Revised 24 February 2020 13:31

8 Recurrent networks of threshold (binary) neurons: Basis for associative memory

8.1 The network

A basic challenge in neurosience is to identify circuits for long-term memory. One solution suggested by theory involves associative networks, also referred to as content-addressable memories. In simplest form, these network solve a particualr form of multistabilty:

> Store a set of P patterns $\vec{\xi}^k$ in such a way that when presented with a new pattern $\vec{\mathbf{S}}^{test}$, the network responds by producing whichever one of the stored patterns most closely resembles $\vec{\mathbf{S}}^{test}$. Close is defined by the Hamming distance, the number of different "bits" in the pattern.

The neurons are labelled by $i = 1,\, 2,\, \ldots\, ,\, N$ and the individual stable patterns are labeled by $k = 1,\, 2,\, \ldots\, ,\, P$.

We denote the activity of the $i - th$ neuron by S_i. The input to neuron i is denoted by μ_i and is given by

$$\mu_i \;= \sum_{j=1;\, j\neq i}^{N} W_{ij} S_j + I_i^{ext} \tag{8.8}$$

where the W_{ij} are analog-valued synaptic weights and I_i^{ext} is an external input. The dynamics of the network are:

$$S_i \equiv sgn\left(\mu_i - \theta_i\right) \tag{8.9}$$

where θ_i is the threshold and we take the sign function sign $sgn(h)$ to be

$$sgn(x) = \begin{cases} \;\;1 & \text{if } x \geq 0 \\ -1 & \text{if } x < 0 \end{cases}$$

Clearly the output S_I is driven by the external input when I_i^{ext} is sufficiently large.

Going forward, we may take $\theta_i = 0\ \forall i$ as befits the case of random patterns on which neuronal outputs take on the values $+1$ and -1 with equal probability. In the further absence of external input, we have the minimal description

$$S_i \equiv sgn\left(\sum_{j\neq i}^{N} W_{ij} S_j\right). \tag{8.10}$$

There are at least two ways in which we might carry out the updating specified by the above equation. We could do it *synchronously*, updating all units simultaneously at each time step. Or we could do it *asynchronously*, updating them one at a time. Both kinds of models are interesting, but the asynchronous choice is more natural for both brains and artificial networks. The synchronous choice requires a central clock or pacemaker, and is potentially sensitive to timing errors, as is the case of *sequential* updating. In the asynchronous case, which we adopt henceforth, we can proceed in either of two ways:

- At each time step, select at random a unit i to be updated, and apply the update rule.
- Let each unit independently choose to update itself according to the update rule, with some constant probability per unit time.

These choices are equivalent, except for the distribution of update intervals. For the second case there is vanishing small probability of two units choosing to update at exactly the same moment.

Rather than study a specific problem such as memorizing a particular set of pictures, we examine the more generic problem of a *random* set of patterns drawn from a distribution. For convenience, we will usually take the patterns to be made up of independent bits ξ_i that can each take on the values +1 and -1 with equal probability.

Our procedure for testing whether a proposed form of W_{ij} is acceptable is first to see whether the patterns to be memorized are themselves stable, and then to check whether small deviations from these patterns are corrected as the network evolves.

8.2 Storing one pattern

To motivate our choice for the connection weights, we consider first the simple case whether there is just one pattern ξ_i that we want to memorize. The condition for this pattern to be stable is just

$$sgn\left(\sum_{j\neq i}^{N} W_{ij}\xi_j\right) = \xi_i \qquad \forall i \tag{8.11}$$

since the update rule produces no changes. It is easy to verify this if we take

$$W_{ij} \propto \xi_i\xi_j \tag{8.12}$$

since $\xi_j^2 = 1$. We take the constant of proportionality to be $1/N$, where N is the number of units in the network, which yields

$$W_{ij} = \frac{1}{N}\xi_i\xi_j \quad . \tag{8.13}$$

Furthermore, it is also obvious that even if a number (fewer than half) of the bits of the starting pattern S_i are wrong, *i.e.*, not equal to ξ_i, they will be overwhelmed in the sum for the net input $\sum_{j\neq i}^{N} W_{ij}S_j$ by the majority that are correct so that $sgn(\sum_{j\neq i}^{N} W_{ij}S_j)$ will still give ξ_i.

An initial configuration near to ξ_i will therefore quickly relax to ξ_i. This means that the network will correct errors as desired, and we can say that the pattern ξ_i is an **attractor**. Actually, there are two attractors in this simple case; the other one is at $-\xi_i$. This is called a **reversed state**. All starting configurations with *more* than half the bits different from the original pattern will end up in the reversed state.

8.3 Storing many patterns

How do we get the system to recall the most similar of many patterns? The simplest answer is just to make the synaptic weights W_{ij} by an outer product rule for each of the P patterns, which corresponds to

$$W_{ij} = \frac{1}{N}\sum_{k=1}^{P} \xi_i^k \xi_j^k \quad . \tag{8.14}$$

The above rule for synaptic weights is called the "Hebb rule" because of the similarity with a hypothesis made by Hebb (1949) about the way in which synaptic strengths in the brain change in response to experience: Hebb suggested changes are proportional to the correlation between the firing of the pre- and post-synaptic neurons.

8.4 Scaling for error-free storage of many patterns

We consider a Hopfield network with the standard Hebb-like learning rule and ask how many memories we can imbed in a network of N neurons with the constraint that we will accept at most one bit of error, i.e., one neuron's output in only one of the memory states. The input is

$$\begin{aligned} \mu_i \quad &= \sum_{j\neq i}^{N} W_{ij}S_j \qquad (8.15)\\ &= \frac{1}{N}\sum_{k=1}^{P}\sum_{j\neq i}^{N} \xi_i^k \xi_j^k S_j. \end{aligned}$$

Let $S_j \equiv \xi_j^1$, one of the stored memory states, so that

$$\begin{aligned} \mu_i \quad &= \frac{1}{N}\sum_{k=1}^{P}\sum_{j\neq i}^{N} \xi_i^k \xi_j^k \xi_j^1 \qquad (8.16)\\ &= \frac{1}{N}\sum_{k=1}^{P} \xi_i^k \sum_{j\neq i}^{N} \xi_j^k \xi_j^1 \\ &= \frac{1}{N}\ \xi_i^1 \sum_{j\neq i}^{N} \xi_j^1 \xi_j^1 + \frac{1}{N}\sum_{k\neq 1}^{P} \xi_i^k \sum_{j\neq i}^{N} \xi_j^k \xi_j^1 \\ &= \frac{N-1}{N}\xi_j^1 + \frac{1}{N}\sum_{k\neq 1}^{P} \xi_i^k \sum_{j\neq i}^{N} \xi_j^k \xi_j^1 \end{aligned}$$

Thus, in the limit of large N, the first term leads to stability while the second term goes to zero, so that the average input is

$$< \mu_i > \;\; \simeq \;\; \xi_i^1 \tag{8.17}$$

Even when the second term for pattern 1 is not zero, the state $\vec{\xi}^1$ is stable if the magnitude of the second term is smaller than 1, i.e., if the second term cannot change the sign of the output S_i^l. It turns out that the second term *is* less than 1 in many cases of interest if P, the number of patterns, is sufficiently small. Then the stored patterns are all stable – if we start the system from one of these states the system will remain in that state. A small fraction of bits different from a stored pattern will be corrected in the same way as in the single-pattern case; they are overwhelmed in $\sum_{j\neq i}^N \sum_{k\neq l}^P W_{ij} S_j$ by the vast majority of correct bits. A configuration near to ξ_i^1 thus relaxes to ξ_i^1.

What is the variance, denoted σ^2, induced by the storage of many memories, the so-called structural noise? The second term consists of $(P-1)$ inner products of random vectors with $(N-1)$ terms. Each term is $+1$ or -1, i..e., binomially distributed, so that the fluctuation to the input is

$$\begin{aligned} \sigma \quad &= \frac{1}{N} \cdot \sqrt{P-1} \cdot \sqrt{N-1} \\ &\simeq \sqrt{\frac{P}{N}}. \end{aligned} \tag{8.18}$$

More laboriously,

$$\begin{aligned} \sigma^2 \quad &= \frac{1}{N} \sum_{i=i}^{N} \left(\frac{1}{N} \sum_{k\neq 1}^{P} \xi_i^k \sum_{j\neq i}^{N} \xi_j^k \xi_j^1 \right) \left(\frac{1}{N} \sum_{k'\neq 1}^{P} \xi_i^{k'} \sum_{j'\neq i}^{N} \xi_{j'}^{k'} \xi_{j'}^1 \right) \\ &= \frac{1}{N^3} \sum_{k\neq 1}^{P} \sum_{k'\neq 1}^{P} \left(\sum_{i=i}^{N} \xi_i^k \xi_i^{k'} \right) \sum_{j\neq i}^{N} \xi_j^k \xi_j^1 \sum_{j'\neq i}^{N} \xi_{j'}^{k'} \xi_{j'}^1 \\ &\overrightarrow{N \to \infty} \; \frac{1}{N^3} \sum_{k\neq 1}^{P} \sum_{k'\neq 1}^{P} \; N \; \delta(k-k') \; \sum_{j\neq i}^{N} \xi_j^k \xi_j^1 \sum_{j'\neq i}^{N} \xi_{j'}^{k'} \xi_{j'}^1 \\ &\overrightarrow{N \to \infty} \; \frac{1}{N^2} \sum_{k\neq 1}^{P} \;\; \sum_{j\neq i}^{N} \xi_j^k \xi_j^1 \sum_{j'\neq i}^{N} \xi_{j'}^{k} \xi_{j'}^1 \\ &\overrightarrow{N \to \infty} \; \frac{1}{N^2} \sum_{j\neq i}^{N} \xi_j^1 \sum_{j'\neq i}^{N} \xi_{j'}^1 \left(\sum_{k\neq 1}^{P} \;\; \xi_j^k \xi_{j'}^k \right) \\ &\overrightarrow{N \to \infty;\; P \to \infty} \; \frac{1}{N^2} \sum_{j\neq i}^{N} \xi_j^1 \sum_{j'\neq i}^{N} \xi_{j'}^1 \; (P-1) \; \delta(j-j') \\ &\overrightarrow{N \to \infty;\; P \to \infty} \; \frac{P-1}{N^2} \sum_{j\neq i}^{N} \left(\xi_j^1 \right)^2 \\ &\overrightarrow{N \to \infty;\; P \to \infty} \; \frac{(P-1)(N-1)}{N^2} \\ &\overrightarrow{N \to \infty;\; P \to \infty} \; \frac{P}{N} \end{aligned} \tag{8.19}$$

Noise hurts only if the magnitude of the noise term exceeds $\sigma = 1$. By the Central Limit Theorem, the noise becomes Gaussian for large P and N, but constant P/N. Thus the probability of an error in the recall of all stored states is

$$\begin{aligned} p_{error} &= \frac{1}{\sqrt{2\pi}\ \sigma} \left[\int_{-\infty}^{-1} e^{-x^2/2\sigma^2}\ dx\ + \int_{+1}^{\infty} e^{-x^2/2\sigma^2}\ dx \right] \quad (8.20) \\ &= \frac{\sqrt{2}}{\sqrt{\pi}\ \sigma} \int_{+1}^{\infty} e^{-x^2/2\sigma^2}\ dx \\ &= \frac{2}{\sqrt{\pi}} \int_{\frac{1}{\sqrt{2}\sigma}}^{\infty} e^{-x^2}\ dx \\ &\equiv \mathrm{erfc}\left(\frac{1}{\sqrt{2}\sigma}\right) \end{aligned}$$

where efrc(x) is the complementary error function and we again note that the average of the error term is zero. Thus

$$p_{\mathrm{error}} = \mathrm{erfc}\left(\sqrt{\frac{N}{2P}}\right). \tag{8.21}$$

For $N/P >> 1$ the complementary error function may be approximated by an asymptotic closed form given by

$$p_{\mathrm{error}} \simeq \frac{2}{\sqrt{\pi}} \frac{P}{N}\ e^{-N/2P} \tag{8.22}$$

so that to leading order

$$log\{p_{\mathrm{error}}\} \simeq -\frac{N}{2P} - \log\{\frac{N}{P}\}. \tag{8.23}$$

Now NP is total number of "bits" in the network. Suppose only less than one bit can be in error. Then we equate probabilities of correct to within a factor of one bit, or $1/(NP)$. Thus

$$1 - p_{\mathrm{error}} \geq 1 - \frac{1}{NP} \tag{8.24}$$

or

$$\log\{p_{\mathrm{error}}\} < -\log\{NP\}. \tag{8.25}$$

Thus

$$-\frac{N}{2P} - \log\{\frac{N}{P}\}\ < -\log\{NP\} \tag{8.26}$$

or

$$-\frac{N}{2P} < -2\log\{P\} \tag{8.27}$$

so

$$P\ <\ \frac{1}{4}\ \frac{N}{\log\{P\}}. \tag{8.28}$$

Since P scales sublinearly with N, we can iterate to write

$$P \;<\; \frac{1}{4}\,\frac{N}{\log\{N\}}. \tag{8.29}$$

Thus we see that an associate memory based on a recurrent Hopfield network stores a number of memories that scales more weakly than the number of neurons if one cannot tolerate any errors upon recall. Keep a mind that a linear network stores only one stable state, e.g., an integrator state. So things are looking good. Lastly, we note that P has a similar scaling for the choice of a fixed, nonzero error rate.

8.5 Energy description and convergence

The following notes were abstracted from Chapter 2 of "Introduction to the Theory of Neural Computation" (Addison Wesley, 1991) by Hertz, Krogh and Palmer.

One of the most important contributions of Hopfield was to introduce the idea of an *energy function* into neural network theory. For the networks we are considering, the energy function E is

$$E = -\frac{1}{2}\sum_{ij;i\neq j}^{N} W_{ij}S_iS_j \quad . \tag{8.30}$$

The double sum is over all i and all j. The $i = j$ terms are of no consequence because $S_i^2 = 1$; they just contribute a constant to E, and in any case we could choose $W_{ii} = 0$. The energy function is a function of the configuration S_i of the system, where S_i means the set of all the S_i's. Typically this surface is quite hilly.

The central property of an energy function is that it *always decreases (or remains constant) as the system evolves according to its dynamical rule.* Thus the attractors (memorized patterns) are at local minima of the energy surface.For neural networks in general an energy function exists if the connection strengths are *symmetric*, *i.e.*, $W_{ij} = W_{ji}$. In real networks of neurons this is an unreasonable assumption, but it is useful to study the symmetric case because of the extra insight that the existence of an energy function affords us. The Hebb prescription that we are now studying automatically yields symmetric W_{ij}'s.

For symmetric connections we can write the energy in the alternative form

$$E = -\sum_{(ij)}^{N} W_{ij}S_iS_j + \text{constant} \tag{8.31}$$

where (ij) means all the distinct pairs of ij, counting for example "1,2" as the same pair as "2,1". We exclude the ii terms from (ij); they give the constant. It now is easy to show that the dynamical rule can only decrease the energy. Let S_i' be the new value of S_i for some particular unit i:

$$S_i' = sgn\left(\sum_{j\neq 1}^{N} W_{ij}S_j\right) \quad . \tag{8.32}$$

Obviously if $S_i' = S_i$ the energy is unchanged. In the other case $S_i' = -S_i$ so, picking out the terms that involve S_i

$$\begin{aligned} E' - E &= -\sum_{j\neq i}^{N} W_{ij} S_i' S_j + \sum_{j\neq i}^{N} W_{ij} S_i S_j \qquad (8.33) \\ &= 2S_i \sum_{j\neq i}^{N} W_{ij} S_j. \end{aligned}$$

This term is negative from the update rule. Thus the energy decreases every time an S_i changes, as claimed.

The idea of the energy function as something to be minimized in the stable states gives us an alternate way to derive the Hebb prescription. Let us start again with the single-pattern case. We want the energy to be minimized when the overlap between the network configuration and the stored pattern ξ_i is largest. So we choose

$$E = -\frac{1}{2N} \sum_{k=1}^{P} \left(\sum_{i=1}^{N} S_i \xi_i^k \right)^2 \quad . \qquad (8.34)$$

Multiplying this out gives

$$\begin{aligned} E &= -\frac{1}{2N} \sum_{k=1}^{P} \left(\sum_{i=1}^{N} S_i \xi_i^k \right) \left(\sum_{j=1}^{N} S_j \xi_j^k \right) \qquad (8.35) \\ &= -\frac{1}{2} \sum_{i\neq j}^{N} \left(\frac{1}{N} \sum_{k=1}^{P} \xi_i^k \xi_j^k \right) S_i S_j \end{aligned}$$

which is exactly the same as our original energy function if W_{ij} is given by the Hebb rule. This approach to finding appropriate W_{ij}'s is generally useful. If we can write down an energy function whose minimum satisfies a problem of interest, then we can multiply it out and identify the appropriate strength W_{ij} from the coefficient of $S_i S_j$.

8.6 The issue of spurious attractors

The following notes were abstracted from Chapter 2 of "Introduction to the Theory of Neural Computation" (Addison Wesley, 1991) by Hertz, Krogh and Palmer.

We have shown that the Hebb prescription gives us (for small enough P) a dynamical system that has attractors – local minima of the energy function – for the desired states $\vec{\xi}^k$. These are sometimes called the **retrieval states**. But we have not shown that these are the only attractors. And indeed there are others, as discovered by by Amit, Gottfried and Sompolinsky (1985).

First of all, the reversed states $-\vec{\xi}^k$ are minima and have the same energy as the original patterns. The dynamics and the energy function both have a perfect symmetry, $S_i \leftrightarrow -S_i \ \forall\, i$. This is not too troublesome for the retrieved patterns; we could agree to reverse all the remaining bits when a particular "sign bit" is –1 for example.

Second, there are stable **mixture states** $\vec{\xi}^{mix}$, which are not equal to any single pattern, but instead correspond to linear combinations of an odd number of patterns. The simplest of these are symmetric combinations of three stored patterns with components:

$$\xi_i^{mix} = sgn(\pm\xi_i^1 \pm \xi_i^2 \pm \xi_i^3) \quad . \tag{8.36}$$

All $2^3 = 8$ sign combinations are possible, but we consider for definiteness the case where all the signs are chosen as +'s. The other cases are similar. Observe that on average ξ_i^{mix} has the same sign at ξ_i^1 three times out of four; only if ξ_i^2 and ξ_i^3 both have the opposite sign can the overall sign be reversed? So ξ_i^{mix} is Hamming distance $N/4$ from ξ_i^1, and of course from ξ_i^2 and ξ_i^3 too; the mixture states lie at points equidistant from their components. This also implies that $\sum_i \xi_i^1 \xi_i^{mix} = N/2$ on average. To check the stability pick out the three special states with $k = 1, 2,$ and 3, still with all + signs, to find:

$$\begin{aligned} \mu_i^{mix} &= \frac{1}{N}\sum_{j=1}^{N}\sum_{k=1} \xi_i^k \xi_j^k \xi_j^{mix} \qquad (8.37) \\ &= \frac{1}{2}\xi_i^1 + \frac{1}{2}\xi_i^2 + \frac{1}{2}\xi_i^3 + \text{cross} - \text{terms} \quad . \end{aligned}$$

Thus the stability condition is satisfied for the mixture state. Similarly 5, 7, ... patterns may be combined. The system does not choose an *even* number of patterns because they can add up to zero on some sites, whereas the units have to have nonzero inputs to have defined outputs of ± 1.

Third, for large P there are local minima that are not correlated with any finite number of the original patters $\vec{\xi}^k$.

8.7 The phase diagram of the Hopfield model

The following notes were abstracted from Chapter 2 of "Introduction to the Theory of Neural Computation" (Addison Wesley, 1991) by Hertz, Krogh and Palmer.

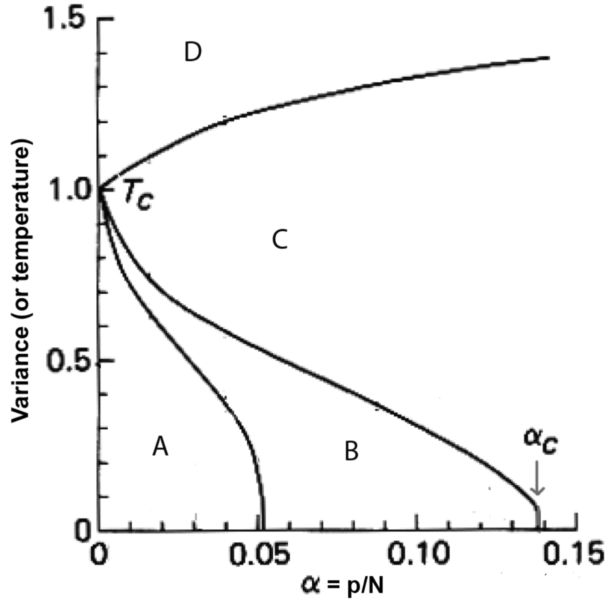
A statistical mechanical analysis by Amit, Gottfried and Sompolinsky (1985) shows that there is a crucial value of P/N where memory states no longer exist. A numerical evaluation gives

$$\alpha_C \equiv \frac{P}{N}|_{\text{critical}} \approx 0.138 \quad . \tag{8.38}$$

The jump in the number of memory states is considerable: from near-perfect recall to zero. This tells us that with no internal noise we go discontinuously from a very good working memory with only a few bits in error for $\alpha < \alpha_C$ to a "useless" memory system for $\alpha > \alpha_C$.

The **phase diagram** for the Hopfield model delineates different regimes of behavior in the $Variance - \alpha$ plane (variance is σ^2 in our notation, but the statistical mechanics literature uses T for temperature). There is a roughly triangular region where the network is a good memory device, as indicated by regions A and B of the embedded figure. The result corresponds to the upper limit of α_C on the α axis, while the critical variance $T_C = 1$ for the $P \ll N$ case sets the limit on the nariance axis. Between these limits there is a maximum variance or maximum load defined by a phase-transition line. As $Variance \to 1$, $\alpha_C(T)$ goes to zero like $(1-T)^2$.

In region C of the phase diagram the network still turns out to have many stable states, called **spin glass states**, but these are not correlated with any of the patterns ξ_i^k. However, if T is raised to a sufficiently high value, into region D, the output of the network continuously fluctuates with $\langle S_i \rangle = 0$.



Regions A and B of the phase diagram both have the desired retrieval states, beside some percentage of wrong bits, but also have spin glass states. The spin states are the most stable states in region B, lower in energy than the desired states, whereas in region A the desired states are the global minima. For small enough α and $Variance$ there are also mixture states that are correlated with an odd number of the patterns as discussed earlier. These always have higher free energy than the desired states. Each type of mixture state is stable in a triangular region (A and B), but with smaller intercepts on both axes. The most stable mixture states extend to 0.46 on the $Variance$ axis and 0.03 on the α axis (a subregion of A).

8.8 Noise and spontaneous excitatory states

It is worth asking if, by connection with ferromagnetic systems, rate equations of the form used for the Hopfield model naturally go into an epileptic state of continuous firing, but not necessarily with every cell firing. This exercise also allows us to bring up the issue of fast noise (variance) that is uncorrelated from neuron to neuron.

We consider N binary neurons, with $N >> 1$, each of which is connected to all other neighboring neurons. For simplicity, we assume that the synaptic weights W_{ij} are the same for each connections, *i.e.*, $W_{ij} = W_0$. Then there is no spatial structure in the network and the total input to a given cell has two contributions. One term from the neighboring cells and one from an external input, which we also take to be the same for all cells and denote I^{ext}. Then the input is

$$\mu_i \;=\; W_0 \sum_{j=1}^{N} S_j \;+\; I^{ext}. \tag{8.39}$$

The energy per neuron, denoted ϵ_i, is then

$$\begin{aligned} \epsilon_i \;&=\; -S_i\, \mu_i \\ &=\; -S_i\, W_0 \sum_{j=1}^{N} S_j \;-\; S_i\, I^{ext} \end{aligned} \tag{8.40}$$

The insight for solving this system is the mean-field approach. We replace the sum of all neurons by the mean value of S_i, denoted $< S >$, where

$$< S > = \frac{1}{N} \sum_{j=1}^{N} S_j. \tag{8.41}$$

so that

$$\epsilon_i \;=\; -S_i\, (W_0 N < S > \;+ I^{ext}). \tag{8.42}$$

We can now use the expression for the value of the energy in term of the average spike rate, $< S >$, to solve self consistently for $< S >$. We know that the average rate is given by a Boltzman factor over all of the S_i. Thus

$$\begin{aligned} < S > \quad &= \frac{\sum_{S_i=\pm 1} \; S_i \; e^{-\epsilon_i/k_B T}}{\sum_{S_i=\pm 1} \; e^{-\epsilon_i/k_B T}} \\ &= \frac{\sum_{S_i=\pm 1} \; S_i \; e^{S_i(W_0 N<S>+I^{ext})/k_B T}}{\sum_{S_i=\pm 1} \; e^{S_i(W_0 N<S>+I^{ext})/k_B T}} \\ &= \frac{e^{-(W_0 N<S>+I^{ext})/k_B T} \;-\; e^{(W_0 N<S>+I^{ext})/k_B T}}{e^{-(W_0 N<S>+I^{ext})/k_B T} \;+\; e^{(W_0 N<S>+I^{ext})/k_B T}} \\ &= \tanh\left(\frac{W_0 N < S > + I^{ext}}{k_B T}\right). \end{aligned} \tag{8.43}$$

where we made of the fact that $S_i = \pm 1$. This is the neuronal equivalent of the famous Weiss equation for ferromagnetism. The properties of the solution clearly depend on the ratio $\frac{W_0 N}{k_B T}$, which pits the connection strength W_0 against the noise level T/N. We also see how the input-output function $tanh\{x\}$ naturally arises.

- For $\frac{W_0 N}{k_B T} \;<\; 1$, the high noise limit, there is only the solution $< S >= 0$ in the absence of an external input h_0.
- For $\frac{W_0 N}{k_B T} \;>\; 1$, the low noise limit, there are three solutions in the absence of an external input h_0. One has $< S > = 0$ but is unstable. The other two solutions have $< S > \neq 0$ and must be found graphically or numerically.

- For sufficiently large $|I^{ext}|$ the network is pushed to a state with $< S >= sgn(I^{ext}/k_B T)$ independent of the interactions.

We see that there is a critical noise level for the onset of an active state and that this level depends on the strength of the connectionsand the number of cells. We also see that an active state can occur spontaneously for $\frac{W_0 N}{k_B T} > 1$ or $T < \frac{W_0 N}{k_B}$. This is a metaphor for epilepsy, in which recurrent excitatory connections maintain a spiking output (although a lack of inhibition appears to be required as a seed).