really suitable and a structure really needs to be designed which "fits the bill". However, once this is done, there are two way to use the fractal. Firstly, the nodes of the fractal could be used as placement points for neurons and the branches their connections; this is illustrated in figure 15 below. Or, alternately, the nodes could be placement points for network modules (probably fully interconnected). Looking at figure 15, one can see why the basic Biomorph is not suitable for use. However, other fractals do display fully interconnected modules with sparse interconnectivity between them.

Fig 15, Using a fractal to produce connectivity.



Black blobs are neurons or modules.

B.4.5 Altering existing Evolutionary Algorithms

One obvious area which we have not yet touched on is whether the standard evolutionary algorithms such as Genetic Algorithms or Evolutionary Strategies could be modified to produce a modular result. Some possible ways of achieving this are:

1) Define each module by a section of chromosome within the population of the GA. Each neuron appears as a sub-string within the module and for each might be coded: (1) The weights, (2) The function of the unit and (2) The wiring or topology. Some neurons are designated inputs and some outputs. These allow connections to other modules (for ease of coding the inputs and outputs can come at the beginning and end of each string and delineate it). As modules are added to the structure, the strings are allowed to grow.

2) An extension of this idea is to have a fixed string length for each module, but to allow certain parts of the string to turn on or off or define the network (akin to pruning). This is an attempt to get around the problem of strings having to grow if modules become bigger or alternatively, have an independent GA for each module.

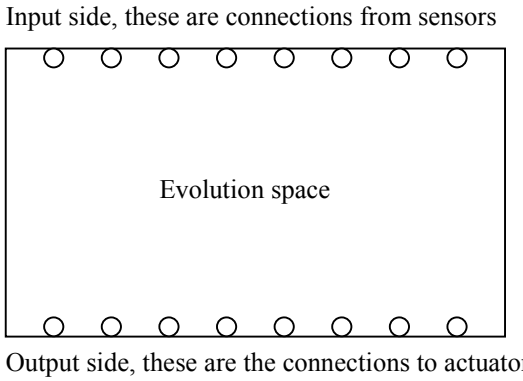
A new module could be created when the GA string reaches a certain fixed size - at which point it would automatically "bud off" a sub-network which would be wired independently of the first. Alternately this could be done once the network had fulfilled its function (once the fitness function was as high as possible).

B.4.6 Direct Growth

At the opposite end of the spectrum from the biological approach is a purely engineering point of view. From an engineering view we simply want to place groups

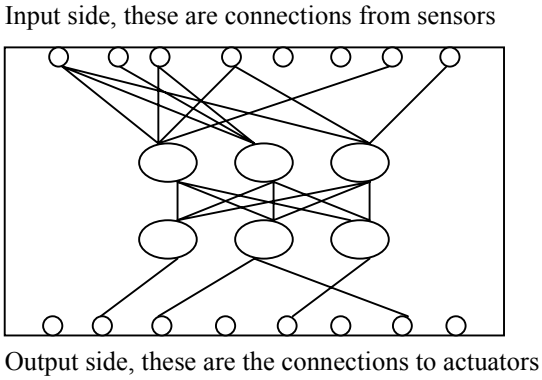
of neurons into a modular structure and connect them together to form a system under the control of evolution. There are a number of approaches we could adopt to achieve this. Consider the concept of an "evolution space" were the network will develop as shown in figure 16.

Fig 16, An "evolution space"



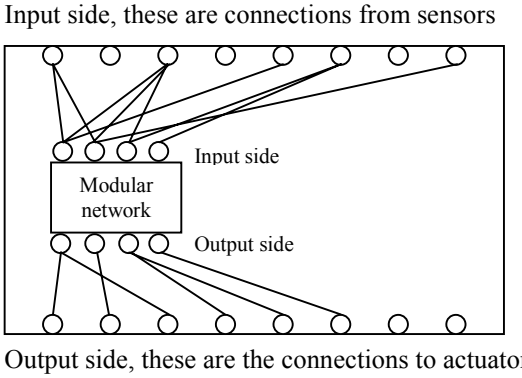
In the systems developed in McMinn's thesis and in previous projects a single network is caused to evolve in the space as shown below in figure 17

Figure 17, A network, as evolved in McMinn's thesis.



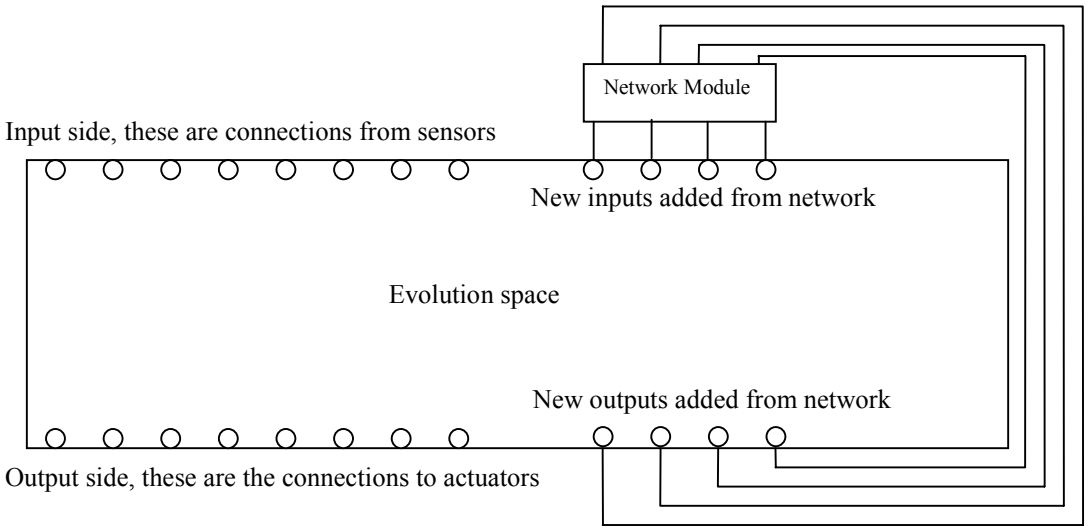
However, the concept can be easy adopted to serve modular networks in two obvious ways. Firstly, by replacing individual neurons in the diagram above by networks, figure 18.

Fig 18, Adapting the above idea to modular networks.



The internal wiring of the modular network would be determined by one evolutionary algorithm (the local algorithm), the wiring between networks and inputs/outputs by another (the global algorithm). The algorithm would start with few inputs and outputs and build up by added these and more modular networks (this is the incremental part of the system described in detail in the next section). How much of the previous network is conserved on each iteration could be varied in different algorithms. Another, possibly more flexible system is shown in figure 19.

Figure 19, Another system.



In this case only connections evolve in the evolution space (although one could also allow single neurons to be placed in this space). The operation other than this is the same as the previous system described.

Naturally, the best aspects of the systems described in the previous section may be "mixed and matched" to provide a better result.

B.4.7 Other issues in Modular Evolution

i) Limitations of Evolutionary Techniques

There are several problems with Evolutionary Algorithms. Some are fairly obvious and others are subtle. One of the more subtle problems has to do with the inter-reaction between different parts of the system and is well illustrated by Timetabling.

Time Tabling is extremely important in academic and some other environments and basically involves assigning lecture classes, lecturers, rooms and timeslots so that classes run smoothly. Because the resources are limited it is not as trivial a problem as it might, at first, appear. Generally it is done by a human carefully placing the correct classes and teachers into the correct time slots, but this is tedious and time consuming. Genetic Algorithms have been devised to try and solve the Timetabling Problem.

The Genetic Algorithm is actually very good at *helping* with the problem. It can often nearly complete the timetable - however, a human can always do better and usually has to finish the task. The reason for this turns out to be important: a human can see

intuitively that placing certain time slots in a particular order early in the structure allows other slots to fit more easily later. Of course, given enough time, a GA would stumble across the same combination, but it is just one in a huge number. This is the advantage of design or intuition over evolution.

The same argument also applies to modular networks. Evolution has many different combinations to search and in any large network it may be a next to impossible task for it to find a particularly important combination of parameters which satisfy a particular requirement.

The other problem of modular network design is anything but subtle - what do the modules actually do? If the weights in a module have to be trained then how is the fitness function determined. This is analogous to training the hidden layer of a normal neural net.

In the following discussion, a possible answer to these two questions will be given in terms of incremental evolution. It should be remembered that, whatever the researcher's doubts about the difficulty of implementing this, the two tenants of evolutionary connectionism should be borne in mind:

1. Can a machine be intelligent to the point of consciousness?
Yes. The biological brain shows this to be true and it is a machine.

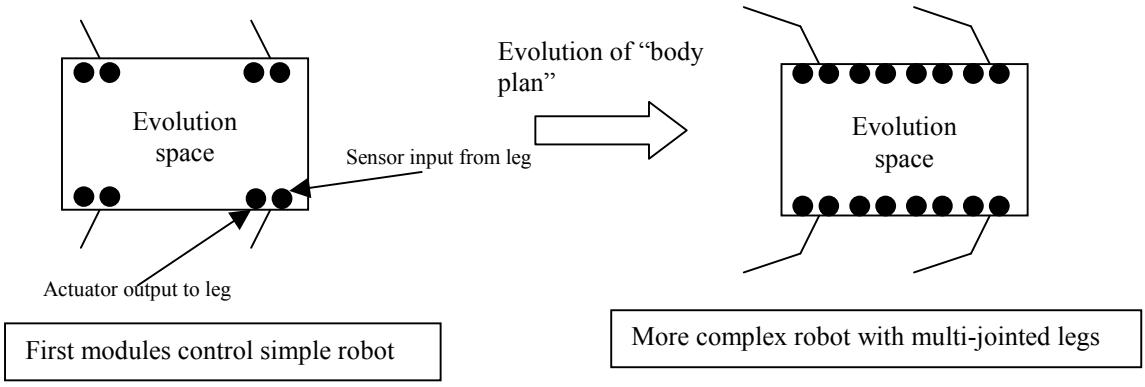
2. Is it possible to create such a machine?
Yes. Nature has designed such a machine through blind biological evolution and so therefore can artificial evolution.

ii) Possible solutions to these dilemmas.
The solution to the problems mentioned above lies in *Incremental Evolution* [1]. To understand this, consider the evolutionary development of the human brain. As the monkey brain developed into the ape brain and the ape into the human brain, did the whole structure re-wire itself each time? No, re-wiring tens of billions of neurons is simply not feasible. A much more reasonable explanation is that the brain *added* to its structure, building new modules on top of old. The scaffolding of the mammalian (and indeed the reptilian) nervous system is still present, buried deep within our own [24].

This is the process which can help us to evolve Modular Artificial Nervous Systems. Start with a few simple modules which allow the system to function at a primitive level in a constricted environment and build on these as the system develops.

Such an approach requires that sensor inputs and actuator outputs from the network must also evolve in complexity along with network - in effect evolving the body plan of the robot. For example, a legged robot might start of with simple, single degree of freedom legs, each with a single control input and a single sensory output as shown in figure 20a.

Fig 20, Robot evolution.



a) b)

Once the system can control the simple legs, extra degrees of freedom can be added and the network allowed to evolve (incrementally), figure 20b. The control system for a prosthetic might also proceed along similar lines, starting with gross movements and working down finally to digits. Likewise, a sensory system like vision would start perhaps with a single eye spot - only able to perceive light and dark and evolve complexity from there.

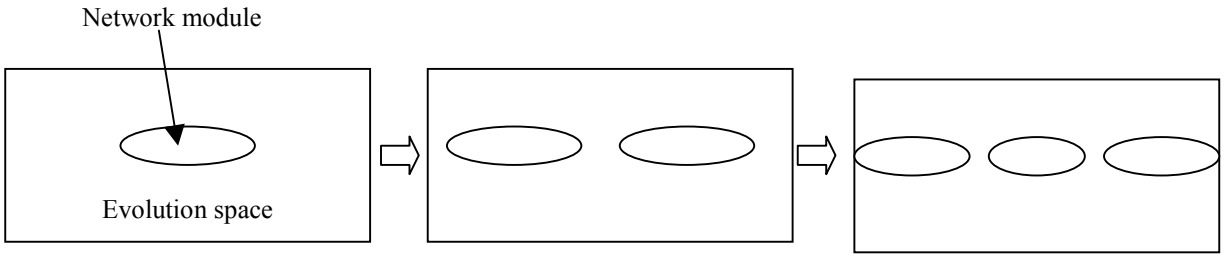
Obviously, any complex artificial organism would start life (as in both evolutionary and developmental biology) as a simple group of cells and develop.

One can perhaps also see some limitations in this structure. Although it is possible to see intuitively that it might work for robots, it probably would not work (or at least to work so well) for systems with a high degree of serialism in their design. One might postulate from this that there are some systems which nature would find it very difficult to design but which humankind would find easy. This is an aspect which has yet to be investigated.

iii) Operation of a practical algorithm

Let us consider how such a system might operate in a practical sense. This is best illustrated by an example. Suppose that we have a direct growth system as described in section B.4.6. As described above, the algorithm will start with a simple evolution space with few modules (one in this case), figure 21a.

Figure 21, Evolutionary algorithm operates on direct growth.



a) b) c)

This one module can be configured using two nested Evolutionary Algorithms. The first EA wires the module to the inputs and outputs of the system (the Global EA), the second wires the module internally (the local EA).

To illustrate this further: the global EA first selects an external connection topology, then the local EA configures the module's internal wiring. Once the internal wiring has reached its peak fitness this configuration is stored and the global EA runs onto its next set-up. In this way, the combination of best local and global connections can be found.

Having wired the simple system up, the algorithm next adds another module and more input/outputs to the outside world. The process is then repeated except that the previously wired modules are retained and the new modules wired on top (global connections to previous modules may change but local connections are retained), figures 21b and c.

The exact formation of the algorithm has yet to be developed. For example, it may be that operation is better if some changes to previously wired modules are allowed, particularly to modules which have been recently placed. This could be implemented using a simulated annealing algorithm running alongside the EAs. Multiple levels of modular networks (networks of networks) could be dealt with in the same way – using nested EAs.

Another important aspect of this type of system is the genetic operators which are used in the EA. Conventional EAs use recombination and mutation, but in nature large sections of DNA can be accidentally deleted, inserted, copied and inverted. This potentially allows chunks of network to be duplicated. Such duplication is a huge advantage. It could allow hard-to-evolve sections of network or whole modules to be reused. One can see the advantage of this given the problem described in time tabling above. In fact large parts of the human brain consists of millions of repeated modules, consisting of a few hundred neurons. These may be "learning units", configured to expedite learning – which may be heavily intertwined with topology as described in section B.3.2.

Conclusions

This paper has attempted to explain the lack of success of traditional Evolutionary Algorithms by outlining the differences between them and Biological Evolution. In particular the point that DNA codes self-organising chemicals which can produce repeating (modular) structures. It is pointed out that a artificial equivalent of this is difficult to implement.

As an alternative, several new methods of considering ANNs are suggested including several new methods of organising the ANN topology.

Finally some suggestions about the practical running of the algorithm are made.

References:

[1] The synthesis of Artificial Neural Networks using Single String Evolutionary Techniques, C MacLeod, PhD Thesis, The Robert Gordon University, 1999.
[2] D McMinn, PhD Thesis, The Robert Gordon University, 2001.
[3] A Framework for evolution of an animat nervous system, C MacLeod et al, EUREL Advanced Robotics Systems Development (conf), 1998. Paper 18.
[4] An evolutionary artificial nervous system for Animat locomotion, D McMinn et al, ICEANN -Engineering applications of neural networks (conf), 2000. p170-6.
[5] Combinations of Genetic Algorithms and Neural Networks: A survey of the state of the Art, J D Schaffer D Whitley L J Eshelman, COGANN -92 (conf), IEEE comp soc press, 1992.
[6] Molecular biology of the cell (3rd edition), B Alberts, D Bray et al, Garland Publishing Inc, 1994. p104 - 10.
[7] As reference 6, but p204-5, 404 -5, 573-4.
[8] As reference 6, but p104 - 10, 56 - 7, 111 - 35.
[9] As reference 6, but p24 -5.
[10] As reference 6, but p556 -60, also ch12.
[11] As reference 6, but p95 - 7, 111 - 35.
[12] As reference 6, but p556 - 560, also ch12.
[13] As reference 6, but p721 – 27, 1050 – 66, 1080 – 92, ch 21.
[14] Open University video – course s202, Biology, form and function course, video title : Patterns in developing gradients.
[15] Cellular Development, D R Garrod, Chapman-Hall, 1973.
[16] As reference 6, but p1119 – 30.
[17] As reference 1, but pA78 - A98.
[18] Investigation of an advanced neural network model, S I Monsen, MSc Thesis, The Robert Gordon University, 2000.
[19] Genetic synthesis of Boolean neural networks with a cell rewriting development process, F Gruau, *In* Combinations of Genetic Algorithms and Neural Networks, IEEE Computer Soc Press, 1992.
[20] A new approach to neural network topology, J Murray, MSc Thesis, The Robert Gordon University, 2000.
[21] Life's other secret, Ian Stewart, Penguin books, 1989.
[22] As reference 1, but p70-76, A3-A5.
[23] The blind watchmaker, R Dawkins, Penguin books, 1991.
[24] The brain: The last frontier, R M Restak, Warner Books, 1979. p 50 - 52.