Hopfield Networks and the Hippocampus

Hippocampal CA3 neurons are very extensively interconnected by recurrent excitatory synapses, as well as strong inhibition. In the linear associator we studied how patterned activity (represented by the input vector **f**) of a set of input neurons would influence the patterned activity (output vector **g**) of a set of output neurons, in the one step computation represented by the equation **g = Wf.** But in a recurrent network the output activity is not just *applied* to the input for further rounds of computation, but actually *is* the input activity. In other words, activity could continue to swirl through the network indefinitely, never settling down to a fixed pattern; the changes could become repetitive, so the network cycles, or chaotic; or the changes could eventually (perhaps quite rapidly) stabilize, perhaps with all neurons firing the same way, or perhaps according to some complex, yet unchanging, pattern. If the neurons' firing rate is simply linearly related to the summed synaptic input, as we assumed in the Linear Associator, it could easily happen that the activity of some neurons could grow indefinitely without limit. Clearly this is biologically impossible, so at the very least we would have to assume that their firing rate was NOT a linear function of the summed synaptic input. Hopfield made a number of assumptions that allow a simple analysis of the behavior of a recurrent network. These assumptions are not very realistic biologically, but the resulting behavior is qualitatively similar to that of much more realistic networks, and much easier to understand. The crucial outcome of his analysis is that these simplified recurrent networks tend to stabilize to a limited set of patterns, and that these patterns can be made to coincide with memories if synaptic weights are set using the Hebb rule. Furthermore, if the network is started in a pattern that resembles one of the memorized patterns, its activity will usually quickly stabilize to that pattern.

Hopfield simplified the rate assumption even further, by assuming that a neuron i can exist in only 2 states, firing or not firing. These 2 states are represented by the numbers $f_i = 1$ or $f_i = -1$ (the math is simpler this way than using 1 and 0). The state of a neuron is determined by the weighted sum of the synaptic inputs to that cell. He assumed that a cell i cannot make a synapse onto itself, and that if the net input (from all the other cells, labeled j, weighted by the strength of their connection, $w_{i,j}$) is positive the cell fires, while if it is negative, it doesn't:

$f_i = 1$ if $\Sigma\, w_{i,j}\, f_j > 1$ or $f_i = -1$ if $\Sigma\, w_{i,j}\, f_j < 1$ ……………..Eq 1

This assumption is known as the "sign function" – it is completely different from the "linearity function" we used before. However, both of these contrasting assumptions do capture some aspect of the biology: neurons can fire at a range of frequencies, but they do have all-or-none behavior; in a sense Hopfield took a very short term view of neural behavior, while the Linear Associator took a very long term view.

Hopfield made one further simplifying assumption: he assumed that reciprocal weights are equal i.e. $w_{i,j} = w_{j,i}$

This assumption is probably not quite correct for biological recurrent networks, but reciprocal pairs of neurons do tend to have similar reciprocal synaptic strengths. We will see later how this assumption flows from the Hebb rule.

We will now define a quantity called "harmony", represented by the symbol h (for any pair of neurons) or H (for all possible pairs). The harmony of a pair of neurons i and j measures how well these 2 neurons' activities agree with each other (their neural "compatability") , taking into account the nature of their synaptic interaction:

$$h_{i,j} = w_{ij} f_i f_j$$

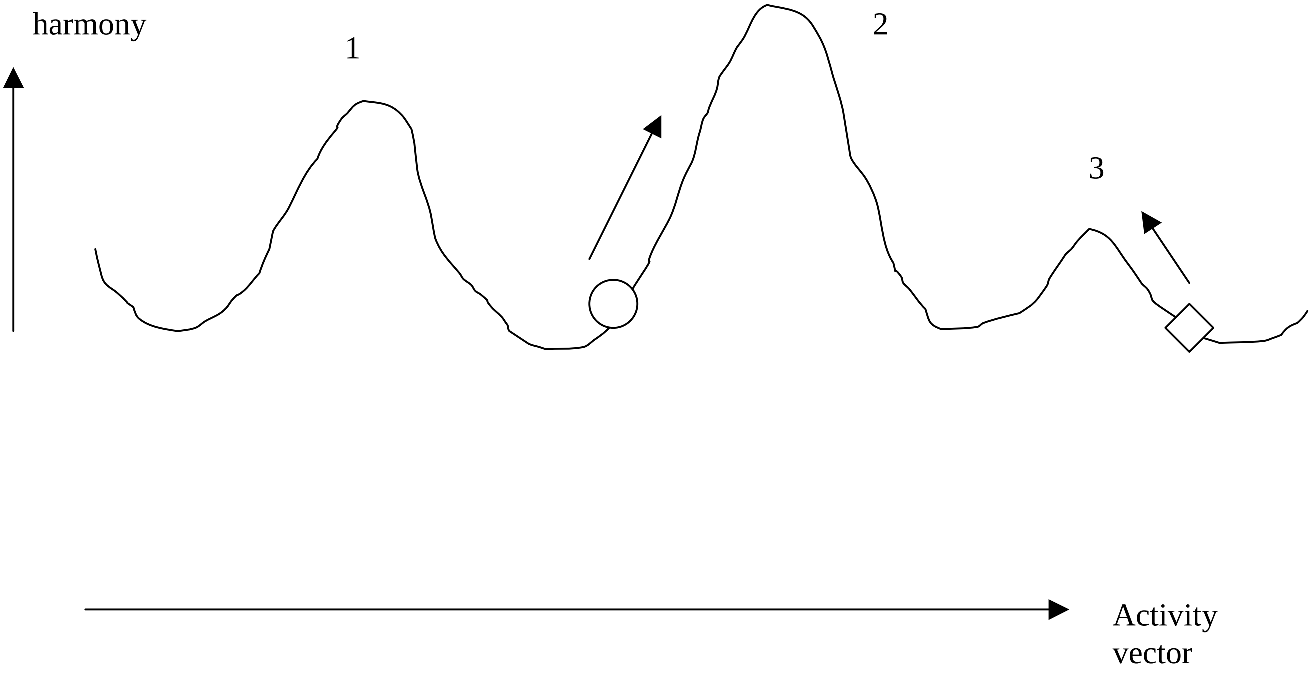
Thus if the connection between them is inhibitory harmony demands that the neurons have opposite sign, but if the connection is excitatory, they should have the same sign. Note harmony can be negative (i.e. disharmony). The harmony of the whole network is just the sum of all the individual pairwise harmonies.

In the Hopfield network it is assumed that the network changes due to isolated changes in the state of individual neurons, rather than simultaneous changes in the state of many neurons. This is biologically plausible, because there is no master clock (on the millisecond timescale) in the brain. Formally, we pick a neuron at random, calculate the sign of the net input, and (according to Eq 1) either (A) change the neuron's state (if the sign of the net input disagrees with the current state) or (B) leave the neuron unchanged (if the sign of the input agrees with the current state). This procedure is called "updating" the neuron.

What will be the change in harmony due to updating? The change in total harmony due to updating the ith neuron, ΔH, is only due to case A, i.e. possible changes in the state of the ith neuron, and is given by

$$\Delta H = H_{new} - H_{old} = (f_{i,\,new} - f_{i,\,old}) \left(\sum_j w_{ij} f_j \right)$$

where we took the second bracketed term out because it does not change during updating. The sum runs over all the other neurons j not equal to i. In case A, the update could be 1 to –1 or –1 to 1. In the first case the first term would be negative (-1 –(+1)) but to have flipped from 1 to –1 the neuron must have received a net negative input $\Sigma\, w_{ij} f_j$. Since both bracketed terms are negative, harmony must increase. Likewise, if the neuron went from –1 to +1, both terms must be positive, and again harmony increased. Since either a neuron flips during updating, increasing harmony, or a neuron doesn't, leaving harmony unaltered, total harmony must increase or remain constant. We have thus proved that the network moves to a state of high harmony, consistent of course with the weight matrix **W**. When it reaches a highly harmonious state, no further changes in activity are possible – the network stabilizes, with some of the neurons remaining "on" and the rest staying "off". We have not however proved that there is only one state of maximum harmony, merely that the network moves to a stable state of high harmony – a harmony peak. Depending on the structure of **W**, there could be many highly harmonious states, and if the network is started in an inharmonious state, the Hopfield rule guarantees (together with the assumptions about the weights) the network will move towards the harmony peak which is most similar to the initial state (similarity being defined in the way we used previously – the cosine of the angle between the initial activity vector and the activity vector corresponding to the harmony peak. This idea of the Hopfield network climbing harmony peaks can be visualized in the following diagram.



Climbing harmony peaks. The total network harmony is plotted on the ordinate, and the state of the network on the abscissa. A possible initial state of the network is shown as a circle. This network state moves to local harmony peak 2 as a consequence of Eq 1. If the network starts in the state represented as a diamond, it will move to harmony peak 3. This occurs because the Hopfield rule Eq 1 either flips neurons to increase harmony, or leaves them unchanged. Note that the representation of activity states on a single axis is diagrammatic only. In reality, the states lie in a high dimensional space, in which state 1 could be as close to state 3 as to state 2.

This situation is rather reminiscent of our earlier discussion of polynucleotide replication, and more generally of Darwinian evolution. If we start with a sequence that has low "fitness" (i.e. a relatively low selfreplicating ability), then as mutations occur, new but related sequences will be generated one of which may have increased replicative ability. If there is a limited supply of monomers, competition will lead to extinction of the initial sequence and takeover by the new, more efficient sequence. The population of molecules therefore moves in sequence space in the direction of increasing fitness, and will end up a local fitness peak.

An analogy with ferromagnetism also emerges, if we consider the relation between "harmony" and "potential energy". A rock placed on a slope falls to the bottom of the valley because it minimizes its potential energy. Likewise, if a lump of iron is cooled below T_c, its potential energy (reflecting the summed potential energy of all the magnetically interacting spins) decreases, as the spins adopt either the mostly up on mostly down states. Physical systems evolve to minimize their potential energy, just as species increase their fitness, and Hopfield neural networks maximize their harmony. In fact, Hopfield, trained as a physicist, analysed his network by defining a quantity he called "energy", which is defined as –Harmony. He showed that his rule minimized "energy", in just the same way we showed that it maximizes harmony. I have used the term harmony because it expresses the underlying idea of measuring the agreement between pairs of neurons, whereas energy is a less intuitive concept for biologists. These analogies can be made even more precise by introducing the concept of "temperature" (see below).

What relation do these ideas have to memory? Hopfield's brilliant insight was that if we can make the harmony peaks correspond to learned vectors, we would have an efficient autoassociative memory system, because if we start the network in a state that resembles a stored vector (e.g. a fragment of that vector) it would move toward that vector, and stabilize at that vector. Thus in the diagram above, the diamond might correspond to "Paul" and the harmony peak 3 to "Paul Adams", and the circle to "Fred" and the harmony peak 2 to "Fred Bloggs". There is an obvious way to ensure that harmony peaks correspond to stored vectors – use Hebb's rule to set up the weights during the learning phase. Since a weight is created using the rule $\Delta w^1_{i,j} = \Delta w^1_{j,i} = k f^1_i f^1_j$ (i.e. the increase in weight from neuron i to neuron j, and also from j to i, due to pattern 1 is determined by the product of the activities of the 2 neurons produced by pattern 1) it ensures that if the neurons' states correspond to pattern 1, the harmony due to this pair of neurons will be high.

Of course if several subsequent and different patterns are also stored , the final weight may differ from that prescribed by the first pattern, so the network cannot achieve as high a harmony level as would have been possible if only one pattern were stored. However, in an argument exactly analogous to the Linear Associator, it can be shown that as long as the stored patterns are all orthogonal, there is no interference between the weight changes produced by successive stored patterns, provided that the number of stored patterns is no greater than the number of neurons. Thus for orthogonal patterns recall is perfect – but this is no better than the Linear Associator. The real strength of the Hopfield network is that, unlike the Linear Associator, it can, within limits, *exactly* recall non-orthogonal patterns, because each of these patterns lies at a harmony peak and, as we have seen, the network moves uphill to harmony peaks and then stays there. It is also much better at using "hints" – fragments of memories – to reconstruct entire memories.

What are the limits? An intuitive grasp of these limits can be gained by introducing the notion of "basin of attraction". A harmony peak is described, in dynamical systems theory, as an "attractor" because it "attracts" to itself all similar states, like bathwater converging on a plughole, or iron converging to mostly- up or -down spin states. The set of states (the hill slopes in Fig 1) that converge on the attractor are called a "basin of attraction". The process of learning a set of patterns sets up a series of hills and valleys in the harmony landscape that constitute basins of attraction. (In the Harmony representation the network state flows up the harmony basins, while in the Energy representation the state flows down the basin). The more harmony peaks the network learns, the more basins of attraction are set up – but clearly the number of possible states of the system (2^n) remains constant. So as more patterns are learned, the smaller the basins become, and the more crowded the peaks. However, this increases the number of states that correspond to harmony minima, and as soon as these become as numerous as hillside states, errors occur (since these states cannot unambiguously resolve to unique harmony peaks). If the patterns are assumed to be random vectors, then it can be shown that the network can store about 0.15 n patterns if recall is not more than 1 % inaccurate. However, if more than this number of patterns is learned, none of them can be recalled – the network either stabilizes in a random pattern unrelated to any learned pattern, or recalls combinations of stored patterns – it gets confused. This sudden transition from excellent recall to zero recall at the critical network "loading" is quite unlike the gradual deterioration we saw in the linear associator (S/N = n/k), and is essentially a "phase transition from "ordered" behavior to "disordered" behavior.

The Hopfield net has a family resemblance to the models of ferromagnetism we considered at the start of this course. In the Ising model, neighboring spins try to flip each other (depending on the strength of their interaction J, which is equal for all neighbors, and zero for nonneighbors). In the Hopfield model, neurons try to flip each other according to the strength and sign of their interactions (which is $w_{i,j}$) and their physical location does not matter. In the Ising model, there are only 2 spin patterns that are compatible with the interaction rules: all up or all down. If the iron starts in a pattern that most resembles the "up" pattern, it converges to the all "up" pattern, and *vice versa* (at least below Tc). In the Hopfield model, there are many activity patterns that are at least partly compatible with the interaction rules defined by the Hebbian learning phase: the memorised patterns. The network converges to the pattern that most closely resembles the initial pattern.

Real neurons are noisy, partly because they have finite numbers of ion channels (so chance, which dominates at the molecular level, still affects macroscopic behavior) and partly because synapses operate probabilistically. This can be incorporated into the Hopfield model by making the synaptic input to a neuron affect its state in a probabalisitc fashion, rather than deterministically (as in Eq 1). If the net synaptic input is very strongly negative, we would want the neuron to be certainly "off", and if it is strongly positive the neuron should be "on"; if the net input is zero, the neuron should have equal chances of being on or off, and as the input grows the neuron should become more likely to be on or off. A suitable function that gives this behavior is the hyperbolic tangent function (see Ferromagnetism lecture). Thus Eq 1 is modified to

$$p_{1,i} = (\tanh \{[\Sigma\, w_{i,j}\, f_j\,]/T\} + 1)/2$$

where $p_{1,i}$ is the probability the ith neuron is in the +1 (i.e. "on") state, and T is a temperature-like steepness parameter. T reflects how noisy the neuron is, just as in the ferromagnet it reflects how “noisy” (i.e. subject to thermal buffeting) the spins are. (If Gaussian noise is added to the deterministic neuron of eq 1, then the additional noise term on the RHS can bring the neuron below the zero level threshold, preventing firing even though the net synaptic input is positive. This type of “mistake” becomes progressively less likely as the net synaptic input gets larger, and the net effect reflects the area under the Gaussian, which, like tanh, is sigmoid. At T = 0 Eq 4 reduces to eq 1; as T approaches infinity, it approaches linearity).

Making the neurons noisy makes it less likely the network can exactly recall a stored pattern. In fact, because the network is no longer deterministic it will continue to fluctuate slightly even though on average almost all the neurons are in the “correct” state. Nevertheless, below a critical temperature Tc , which will depend on how many memories were learned, the average final state will be correct. If the temperature is raised above Tc, recall will fail catastrophically i.e. the network undergoes a phase transition to a disordered state.

Even in the absence of added noise, if the network has learned many nonorthogonal patterns, then the “crosstalk” between patterns (see the second term on the right in Eq 2 of the Linear Associator lecture) acts as a source of noise. If the crosstalk noise exceeds a certain value, the network undergoes a phase transition to a disordered (non-retrieval) state. This limits the number of random patterns to 0.14 N.

What happens if the neurons are not noisy binary, but continuous? If the relation between neuron firing rate and input is specified by the above tanh function, then the continuous and noisy cases become almost exactly the same. The “temperature” T is now given by the steepness of the sigmoidal relation between input and firing rate. Clearly, the steeper the function, the better recall of random patterns will be. If the T is very high, the network approaches the performance of the linear associator, which we saw was very bad at recalling random patterns.

We are now in a position to visualize the associative memory properties of the hippocampus. During the learning phase, entorhinal activity patterns (collected from the rest of the neocortex) are imposed on the CA3 neurons via the granule cells and their strong mossy fiber synapses. Somehow the recurrent activity of the CA3 network is suppressed during this learning phase (possibly because cholinergic septal input is suppressed; this might allow strong adaptation by relieving inhibition of M and other K channels) but the plasticity of the recurrent synapses is enabled (allowing Hebbian learning). During recall, a memory fragment arrives via the mossy fibers, and this initiates reverberatory activity in the CA3 recurrent network (possibly because septal Ach release activates muscarinic receptors and blocks adaptation). During this recall phase the plasticity of the recurrent synapses would be disenabled (to prevent the system from learning fragments, or wrong answers). If the fragment is sufficiently similar to a stored memory, the activity of the CA3 network stabilises in the pattern provoked by the original neocortical pattern. However, this is still not really a “memory”, merely a label that the HC learned. To recall the true memory – i.e. the experience – this CA3 pattern is applied to CA1, which probably functions more like a feedforward linear associator, recalling another form of label which is then led back to neocortex. Note however that this CA1 label is identical to the activity pattern that reached CA1 during initial learning, so that the neocortex can, in turn, associate the CA1 label with the pattern of activity that the experience provoked. This means that the CA1 label, fed back to neocortex during recall, will re-elicit the original neocortical pattern.

The role of the granule cells is probably to “orthogonalise” the incoming entorhinal patterns, which will make them easier to store. One way this could happen is if there is strong inhibition between these cells (via GABAergic interneurons) which will enhance the “contrast” i.e. the difference between incoming patterns. It will also help that the granule cells are very numerous, so that the output is much “sparser” than the input (i.e a much smaller fraction of mossy fibers are active than of perforant fibers).