Physics 178/278 - David Kleinfeld - Winter 2021 - Lecture 2

Revised 7 January 2021 20:00

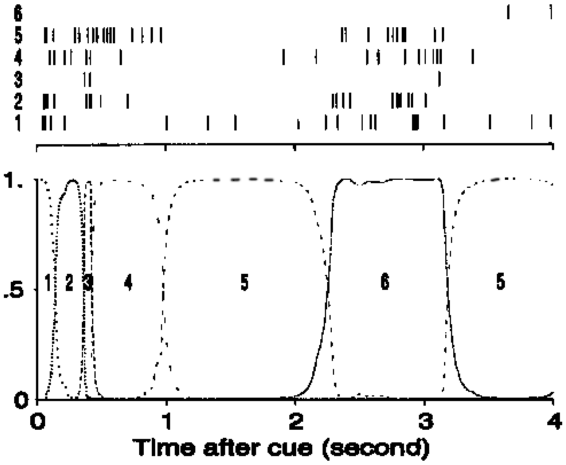
2 Recurrent neuronal networks: Associative memory 1

2.1 What is a state?

We previously considered the output from neuronal networks with only two cells, so the notion of a state was pretty obvious. In general, the state is simply the arrangement of ON or active neurons (+1) and OFF or quiescent neurons (-1) under observation. Ideally this is every neuron in the circuit, which is possible in some preparations, like the invertebrate preparations at the end of lesson 1. In some large preparations, the size of the animal or brain region is sufficiently small that preparations with hundreds to thousands of contiguous neurons can be imaged with sufficient speed and reliability. Do large systems also exhibit detectable states?

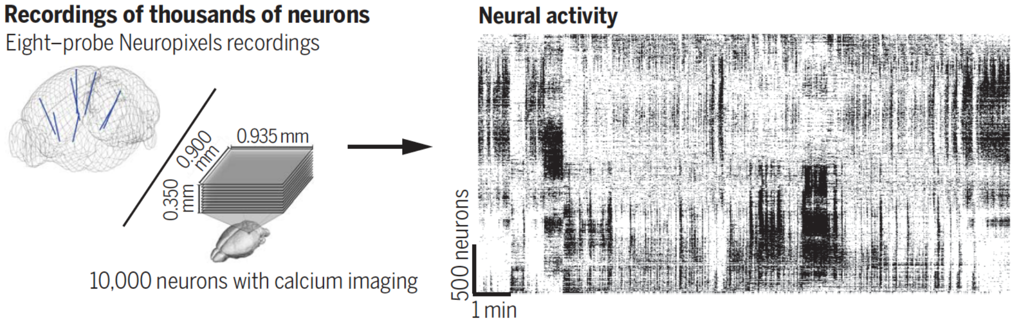
The ideal of repeating patterns came to the front many years ago in the cortical studies of Moshe Abeles. They recorded from frontal areas of monkey cortex and tended to see repeated patterns of spikes even though they recorded from relatively few cells. Judge for yourself!

Figure 1: Firing times of six neurons in monkey frontal cortex over a total of 93 trials were used to construct an HMM. Six states were identified. From Abeles, Bergman, Gati, Meilijson, Seidemann, Tishby and Vaadia 1995.



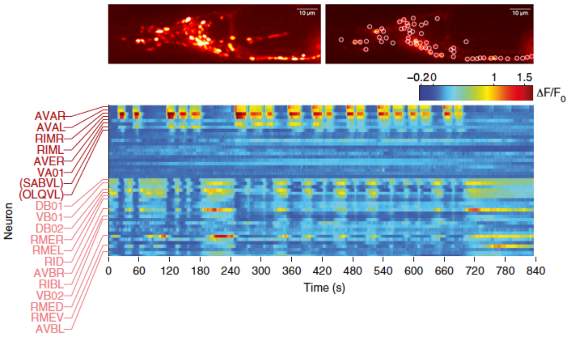
Zooming up to modern times, the technology has vastly improved to gett a much better view across very many neurons, as recently measured with electrodes across wide volumes of the brain by Mateo Carandini and Kenneth Harris. We see many repeating or near repeating patterns among what is really a very sparse sample, i.e, 10^4 neurons among the 10^8 neurons in the mouse brain. The same neurons can be active or quiescent across a multitude of states.

Figure 2: Sorted output from Neuropixels probes in the brain of mouse, From Stringer, Pachitariu, Steinmetz, Reddy, Carandini and Harris 2019



Finally, states appear to occur in preparations that contain tens to hundreds of neurons in which every cell can be observed at effectively the same time. This is shown for the worm c. Elegans. Again, we leave interpretation aside and simple note the clear occurrence of four states.

Figure 3: Calcium imaging from c. Elegan neurons during movement. Kato, Kaplan, Schrodel, Skora, Lindsay, Yemini, Lockery and Zimmer 2015



One special aspect of all of these and related data is that stable firing patterns exist. In the last two case would could see patterns without special statistical tools - just recording of the presentation. A second aspect is that the number of states are few, i.e., far less than the number of cells, denoted N, and far, far less than the number of possible states, i.e., 2^N, although likely large than the minimum number of connectivity of a graph, i.e., $NlogN$.

2.2 Are real networks highly interconnected?

The connectome of very few animals has been brain completed. In fact, only the connections among the neural integrator for horizontal eye position position in the juvenile zebrafish has been reconstructed over a large enough region - to date - to draw any conclusions. Here about 0.1 of the neurons make recurrent connections on each other; this should be taken as a lower bound on connections. In any case all this means is that we need $0.1 * N >> logN$ or $N >> 35$, which is consistent with about 500 neurons in the integrator.

Figure 4: Velocity-to-position neural integrator. Schematic showing the proposed wiring of modO, cells that project to the periphery, along with the two submodules modOI and modOM, and DO neurons that synapses onto ABDM and ABDI. From Vishwanathan, Ramirez, Wu, Sood, Yang, Kemnitz, Ih, Turner, Lee, Tartavull, Silversmith, Jordan, David, Bland, Goldman, Aksay and Seung, unpublished

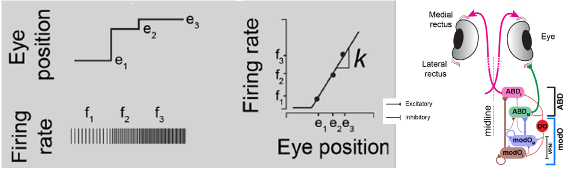
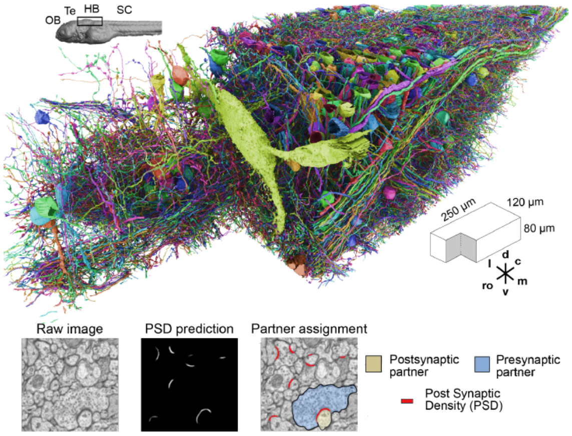


Figure 5: Cut-section view of the reconstructed volume and labeling of a synapse. From Vishwanathan, Ramirez, Wu, Sood, Yang, Kemnitz, Ih, Turner, Lee, Tartavull, Silversmith, Jordan, David, Bland, Goldman, Aksay and Seung, unpublished



2.3 The network

We consider the dynamics of a fully connected recurrent neuronal network. We will begin our analysis guided by this task:

> 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

$$\mu_i \ = \sum_{j=1;\ j\neq i}^{N} W_{ij} S_j + I_i^{ext} \tag{2.2}$$

Figure 6: Connectivity matrix of center neurons organized into two modules (modA, modO). Neurons in the center were clustered whereas neurons in the periphery were not. Neurons in the periphery were organized by known cell types, vSPNs and ABD neurons. Colored dots represent the number of synapses. From Vishwanathan, Ramirez, Wu, Sood, Yang, Kemnitz, Ih, Turner, Lee, Tartavull, Silversmith, Jordan, David, Bland, Goldman, Aksay and Seung, unpublished

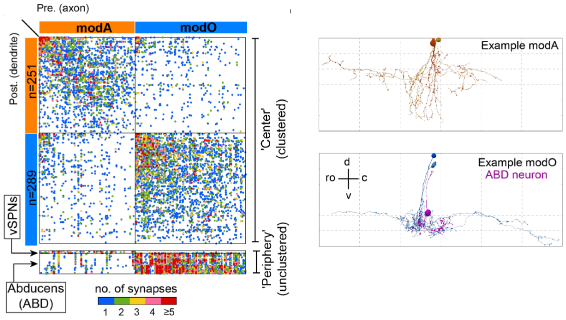
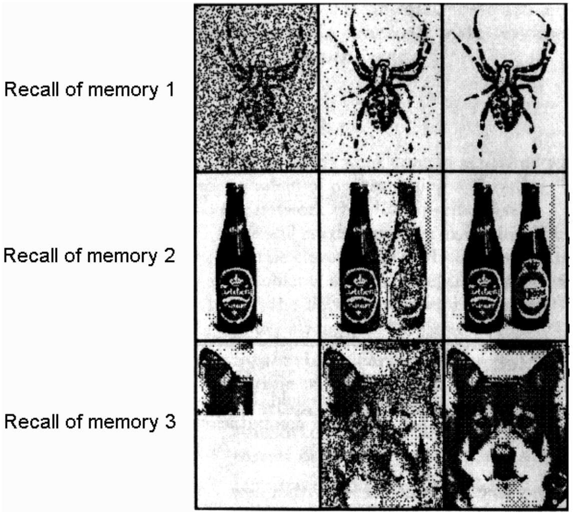


Figure 7: Function of the network as a content addressable memory in the recovery of a full memory from partial initial information. from Hertz, Krogh and Palmer 1991, following Hopfield 1982.



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

$$S_i \equiv sgn\left(\mu_i - \theta_i\right) \tag{2.3}$$

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

Figure 8: Basic associative or "Hopfield" network. From Hertz, Krogh and Palmer 1991, following Hopfield 1982.

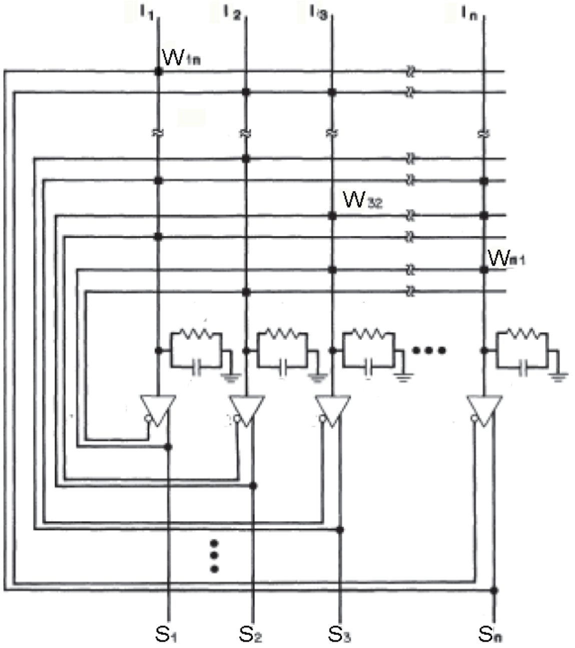
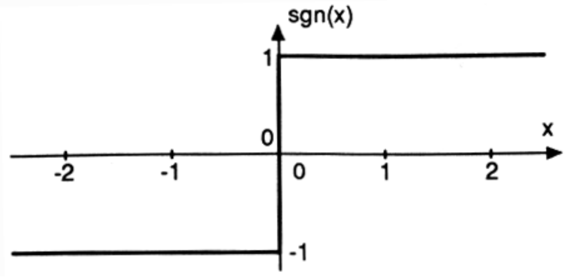


Figure 9: Input-output relation. From Hertz, Krogh and Palmer 1991, following Hopfield 1982.



the further absence of external input, we have the minimal description

$$S_i \equiv sgn\left(\sum_{j\neq i}^{N} W_{ij}S_j\right) . \qquad (2.4)$$

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.

2.4 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

$$sgn\left(\sum_{j\neq i}^{N} W_{ij}\xi_j\right) = \xi_i \qquad \forall i \tag{2.5}$$

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

$$W_{ij} \propto \xi_i\xi_j \tag{2.6}$$

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

$$W_{ij} = \frac{1}{N}\xi_i\xi_j \quad . \tag{2.7}$$

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.

2.5 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

$$W_{ij} = \frac{1}{N} \sum_{k=1}^{P} \xi_i^k \xi_j^k \quad . \tag{2.8}$$

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.

2.6 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

$$\begin{aligned} \mu_i \quad &= \sum_{j \neq i}^{N} W_{ij} S_j \qquad (2.9) \\ &= \frac{1}{N} \sum_{k=1}^{P} \sum_{j \neq i}^{N} \xi_i^k \xi_j^k S_j . \end{aligned}$$

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

$$\begin{aligned} \mu_i \quad &= \frac{1}{N} \sum_{k=1}^{P} \sum_{j \neq i}^{N} \xi_i^k \xi_j^k \xi_j^1 \qquad (2.10) \\ &= \frac{1}{N} \sum_{k=1}^{P} \xi_i^k \sum_{j \neq i}^{N} \xi_j^k \xi_j^1 \\ &= \frac{1}{N} \, \xi_i^1 \sum_{j \neq i}^{N} \xi_j^1 \xi_j^1 + \frac{1}{N} \sum_{k \neq 1}^{P} \xi_i^k \sum_{j \neq i}^{N} \xi_j^k \xi_j^1 \\ &= \frac{N-1}{N} \xi_j^1 + \frac{1}{N} \sum_{k \neq 1}^{P} \xi_i^k \sum_{j \neq i}^{N} \xi_j^k \xi_j^1 \end{aligned}$$

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

$$< \mu_i > \;\; \simeq \;\; \xi_i^1 \tag{2.11}$$

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

$$\begin{aligned}\sigma &= \frac{1}{N}\cdot\sqrt{P-1}\cdot\sqrt{N-1} \\ &\simeq \sqrt{\frac{P}{N}}.\end{aligned} \tag{2.12}$$

More laboriously,

$$\begin{aligned}\sigma^2 &= \frac{1}{N}\sum_{i=i}^{N}\left(\frac{1}{N}\sum_{k\neq 1}^{P}\xi_i^k\sum_{j\neq i}^{N}\xi_j^k\xi_j^1\right)\left(\frac{1}{N}\sum_{k'\neq 1}^{P}\xi_i^{k'}\sum_{j'\neq i}^{N}\xi_{j'}^{k'}\xi_{j'}^1\right) \\ &= \frac{1}{N^3}\sum_{k\neq 1}^{P}\sum_{k'\neq 1}^{P}\left(\sum_{i=i}^{N}\xi_i^k\xi_i^{k'}\right)\sum_{j\neq i}^{N}\xi_j^k\xi_j^1\sum_{j'\neq i}^{N}\xi_{j'}^{k'}\xi_{j'}^1 \\ &\xrightarrow[N\to\infty]{} \frac{1}{N^3}\sum_{k\neq 1}^{P}\sum_{k'\neq 1}^{P}\ N\ \delta(k-k')\ \sum_{j\neq i}^{N}\xi_j^k\xi_j^1\sum_{j'\neq i}^{N}\xi_{j'}^{k'}\xi_{j'}^1 \\ &\xrightarrow[N\to\infty]{} \frac{1}{N^2}\sum_{k\neq 1}^{P}\ \sum_{j\neq i}^{N}\xi_j^k\xi_j^1\sum_{j'\neq i}^{N}\xi_{j'}^{k}\xi_{j'}^1 \\ &\xrightarrow[N\to\infty]{} \frac{1}{N^2}\sum_{j\neq i}^{N}\xi_j^1\sum_{j'\neq i}^{N}\xi_{j'}^1\left(\sum_{k\neq 1}^{P}\ \xi_j^k\xi_{j'}^k\right) \\ &\xrightarrow[N\to\infty;\ P\to\infty]{} \frac{1}{N^2}\sum_{j\neq i}^{N}\xi_j^1\sum_{j'\neq i}^{N}\xi_{j'}^1\ (P-1)\ \delta(j-j') \\ &\xrightarrow[N\to\infty;\ P\to\infty]{} \frac{P-1}{N^2}\sum_{j\neq i}^{N}\left(\xi_j^1\right)^2 \\ &\xrightarrow[N\to\infty;\ P\to\infty]{} \frac{(P-1)(N-1)}{N^2} \\ &\xrightarrow[N\to\infty;\ P\to\infty]{} \frac{P}{N}\end{aligned} \tag{2.13}$$

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

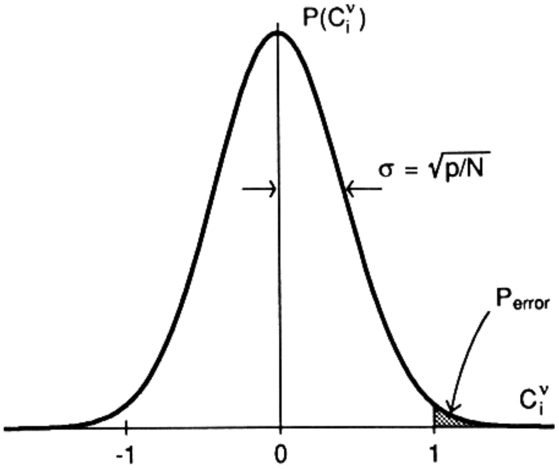
$$p_{error} = \frac{1}{\sqrt{2\pi}\ \sigma}\ \left[\ \int_{-\infty}^{-1} e^{-x^2/2\sigma^2}\ dx\ + \int_{+1}^{\infty} e^{-x^2/2\sigma^2}\ dx\ \right] \tag{2.14}$$

$$
\begin{aligned}
&= \frac{\sqrt{2}}{\sqrt{\pi}\ \sigma} \int_{+1}^{\infty} e^{-x^2/2\sigma^2}\ dx \\
&= \frac{2}{\sqrt{\pi}} \int_{\frac{1}{\sqrt{2}\sigma}}^{\infty} e^{-x^2}\ dx \\
&\equiv \operatorname{erfc}\left(\frac{1}{\sqrt{2}\sigma}\right)
\end{aligned}
$$

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

$$p_{\text{error}} = \operatorname{erfc}\left(\sqrt{\frac{N}{2P}}\right). \tag{2.15}$$

Figure 10: We compute the probability in the tail of the Gaussian. From Hertz, Krogh and Palmer 1991.



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

$$p_{\text{error}} \simeq \frac{2}{\sqrt{\pi}} \frac{P}{N}\ e^{-N/2P} \tag{2.16}$$

so that to leading order

$$log\{p_{\text{error}}\} \simeq -\frac{N}{2P} - \log\{\frac{N}{P}\}. \tag{2.17}$$

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

$$1 - p_{\text{error}} \geq 1 - \frac{1}{NP} \tag{2.18}$$

or

$$\log\{p_{\text{error}}\} < -\log\{NP\}. \tag{2.19}$$

Thus

$$-\frac{N}{2P} - \log\{\frac{N}{P}\} \ < - \log\{NP\} \tag{2.20}$$

or

$$-\frac{N}{2P} < -2\log\{P\} \tag{2.21}$$

so

$$P \ < \ \frac{1}{4}\,\frac{N}{\log\{P\}}. \tag{2.22}$$

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

$$P \ < \ \frac{1}{4}\,\frac{N}{\log\{N\}}. \tag{2.23}$$

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.

This is a worst case analysis that holds in the limit of $N \to \infty$. More typically we want to store states with a fixed, nonzero albeit small error rate. We will explore this possibility next and see if the scaling among P and N changes.