HOPFIELD NEURAL NETWORKS

Plan of the lesson

- Motivations: biological, historical, mathematical viewpoint.
- Introduction of Hopf eld Neural Network (mathematical formulation and example of connectivity matrix).

  Energy;

  Robusteness / Autoassociative memory;

  Signal to noise / Capacity
- Understanding biological elements

  Noise

  Sparsity

  Limited synapsis.
- Simulations

1 Motivations

Hopf eld Neural Networks have been crucial in the development of neuroscience. There are at least three ways to state this:

1.1 Biological

Think of a monkey that does the following experiment: at time zero it is f xating a point and is given a cue look either left or right after a delay of 2 seconds. After 2 seconds a go signal is given to the monkey and the monkey should look in the direction indicated. What happens during the delay in the brain of the monkey? Some neurons must hold the information regarding the cue. A way that Funahashi (J.Phisiology 1989) has shown

is that some neurons have a selective sustained activity during the delay. In other words specific neurons are active throughout the delay if cue-right was shown and others if cue-left was shown.

How can neurons fire in a sustained way if their membrane potential is continuosly reset? We need recurrent connections with delay of the firing neuron into itself. This is not possible at the level of single neuron but it is at the level of population. We are thus talking of recurrent neural networks and in particular we are interested in how to have sustained activity that is selective to a cue.

drawing of the monkey, sustained activity, recurrent neural network, selectivity to the cue.

1.2 Mathematical

In the realm of recurrent networks Hopfield neural networks have been one of the few examples that could be analyzed mathematically. This means that we know the behaviour of the network extremely well under different regimes. The simplest idea being that memories are energy minima of the energy of the network. This work started with Amit, Sompolinsky, Misha and others that in the late 80's and 90's have characterized this system with the tools of statistical physics. This was possible, in simple words, because this system has an energy function and all the statistical physics properties can be computed on it. This is really nice and made the system extremely useful for the purpose of modeling different brain functions. *mathematically analyzable recurrent network, statistical physics tool, memories as energy minima*

1.3 Historical

The Hopfield Network has been the king way of describing memories and learning for many experiments. Many publications have been using different variants of the Hopfield Network and extended it in different ways. Still it is a simple and succesful model. Historically has been the 'only known way' to design a neural network knowing its outcome and dynamic. Only in the last few years (deep learning) we have learnt how to teach networks arbitrary things not in a Hopfield/Hebbian way. But still while we know what happens in a hopfield network during learning, we don't really know what

happens during deep learning. *only successful simple model, 2 decades of successes in describing neuronal functions*

2 Hopfield Neural Network

ATTRACTOR neural network
(neural network converges to our attractors = memory states)
network moves towards local minima in energy (i.e. closest memory)

Consider a neural network with N binary neurons. Each node at every point in time can take either the value 1 or -1.

Memory states are vectors of size N of the type:

have several of these memory states for each memory (ie image) I have

$$\eta = -1111 - 1... - 1 \tag{1}$$

single neuron activity: 1 = ON, -1 = OFF

Every neuron is connected to all others by the connectivity matrix W:

summed combination of connectivity that would make sense for each memory
i.e. if ni and nj are always on at the same time in every memory, it makes sense theyhave a high positive connectivity

Hebian rule: neurons that fire together wire together

$$W_{ij} = \frac{1}{N}\sum_{\mu=1}^{P} \eta_i^\mu \eta_j^\mu \ . \tag{2}$$

where P is the number of memories that I want to store in the network.

We notice that matrix W_{ij} is symmetric.

The activity of neuron i is given by:

dynamics = how system evolves over time

current input into neuron i at a given time

r (t) is the activity vector of the neurons at a given time step
ri = activity of neuron i at the given time

$$h_i = \sum_j W_{ij} r_j \tag{3}$$

$$r_i = sign(h_i)$$

r_i = 1 if activity h_i >= 0 and -1 otherwise

The update of the network can be either synchronous (all nodes at ones) or asynchronous (the node to update is chosen randomly at each time step).

Example: connectivity matrix Example: update rule (12) 1-(3)-1, (13) 1-(-1)-0, (23) 1-2-0, (24) 1-(-4)-0, (34) 0-3-0, (45) 0-3-0, (35) 0-(-1)-0.

2.1 Energy Description

If we keep iterating, we will reach a fixed point!
corresponding to an energy minima

The energy function of the Hopfield model is:

Wij = connectivity matrix

$$E = -\sum_{ij} r_i w_{ij} r_j \tag{4}$$

= — sum (ri hi)

It assigns a number to each state and always decrease. If this is true there is a theorem (Lyapunov) that guarantees to us that the dynamic will converge to a fix point.

as our connectivity matrix is symmetric

Lyapunov function!

To show that the energies always decreases we compute the difference between the energy before and after the update of every single neural value. **Let us comput the change in energy after the update of neuron i:**

$$\Delta E = E(t+1) - E(t) = -(r_i(t+1) - r_i(t)) \sum_j W_{ij} r_j(t) = -(r_i(t+1) - r_i(t)) h_i(t) \ . \tag{5}$$

but both $h_i(t)$ and $r_i(t+1) - r_i(t)$ have the same sign so the product of the two is positive and thus the energy always decreases. or stays the same

Example

2.2 Robustness / autoassociativity

Now we want to understand better what is going on during the updates of the network. In particular we want to understand how robust is the network to random perturbations.

Consider the case where only one memory is stored:

$$r_i(t+1) = sign(\sum_j W_{ij} r_j(t)) = sign(\sum_j \frac{1}{N} \eta_i^1 \eta_j^1 r_j(t)); \tag{6}$$

Consider the case $r_j(t) = \eta_j^1$

$$r_i(t+1) = sign(\sum_j \frac{1}{N} \eta_i^1 \eta_j^1 \eta_j^1) = sign(\eta_i^1) = \eta_i^1 \tag{7}$$

The pattern is a fix point of the dynamics.

If we perturb $r_j(t)$ for a lot of neurons from being $r_j(t) = \eta_j$ I will still have the same activation of the memory. This is the idea that the memories are fix point attractors. To carry out this idea more formally and concretely we need to to a signal to noise analysis.

2.3 Capacity: Signal to Noise Analysis

Consider the case where P memories are stored in the network.

$$r_i(t+1) = sign(\sum_j \frac{1}{N} \sum_{\mu=1}^{P} \eta_i^\mu \eta_j^{mu} r_j(t)) \tag{8}$$

Let us assume that $r_j(t) = \eta_j^1$.

$$
\begin{aligned}
r_i(t+1) &= sign(\sum_j \frac{1}{N} \sum_{\mu=1}^{P} \eta_i^\mu \eta_j^\mu \eta_j^1(t)) = \\
&= sign(\sum_j \frac{1}{N} \eta_i^1 \eta_j^1 \eta_j^1 + \frac{1}{N} \sum_j \sum_{\mu=2}^{P} \eta_i^\mu \eta_j^\mu \eta_j^1) = \\
&= sign(\eta_i^1 + \frac{1}{N} \sum_j \sum_{\mu=2}^{P} \eta_i^\mu \eta_j^\mu \eta_j^1) = sign(\text{signal} + \text{noise})
\end{aligned} \tag{9}
$$

The noise is a term that can change the activation of the memory. This is given by the interference of the other stored memories with the desired one.

Let us analyse the noise term:

$$
\text{noise} = \frac{1}{N} \sum_j \sum_{\mu=2}^{P} \eta_i^\mu \eta_j^\mu \eta_j^1 \tag{10}
$$

The mean of the noise term is:

$$
< noise > = < \frac{1}{N} \sum_j \sum_{\mu=2}^{P} \eta_i^\mu \eta_j^\mu \eta_j^1 > = 0 \tag{11}
$$

The variance of the noise term is:

$$
\sigma^2_{\text{noise}} = < \text{noise}^2 > = < \frac{1}{N^2} \sum_{\mu=2}^{P} \sum_{\nu=2}^{P} \sum_j \sum_k \eta_i^\mu \eta_j^\mu \eta_j^1 \eta_i^\nu \eta_k^\nu \eta_k^1 > = \tag{12}
$$

$$
= (\text{k=j}, \mu = \nu) = \tag{13}
$$

$$
= < \frac{1}{N^2} \sum_{\mu=2}^{P} \sum_j \eta_i^\mu \eta_j^\mu \eta_j^1 \eta_i^\mu \eta_j^\mu \eta_j^1 > = \tag{14}
$$

$$
= \frac{N(P-1)}{N^2} \tag{15}
$$

We can also see this as for each i we have the sum of binomially distributed random number with mean zero and variance $\sqrt{P-1}$ and summing over N I obtain $\sqrt{(P-1)N}$ so that:

$$
\text{noise} = \mathcal{P}(\mu = 0, \sigma = \sqrt{\frac{P-1}{N}}) \tag{16}
$$

The probability of making an error in updating r_i is exactly the probability that the distribution $\mathcal{P}$ is minor than -1 or major than 1 so to change the sign of activation of the memory:

$$P_{error} = 1/2(1 - erf(\frac{S}{\sqrt{2}\sigma})) \tag{17}$$

Drawing of the gaussian. **So if we want no erro we need to have that $\sigma << 1$ and thus $P << N$.**

The capacity scales directly with N and in particular the actual value of threshold is $\alpha = \frac{P}{N} = 0.138$. where S is the signal S=1 and from the central limit theorem we know that $\mathcal{P}$ is gaussian with good approximation. This analysis makes us understand concretely how crossinteractions between memories are detrimental for a given memory to be stable. In other words, the more we store memories in the network and the more we have noise in retrieving correctly one memory. Usually this problem is addressed as the memory capacity problem of a neural network (aka. How many memories can I store?).

In particular we can fix P_{error} and define the loading level $\alpha = \frac{P}{N}$ as a function of P. A more complicated analysis shows that if $\alpha > 0.138$ the errors start piling up and the memory is lost.

To our purposes the interesting thing is that the probability depends only on α so we say that the capacity of the neural network is linear on N. The more neurons we have and the more memories we can store directly in the network.

3 Understanding biological elements

3.1 Noise

The problem of noise, Phase diagram Just in order to show more advanced results in the theory of Hopfield Neural Networks and understand why is it considered successful, we present the phase diagram. We just said that if the loading level α is higher than 0.138 than the memories in the network are stable anymore.

The noise term that we analysed is usually to reach such a conclusion is usually referred as cold noise. The reason being that it is due to the encoding of other memories. This name is in contrapposition to with the hot termal noise given by a probabilistic

description of the neurons themselves. In particular we would say that there is hot or thermal noise in the system if the update rule for a neuron is something of this kind: **To introduce hot thermal noise we make our neurons stochastics**:

$$\mathcal{P}(r_i(t) = \pm 1) = \frac{1}{1 + exp(\mp 2h_i(t)/T)} \tag{18}$$

instead of the one introduced above $r_i(t) = sign(h_i(t))$. **In this way we have introduced a temperature.** If the temperature tends to zero then we are in the limit studied so far. Otherwise we would have to analyse the system as a function of the temperature. **The result of the analisys can be drawn in a Phase diagram.**

The advantage of noise, Spurious States We have showed that the dynamic of the network converges to fix points through by seeing that the energy function always decreases and invoking the Lyapunov theorem. We have also seen that memories are fixed points of the system whenever we are in the loading capacity of the network. Although we haven't shown that the stored memories are the only fixed points of the system. **There are other fixed points or local minima of the energy function $\eta^1 + \eta^2 + \eta^3$. The noise destabilizes these patterns.** These are usually referred as spurious states. These states are usually given by the sum of three memories such as $\eta^1 + \eta^2 + \eta^3$ and are less stable than normal memories. For less stable we mean that their basin of attraction is smaller. Talking in these terms we can imagine that by introducing thermal noise these states may loose their stability before others. This is interesting as shows a simple but deep argument in favor of a system function for the noise. The function of destabilizing spurious states so that the network can easily exploits dynamics between the stored memories.

In the calculations so far we have seen the fact that memories are autoassociative or what is usually called pattern completion, that they are fixed points or energy minima of the system, that the capacity of a network as a given scaling and that noise may play a very key role in these networks.

Most of these considerations match our initial biological motivations for studying Hopfield Networks. We see now that Hopfield Networks can be seen as both models of long term memory as the memory are hard coded in the network, and short term memory as the network shows sustained activation of the selected memory.

3.2 Sparse Coding

One of the biological limitations of the binary Hopfield model is that memory patterns are random. In this way all neurons take part to encoding the memory. In the brain we see that memory are sparse. Neurons encoding specific memories are a few in all the brain and when the memory is retrieved only a few neurons are encoding this given memory. **Can we explain whether sparse coding is beneficial with Hopfiel Neural Network?** The answer is yes, we can show that with a sparse coding the capacity of the network improves.

Let us redefine the memories as:

$$\eta = 01110...0 \tag{19}$$

where a neuron is 1 with probability f and zero viceversa. We call f sparseness parameter as it controls how many neurons encode for a given memory.

The connectivity matrix is:

$$W_{ij} = \frac{1}{N}\sum_{\mu=1}^{P}(\eta_i^\mu - f)(\eta_j^\mu - f)\ . \tag{20}$$

We compute again the Signal to Noise ratio:

$$\begin{aligned} r_i(t+1) &= sign(\sum_j \frac{1}{N}\sum_{\mu=1}^{P}(\eta_i^\mu - f)(\eta_j^\mu - f)r_j(t) - \theta) = \\ &= sign(\sum_j \frac{1}{N}(\eta_i^1 - f)(\eta_j^1 - f)\eta_j^1 + \frac{1}{N}\sum_j\sum_{\mu=2}^{P}(\eta_i^\mu - f)(\eta_j^\mu - f)\eta_j^1 - \theta) = \\ &= sign((\eta_i^1 - f)(1-f)f + \frac{1}{N}\sum_j\sum_{\mu=2}^{P}(\eta_i^\mu - f)(\eta_j^\mu - f)\eta_j^1 - \theta) = \\ &= sign(\text{signal} + \text{noise} - \theta) \end{aligned} \tag{21}$$

The two terms are respectively:

signal $= (\eta_i^1 - f)(1-f)f$

$<$ noise^2 $>= \frac{1}{N}\sum_{\mu=2}^{P}(\eta_i^\mu - f)(\eta_j^\mu - f)\eta_j^1 = \frac{P-1}{N}f^3 = \alpha f^3$.

So as $\frac{\text{signal}}{\text{noise}} >> 1$ and the minimal value of signal is $-p^2$ we need to have $\alpha << p$.

Drawing. **If we adapt θ to be $\theta = \theta_0 p$ then the stability condition drastically changes**

to be:

$$\alpha << min(\frac{\theta^2}{p}, \frac{(1-\theta)^2}{p}) \tag{22}$$

This is a huge improvement in capacity.

3.3 limited synapses

One of the limitation of the Hopfield model is that synapsis can indefinitely increase as more memories are stored. **What happens if synapsis are limited?** Let us consider a model where we binarize synapses. How can we now store the patterns in a network of binary synapses?

A good way is to build the connectivity matrix as:

$$\begin{aligned} W_{ij}^{(1)} &= \frac{1}{N}\eta_i^1\eta_j^1 \\ W_{ij}^{(\mu+1)} &= \frac{1}{N}(W_{ij}(1-\beta_{ij}^{\mu+1}) + \eta_i^{\mu+1}\eta_j^{\mu+1}\beta_{ij}^{\mu+1}\ , \end{aligned} \tag{23}$$

where $\beta_{ij}^{\mu} = 0, 1$ are indipendend random numbers with probabilities:

$$\mathcal{P}(\beta = 1) = p;\ \mathcal{P}(\beta = 0) = 1 - p; \tag{24}$$

In this case if we repeat the signal to noise analysis we see that the number of patterns that we can store is not linear anymore in N but is of the order $N^{1/2}$.

Reservoir networks?

random reccurrent neural network of N neurons; random weights and connections

network creates dictionary of functions, which can compose any aspect of the input?

random dynamical reservoir ; feed some inputs and drive dynamical reservoir with training data; adjust output weights