To: TF
From: S.L.G.
Last revision: 8/5/95

I Am Ron’s Brain

A ‘how it works’ guide to the Albia brain model

One method to use when trying to invent something is to ‘skyhook’ it; in other words, flagrantly ignore any nigglingly impossible problems whilst searching for an otherwise ideal solution. Once you’ve found one, all that remains is to figure out how to solve the impossible bits, which, with hindsight, often turn out to be a lot more tractable than they seemed at first glance. This was the path I took when trying to find a workable design for the norns’ brains, and I think that describing how such a chain of reasoning led me to the present design is also a good way to explain to you how and why it works (and also to what extent it doesn’t).

The problem in hand was to design a mechanism for *taking decisions* based on sensory data, with resulting behaviour that most people would consider to be lifelike. To make things (relatively) easy, I refrained from tackling the problems of how to abstract information from raw sensory data and how to convert a given intention into a series of coordinated actions. These could safely be left to algorithmic solutions (at least to start with). All I was interested in was how a *perception* could be made to lead to an *intention*. To condense several days of mental ramblings into an approximately coherent tale, my train of thought could be described something like this:

Imagine a hypothetical creature who is ‘perfectly knowledgeable’; in other words, *one who knows the best action to take for every perceptibly distinct situation that might arise*. The brain of such a creature must be capable of carrying out two tasks: 1) it must be capable of *recognising* every perceptibly distinct situation, and 2) it must have a mechanism for *instituting the correct action* when that situation arises. In our skyhooking environment, it is quite easy to think of a neural system that can handle both requirements.

The first problem is to *recognise* the current situation, which effectively means converting a unique *combination* of sensory inputs (“I can see a Grendel” AND “it is moving towards me” AND “it looks angry” AND “I’m unfortunately tied to a tree at the moment”) into a single unique identifier that represents the total situation — presumably a unique neurone, that should fire whenever that situation arises.

Given such a system, the second problem, that of deciding how to react to each situation, is trivial: all we need to do is connect this firing neurone, which I’ll call a *concept neurone* (a misnomer, but one I’ve lived with too long to abandon), to the *motor neurone* that institutes the action most appropriate to that situation.

One could visualise, therefore, a sheet of concept neurones, wired up in various combinations to fibres from the sensory organs, which enter the sheet at intervals from underneath. A mass of fibres extends upwards from the sheet, one fibre leading from each concept neurone to the most appropriate motor neurone. A simple design, but in a skyhook world, one which would work perfectly well. Whenever signals arrive from the sense organs, a single concept neurone representing the current combination of inputs would start to fire. This would direct a signal to the correct motor neurone, which in turn would cause the appropriate response from the creature.

However, so much for skyhooks; now comes the tricky bit! There are at least three embarrassingly large flaws in this design:-

1) Allocating one concept neurone to every perceptible situation would require a ludicrously large number of neurones. Even a human brain wouldn’t be large enough to do the job.

2) Our perfect creature is assumed to ‘know’ in advance which is the most appropriate action to take in every possible circumstance, yet in practice our creatures start off knowing nothing at all, and have to learn everything for themselves. The connections between concept and motor neurones must become established as a result of experience, since they are not wired up in advance by some well-meaning deity.

3) Not only must a practical system be capable of learning the most appropriate response to each situation, it must do it by more than mere trial and error. When it encounters a never-before-experienced situation, it must respond to it *intelligently*, in other words it must be able to use past experiences as a guide for how to react to novel situations. This is a rather more subtle problem than those above, but its solution is critical if our creatures are to respond in *lifelike* ways.

Let’s examine each of these niggling difficulties in turn, and consider possible solutions:-

1) The size of concept space.

The mental image of our skyhooked concept neurones arranged as a two dimensional sheet brings to mind the mathematical notion of *phase space*. Just as points on a two dimensional surface can uniquely represent every possible combination of two values (x and y, as in a graph), so a discrete, two-dimensional array of concept neurones could be wired up to represent each possible combination of signal strengths from two sensory neurones. If our neurones’ output signal strengths were represented, say, by the integers from 0 to 127, then 1282 neurones would be capable of representing all possible combinations of two sensory inputs. In practice, our creatures need a lot more than two sensory inputs, so we need correspondingly more dimensions of phase space to represent all possible combinations. For 80 senses (a reasonable figure) we would need 12880 neurones, or roughly 3x10168, which is 3x10158 times as big as the human brain. Embarrassing or what!?

OK, we can do better than that. We don’t *need* to distinguish between the different *strengths* of each sensory signal. A sensor that detects movement towards the observer might well fire more strongly if the object is moving more rapidly, and certainly an intelligent creature might want to distinguish between things which are approaching slowly and things which are approaching rapidly. However, 128 degrees of distinction is carrying things a bit far, and anyway, many senses will be simple on/off inputs. Let’s suppose then that we can get away with one concept neurone for each *logical* combination of sensory signals, i.e. *whether* one or more sensory inputs is firing, rather than how much they are firing (in practice, we can make use of the *strength* of each sensory input in other ways). For that we’ll need a mere 280 concept neurones for 80 sensory inputs. This is roughly 1024 — zillions of times smaller than before, but still the size of a hundred million million human brains, and just a touch on the big side for a PC to handle!

Clearly we can’t possibly provide the one-neurone-per-perceptible-sensory-situation that our skyhook model requires, so what are we to do? Happily, the first part of the answer is quite easy: It may well be *possible* for a creature to encounter 1024 perceptibly different situations, but any one creature is actually only going to experience a minute fraction of those possible circumstances given a reasonably short lifetime in which to do it. We don’t know which situations a given creature will experience from that vast set, and different creatures will experience different ones, but we do know that there will be a (relatively) small number of them in each case. What we need, therefore, is a system that *wires itself up to represent each new situation as it is experienced*. At worst, such a system would only need enough concept neurones to represent the maximum number of experiences that a single lifetime is likely to encompass; at best it needs only enough to represent all the situations that *turn out to have meaning* — just as we all forget trivial events and remember only those things that turned out to have an impact on us.

That’s the first impossible problem nearly solved, then; all that remains is to devise a *mechanism* that is capable of such behaviour. Unfortunately, this wasn’t as easy as I’d have liked! It sounds like a task for an elegant, self-organising dynamical system, as beloved of A-Lifers, but so far, I’ve been unable to come up with one that solves all the problems and I’ve had to resort to some nasty top-down methods. I’ll leave explaining what mechanisms I would have liked to use, and which one I’ve actually used, until the discussion of Impossible Problem Number 3, below.

2) Making it learn.

Our hypothetical, ‘perfect’ creature already knew what to do in every circumstance, and therefore had a single, hard-wired connection from each concept neurone to the most appropriate motor neurone. The real creatures start off more or less ignorant, however, and must learn the most appropriate responses for themselves (just as well really, otherwise they wouldn’t be much fun to play with). In principle, all that is required is the following simple mechanism:-

I. When you find yourself in a novel situation (one which you haven’t experienced before and therefore don’t know how to deal with), pick an action to take (for now, let’s say one gets selected at random).

II. Form a synaptic connection between the concept neurone that’s firing and the motor neurone for the action that you’ve decided to take, and assign it a signed *weighting*, perhaps set to zero initially.

III. If the action you took turns out to be a rewarding experience, then increase the weighting for that synapse; if it turns out to have been a bad move, then decrease the weighting.

IV. When that situation arises again in the future, the likelihood that you will take the same action depends upon the value of the weighting - if it is negative, then the activity level of that motor neurone will be suppressed and the action is less likely to be taken; if positive, then an excitory signal arrives at the motor neurone, and so the action is even more likely to occur this time.

A successful first choice of action will just cause the relationship to become more and more reinforced, and the creature will never try anything different when confronted with the same situation (unless some other factors come into play to create a degree of fickleness or curiousness). On the other hand (as is statistically more likely), if the first choice of action turns out to have been a bad move, then it will be suppressed next time, and so a different action will be taken. After a while, a whole bunch of connections will arise between a single concept neurone and several motor neurones. In isolation, the creature would simply keep trying new actions until he found one that was successful, and would thereafter stick with it. In practice, however, a number of other factors come into play, and the various interconnections between concept and motor neurones comprise a delightfully subtle and complex ‘voting system’, where many signals may combine to produce a set of ‘recommendations’ about how desirable each of the available courses of action might be. Because of the dynamical behaviour of the neurones I’ve used, these ‘recommendations’ persist over time to a diminishing degree, so that the current ‘best decision’ is based, not only on the situation pertaining at the time, but also on those that preceeded it.

Such a mechanism is obvious and fairly straightforward in principle, but there are a number of messy details that we have to deal with, for example:

a) Reward and punishment often occur well after the actions which logically led to them, and when reinforcement does arrive, it might not actually have been caused by the most recent action, or indeed any of the recent actions, so how do we relate a reinforcement to the correct decision or decisions?

In the present design, I solve this to an acceptable degree by making concept-motor synapses become *susceptible* to reinforcement whenever a signal through that synapse causes an action to be taken, and allowing that susceptibility to decay exponentially over time, with a half life of a few seconds. Consequently, when a punishment or reward arrives, the action most recently taken becomes most strongly reinforced, while preceding actions are less so. Spurious reinforcements (random or unrelated pain or pleasure) will become cancelled out after a number of tries. This method is unsatisfactory, however, on two counts: firstly, the reinforcement must arrive within a fairly short time after the causative action is finished (one of the two reasons that norns will never choose to take out pension plans!); secondly, the wrong actions may get reinforced — it wasn’t scratching your nose with the dynamite that caused the pain, it was lighting the fuse beforehand that did it. I think this is a fundamental flaw, due to the *passive, behaviourist* nature of the model I’m using. I am actively thinking about a new design in which reinforcement is applied only to the situation pertaining at that moment, rather than being propagated back through previous actions, and the process of making a decision is much more active and predictive, looking forward at possible consequences, rather than back at learned history. The way that biological systems tackle this problem is clearly very sophisticated (see Steven Rose’s chickens) and has much to do with assessments of novelty and logical connectedness. However, decaying susceptibility will do for now.

b) What constitutes reinforcement in the first place?

Pain is an obvious form of reinforcement, but there are others, some of them rather subtle, such as disappointment. I decided to assess the level of reinforcement provided by external events according to the sum total of their effect on a set of *drives and needs.* For example: eating food reduces the hunger drive, and is therefore a Good Thing, unless that is, you have eaten too much, when it increases your level of pain, which is a Bad Thing. Events which increase the level of drives are taken to be punishing reinforcers, while those that reduce a drive are rewarding. Drives include such things as “the need to avoid pain”, “the need to keep warm”, “the need for the company of others”, etc. The current level of each drive is ‘stored’ by the activity level of a single Drive Neurone (see ‘neurone dynamics’, below), and the changes in state of those neurones generate reward and punishment ‘hormones’, which flood the brain and alter the weights of susceptible synapses. In real environments, the laws of physics and chemistry directly generate the physiological changes that indirectly cause drive level changes. However, in a virtual world, there is no complex, self-consistent chemistry to do the work; objects are really only imitations of the real thing, and cannot cause physiological changes merely as a result of their physical properties. Instead, I have to explicitly decide what effect any object’s actions may have on a creature’s drives, for example I have to *state* that eating food makes one less hungry and also uses a small amount of energy, thus making one more tired. In the virtual world, therefore, almost every *event* which takes place generates one or more *stimuli*, which may be tactile, audible or visual, and therefore will be received by creatures who are in contact with, in earshot of or can see the event taking place. Each stimulus, amongst other things, is defined as having a given level of effect on one or more of a creature’s drives, and this is the mechanism by which events in the outside world (which may have been triggered by a creature’s actions) generate the rewards and punishments that lead to learning.

c) If I touch something and it’s hot, the pain must cause sufficient reinforcement to make sure that I don’t immediately try the same action again, regardless of how much other considerations are encouraging me to do so. On the other hand, if the first time I kiss a girl she slaps me, and that reinforcement is too strongly applied to my memory, then I am most unlikely to try kissing anyone ever again, and will not discover that the punishment came from the *circumstances* under which I kissed her, rather than the kiss itself. How do we deal with these conflicting needs?

To solve this problem, I made the synaptic weighting a little more sophisticated. Instead of it being a simple value that only gets altered when reinforcement arrives, I gave it a tendency to *relax* fairly rapidly towards a rest state, while the value of that rest state itself has a tendency to relax, albeit much more slowly, in the direction of the current weight. Therefore, when reinforcement arrives, it perturbs the synaptic weight fairly heavily, causing a strong effect on future decisions in the short term. The weighting then relaxes back towards its rest state and the memory starts to fade. However, whilst the weight was displaced from its rest state, the rest state value was also relaxing slowly towards the displaced weight. Thus, when the weight has relaxed completely back to rest, it is now resting at a slightly different value than before. The net result is that reinforcements cause a strong initial learning response, which fairly quickly becomes almost ‘forgotten’, yet repeated reinforcement causes a longer term ‘memory’ of the rewards or punishments that have accumulated over several episodes. This short-term / long-term ‘memory’ seems to have quite a strongly beneficial effect on the dynamics of the system.

d) Real neurones can produce as many dendrites and as many synapses as they need, given space and enough nutritional resources. Inside the computer, however, interconnections demand memory and processor power, and are thus at a premium. Since the majority of links between concepts and actions turn out to have no predictive value (no reinforcement arrives, and so the weighting remains zero), it is wise to ‘prune out’ such valueless associations periodically. To this end, each synapse that forms is given a *strength* value, which is heightened whenever reinforcement arrives, but decays slowly between whiles. Any synapses whose strengths have reached zero, get removed and ‘recycled’. Even a reinforced relationship may turn out to have been a fluke, if it does not get reinforced again in the future. Thus, after a previously reinforced synapse’s strength has become zero, its weighting value also starts to atrophy. A synapse is recycled once both its strength and weight have reached zero, thus allowing creatures to ‘forget’ associations that don’t get sufficient repetition.

3/ Intelligent learning

When a never-before-experienced situation arises, a concept neurone representing that situation will fire, yet will not send signals to any motor neurones, since no associations have yet been learned, and no weighted connections exist. In the above description, it was assumed that an action would thus be taken *at random*, and any feedback from the environment would enhance or inhibit such a decision in future. In an un-formed brain, random factors might be the only way in which an action can be taken in response to a novel situation, but in general this will just not do. Real living systems, when faced with a novel situation, will not just pick an action at random, since most of the time they will turn out to have been wrong (consider how many courses of action are available to a real organism: “Um, I’ve never seen a rattlesnake before. I know, I’ll stand on one leg and whistle ‘Dixie’ and see if that helps”). What real creatures do, and what our norns must do if they are to behave intelligently, is to *decide how to react to a novel situation based on information gleaned in previous, more or less similar situations*. When confronted with a cliff edge, the memory of how much it hurt to step off a tall stool will prove invaluable in telling you how to react. Without such an ability to generalise from past experience, the only way to deal with a cliff edge is to guess at an appropriate response, and if you get it wrong, the lesson might well be learned too late!

This ability to generalise is so very important if norns are to learn and behave in lifelike ways, and the nearest thing I have had to a ‘breakthrough moment’ in this whole project was when I realised how this behaviour might be achieved in a system such as I’ve described. We have decided that concept space must wire itself up (in some as yet undefined way) as new experiences arise. Suppose there was a tendency for concept neurones to allocate themselves in such a way that *neurones which are near to each other tend to relate to similar situations*. The nearer two concept neurones are to each other, the more similar are the situations which will cause them to fire. Further, suppose that when a concept neurone fires, it radiates a signal of some kind that encourages nearby neurones to fire also, in inverse proportion to their distance (lateral excitation).

Now, imagine that Ron the baby norn has learned from past experience (maybe by a random choice of action) that stepping forward when on a tall stool is not a good idea, yet, until now, has never experienced standing on the edge of a cliff. The sensory signals feeding the new “on edge of cliff” concept neurone will cause it to fire, yet if it is connected to any motor neurones at all, it is by zero-weighted synapses, and so the signal fails either to suppress or to recommend any choice of action. Nearby, however, is the “on top of stool” neurone. This is nearby because standing on a stool shares some sensory similarity to standing at a cliff edge. Because the ‘stool’ neurone receives lateral excitation from the firing ‘cliff’ neurone, it will tend to fire too, albeit more weakly than the ‘cliff’ cell. This weak signal will have a tendency (through the previously learned association) to suppress the firing of the ‘step forward’ motor neurone, and thus Ron is made less likely to step forward. In fact, all the ‘cliff’ neurone’s neighbours will be firing to different degrees, and maybe others will be able to contribute to the total signals arriving at one or more motor neurones. The sum total of all these signals will contribute to a decision about what to do, or at least, what not to do (assume that all motor neurones tend to fire moderately all the time, and that excitory signals tend to increase a motor neurone’s activity, while inibitory ones tend to decrease it; if any motor cell exceeds a certain threshold of activity, then that action will be taken, otherwise the creature will continue with the last course of action decided).

All that remains now is for the ‘susceptibility to reinforcement’ of each concept-motor synapse to be adjusted in proportion to its strength of firing. That way, if Ron still decides (as perhaps strongly recommended by the nearby ‘standing on hot coals’ neurone) to step forward off the cliff, and he survives to learn the lesson, then the ‘cliff-step’ synapse will be most strongly reinforced by the resulting pain, while ‘coals-step’ and ‘stool-step’ connections receive less modification. In this instance, the ‘cliff-step’ relationship has been *learned* and the ‘stool-step’ relationship reinforced, while the ‘coals-step’ relationship has been weakened slightly. Some of this may seem undesirable at first sight, but the net result is more-or-less exactly what is required. The very first thing a newborn creature learns (maybe at random — perhaps one motor neurone was firing slightly more than the others) is his sole guide to how he should react to the next situation in which he finds himself. Thereafter, he develops more and more sophisticated, and more and more refined judgement as new confirming or surprising relationships become learned. In short, his predictive abilities increase with every new experience, and the ‘intelligence’ of his responses increases also.

Brilliant! Just what we need. So, how do we achieve it: what rules can be applied to neural growth to ensure that the ‘locality principle’ (similar concepts grow near to each other) naturally arises?

Well, the truth is, I don’t know. I have a lot of faith in the theory, and in my mind’s eye can *almost* visualise several CA-style or biologically believable mechanisms that achieve it. Even the human brain shows just enough similarities to encourage belief in the validity of such a scheme (topological sensory and motor maps; ‘place’ fields and other topologically organised perceptual relations in the Hippocampus). However, when I thought about it, I hit upon a snag that I’ve been unable to solve, and so I have been forced to adopt a more conservative, more easily analysed solution that, though strongly related, doesn’t have the elegance of the self-organising design I was hoping for. Once this project is over (or if inspiration strikes) then I’ll return to this topic. However, for the moment, here is the problem and the less-than-elegant solution I’ve adopted:

Imagine concept space as a sheet of initially unallocated (unconnected) neurones, into which, at intervals across its surface, fibres from the senses feed into a small number of those neurones’ inputs. These sensory fibres might be geographically distributed according to the type of feature they represent, in the manner of the topological maps of the body’s sensory surface in the human cortex. Each concept neurone that is attached to a sensory fibre clearly represents a single-sensory input situation, such as ‘I am moving forward’ or ‘I have just bumped into something’, while the remaining unallocated neurones will become wired up to the single-sense neurones in order to represent compound-sensory situations as they arise, such as ‘I am moving forward AND have bumped into something’. (Before Seymour Papert appears round a corner shouting about how my single layer net can’t represent all the boolean relationships, let me say that I know and I don’t care: concept combinations can represent AND relationships and OR relationships well, whilst NOT is only partially supported (there are no sensory inputs representing the *abscence* of a feature), but in practice this causes no serious problems.)

So, imagine a concept cell representing sensory situation A, and another representing B. When situation A+B arises for the first time, we expect one of the currently unallocated concept neurones to come to represent it. Which one will it be? Well, the logical choice is the nearest available cell to the mid-point of A—B, and it is fairly easy to design biologically or computationally feasible rules that can cause this behaviour to emerge spontaneously. Over time, you can imagine cells competing with each other to represent each situation that arises, and the whole of concept space ‘filling up’ with points representing each combination of sensory inputs in a ‘best-fit’ manner.

However, imagine that A and B are at diagonally opposite corners of a square made up of A, B, C and D cells. A and B have both fired simultaneously in the past, so a cell AB will have arisen near the centre of the square. Now C and D fire. A new concept CD is required, and it will also try to arise near the centre, which is already filled by AB. CD will thus emerge at the nearest unallocated point to the centre, and all looks well. That is, all will be well until the system tries to decide how to react to this novel situation, CD. CD has no associations, and so it relies on recommendations from nearby cells, one of which is AB. Now, if AB’s experiences turn out not to be much of a guide for situation CD, then the creature will soon learn to distinguish between the two. Unfortunately, although no actual harm will be done by using AB as a guide to CD, there is little, if any, logical relationship between these two situations — they are close together, but perceptually unrelated. Thus, AB’s recommendation is no better than a purely random guess, when it comes to generalising about CD, thus showing none of the intelligent predictive behaviour we were hoping for.

When you magnify the problem by imagining 80-odd sensory inputs, it soon becomes obvious that a whole host of ‘common centres’ exist between groups of inputs, making such arbitrary decision-making far too common for comfort. Additionally, we have a problem with situation AZ, where A and Z are at opposite ends of concept space — if AZ is in the middle, then it is too far away to receive lateral excitation from A or Z, and so again no generalisation occurs. I have not lost hope that I might find some scheme by which concept space can organise itself without causing these infringements of the locality rule. I have considered fractals, folding, uneven lateral excitation and other schemes, but so far have not been able to design a system that I could guarantee would do the job.

So, I had to start thinking about methods that didn’t rely on geography to provide the generalising influence. It quickly became obvious that even without the locality rule there already existed a generalising factor that I hadn’t at first noticed. I thought about what ‘similarity’ actually *means* in the context of sensory perception, and realised that whilst it need not always be true, two situations are generally held to be similar because *they share sensory features*. The more features they share, the more similar they are: ABCD is more related to ABYZ than it is to AXYZ. Suppose, then, that when inputs A and B fire simultaneously, not only does the AB neurone fire, but so too do the A and B neurones (though more weakly). A and B are therefore able to generalise on behalf of AB, because the sensory signals themselves are making them fire, not because geography-based lateral excitation is doing it. As long as all the *sub-concepts* fire, as well as the *super-concept*, and as long as super-concepts fire more strongly than sub-concepts, then both the generalisation and the differential reward mechanisms work perfectly well. Indeed, since the creature has no way of knowing, when situation ABC arises, whether it is that exact juxtaposition of A, B and C that is significant, or whether it is only the AB relationship, or AC, or A, or whatever, then it is important that all sub-concepts take part in the reinforcement process, as well as the decision-making one.

There are several plausible bottom-up rules that might result in a network that wires itself up to fire cells representing all permutations of any combined set of sensory inputs. Perhaps a mixed population of cells could exist, some having only one dendrite, some two, etc., each cell trying furiously to grow dendrites towards sources of signal. Upon reflection, this might be a good technique, as it prevents the system from building up huge arrays of neurones just to represent complex and irrelevant superpositions of sensory states — the state ABDGHLMPQRTUVX might occur occasionally, but it is most unlikely to require a unique reaction; it is much more likely that a sub-concept such as ABHL is the significant factor that should be noted. Anyway, for various reasons, including the need for rapid computation, I currently use a different scheme, as follows:

Each unallocated concept neurone has two, and only two, dendrites. Its behaviour is to grow towards any source of signal, whether coming from a primary concept neurone (A, etc.) or from a super-concept (AB, etc.). The system thus develops into a mesh of binary trees: the situation ABC might be represented by a neurone that receives signals from the A neurone plus another neurone fed from B and C (this setup can be described as A+BC). It is fairly easy to devise a mechanism that knows when neurones need to grow, and when they can stop growing, and the binary arrangement makes various other computational processes easier. However, the system is inelegant and unsatisfactory in several ways, for example, it suffers from a temporal sensitivity (A+BC is different from AB+C, yet they represent the same situation, differing only by the order in which the sensory signals arrive) that could, in theory, be useful, but in practice serves only to waste neurones and weaken the generalising ability.

In practice, there are even more itty-bitty complications with all the possible models than I’ve described here. The present mechanism sort-of works, and I may have to wait until the next project before I have enough research time to develop better solutions.

The rest of it

The above is a description of the bulk of the brain model: how sensory inputs map to concept neurones, which, in turn, feed motor neurones; how reinforcement in the form of drive changes is fed back to the system to cause learning; and how the wiring of concept neurones provides a mechanism for generalisation and intelligent prediction. There remain, however, a few odds and ends to be described:

1/ Neurones and dendrites

The ‘brain’ is made up of a number of different regions, divided geographically, such as concept space, the decision layer (motor neurones), drives and needs and attention-directing neurones (see below). Each region has different properties, yet it was important to me that I try to derive these various functions from a single, general-purpose neurone type. The different properties arise because each region’s neurones have different values in one or more of their ‘governing constants’.

Each neurone is essentially a ‘storage element’, whose *output* signal strength is a reflection of the cell’s *internal state*, whenever that state exceeds the cell’s current *threshold* value. The internal state is derived from input signals arriving via fibres called dendrites (see below). However, the internal state is not simply the sum of the input signals. The state value has the property that it tends to *relax* towards a given *rest state*, at an exponentially decreasing rate governed by the *relaxation rate* constant. The internal state is therefore like chewing gum: input signals stretch it away from equilibrium, to which it tries to return, at first rapidly, and then progressively more slowly.

By adjusting the threshold, rest state and relaxation rate of a cell, different properties may be derived. For example, a cell with a zero relaxation rate sums its inputs over infinite time, while one with a very fast relaxation rate reflects only the current state of its inputs. Altering the rest state is also useful: motor neurones have mid-range rest states and moderate relaxation rates. This gives them a tendency to fire moderately all the time, unless enhanced by excitory inputs or suppressed by inhibitory ones, while their relaxation rate allows them a tendency to sum their inputs over time, thus building up an integrated reaction to the tugs and nudges generated by changes in concept space.

One important property provided by this relaxation process is a *damping effect*: firstly, every cell (with a non-zero relaxation rate) has a tendency to return safely to rest when left alone; secondly, as input signals accumulate to push a cell’s state towards the maximum value, so the relaxation process rises up its exponential curve and provides increasing resistance to the input — cells are easy to nudge up to low values, but it is very difficult to force a cell fully ‘on’. This controllable damping effect was from the beginning seen as a desirable property, since experimental complex dynamical systems have an unfortunate tendency to develop positive feedback and ‘lock-in’ to an irreversable state or to thrash wildly and chaotically. The damping from relaxation was one highly controllable mechanism that allows me the liberty to ‘turn up the heat’ and drive the various cell assemblies away from equilibrium towards the ‘lively’ dynamics characteristic of living systems without fear that I would lose control too easily.

Neurones are also potentially capable of exhibiting habituation and sensitisation, though at present I have insufficient need of these features to justify the processing load.

‘Dendrites’ are the devices that conduct signals from one neurone to another. Each dendrite has, as already mentioned a relaxation-enabled signed *weighting* (STW — short term weight) and a relaxation enabled *rest state* (LTW — long term weight), plus a *strength* value and a *reinforcement-susceptibility* value. Dendrites are treated as separate entities, that may be attached in arbitrary quantities to neurones, thus minimising storage demands and removing the need for different classes of neurones. Dendrites are, however, a limited resource, and so various recycling systems are needed.

The brain model currently consumes 1024 neurones and 2500 dendrites.

2/ Attention

Even in a simple virtual world, there is a lot of stuff going on within a creature’s sensory reach, quite apart from the internal, physiological sensory data. There was a serious risk that too much sensory input would cause an explosion of complexity in the neural networks, leading to a rapid overload of resources. For this reason and others, creatures have their sensory streams filtered through an *attention directing system*. This is a simple array of neurones, each one of which represents one of the objects of which the creature is currently aware. Every stimulus generated by an object causes a nudge of an appropriate magnitude to that object’s assigned attention neurone. The neurones have moderate relaxation rates and rest states near zero; they therefore accumulate input nudges over a short period of time, and thus represent *how significant each object currently is*. The outputs of the array of neurones compete, and the one firing most strongly determines to which object the creature pays attention. This object is the only one whose sensory details (what sort of object is it? Is it moving?) are fed to the brain. It is also the object to which any action decisions get applied. For example, if the creature is aware of a clock, and that clock chimes, then a nudge is applied to a certan attention neurone. If the output of that neurone starts to fire the strongest, then the creature shifts attention to the clock. Sensory data about the clock arrive in the brain and the creature may choose to take action. If so, then that action is directed towards the clock. One of the attention neurones is devoted to the creature himself. When that is the strongest firing (caused by internal stimuli, for example) then the creature pays attention to himself, i.e. is ‘introspective’.

3/ Action schema

To minimise the complications and make this project sustainable, I balked from the task of using neural systems to coordinate the fine details of motor activity. Instead, this is handled by algorithms (though these have been designed in such a way as to be amenable to neural or even genetic control later). The brain, therefore, merely has to make a decision about a *general course of action*. The available repertoire of actions was kept to a small number, for obvious reasons, and are quite generalised in nature: their detailed implementation often depends on the type of object to which they are directed, or to the mental and emotional state of the creature at the time. Actions include such things as: attempt to activate it; attempt to approach it; move away from it; rest; be quiescent; pick it up...

4/ Language

The most contrived system in the program is the language system. Creatures speak occasionally at random about what they are doing, as they have a small vocabulary of words relating to the names of objects and verbs representing the various action schema. These words are initially in ‘baby talk’, but the user can ‘teach’ them new words for any object or action by a mechanism connected with the attention director. The user can also issue commands. This works as follows: When the user ‘speaks’ the sentence is broken into words, and the vocabulary is scanned for recognised words. Verbs cause direct stimulation of the appropriate motor neurone, thus increasing the likelihood that the creature will take that action; nouns are sent to the attention director neurones, thus tending to draw the creature’s attention to that object. A few additional twiddly bits provides a system that at least gives the impression that you are able to teach and converse with your pet creatures (creatures can also influence each other in the same way).

I think that’s about it for the main brain mechanisms. The brain is only a small part of the overall system. Next job is to describe the rest of it! Watch this space.