The Hopfield Model

Emin Orhan
eorhan@cns.nyu.edu

February 4, 2014

In this note, I review some basic properties of the Hopfield model. I closely follow Chapter 2 of Herz, Krogh & Palmer (1991) which is an excellent introductory textbook on the theory of neural networks. I motivate the mean field analysis of the stochastic Hopfield model slightly differently than Herz, Krogh & Palmer (1991) and my derivations are a little longer, filling in some of the gaps in the original text, to make them more accessible to the beginners.

1 Deterministic dynamics

The Hopfield model consists of N binary variables or bits, $S_i \in \{+1, -1\}$. These binary variables will be called the units of the network. In the deterministic version of the model (we will later incorporate noise or stochasticity into the model), the units are updated according to:

$$S_i = \text{sign}(\sum_j W_{ij} S_j) \tag{1}$$

where W_{ij} is the weight of the connection between S_i and S_j and sign$(\cdot)$ is the sign function which equals $+1$ if its argument is ≥ 0 and -1 otherwise. The dynamic update described in Equation 1 can be done either synchronously (at each clock cycle updating all S_i simultaneously according to Equation 1), asynchronously (choosing either a random or fixed update order and updating S_i according to that order at each clock cycle) or using a combination of synchronous and asychronous updates. Different update rules can have different consequences for the behavior of the model. In what follows, we will stick to the asynchronous update rule which seems biologically more plausible.

The basic problem setup is that we want to store a number of patterns ξ^μ $(\mu = 1, \ldots, P)$ in the connection matrix of the network. By "storing a pattern", we mean that the pattern should be a stable fixed point of the network, i.e. when the network is initialized with ξ^μ and the units are updated according to Equation 1, the network should stay in the same state. In addition, when the network is initialized with a configuration close to ξ^μ, it should eventually converge to ξ^μ through the update dynamics.

Storing a single pattern

We first consider the case of a single pattern to be stored in the weight matrix of the network. Let us denote this pattern by $\xi = [\xi_1, \ldots, \xi_N]$. From Equation 1, the condition for ξ to be a fixed point of the network dynamics is:

$$\xi_i = \text{sign}(\sum_j W_{ij} \xi_j) \quad \forall i \in \{1, \ldots, N\} \tag{2}$$

So, we need to find a weight matrix W such that Equation 2 is satisfied. It is easy to see that this condition is satisfied if $W_{ij} \propto \xi_i \xi_j$, because then the argument of sign$(\cdot)$ becomes proportional to ξ_i which has the same sign as ξ_i. The constant of proportionality is generally taken to be $1/N$. Therefore, $W_{ij} = \frac{1}{N}\xi_i \xi_j$.

If the network is initialized with a pattern $S = [S_1, \ldots, S_N]$ close (but not identical) to ξ, the network will still converge to ξ. This is because the sum $\sum_j W_{ij} S_j$ inside the sign function will still be dominated by the bits that are identical in S and ξ and will not be affected much by a few misaligned bits in S. Hence, the inside of the sign function will still have the same sign as ξ, attracting the network state to ξ. Therefore, ξ is a *stable* fixed point of the network dynamics, as desired.

Also, note that if ξ is a stable fixed point of the network, so is $-\xi$ (where all bits are flipped with respect to ξ). This is again easy to verify from Equation 2. In fact, all initial configurations with more than half the bits different from ξ will end up in the reverse attractor $-\xi$.

Storing multiple patterns

Now, suppose we want to store multiple patterns ξ^μ $(\mu = 1, \ldots, P)$ in the connection matrix of the network. How should we design the connection matrix W so that all of these patterns are stable fixed points of the network? A straightforward generalization of the prescription suggested above for storing a single pattern in the weight matrix leads to the following expression:

$$W_{ij} = \frac{1}{N} \sum_{\mu=1}^{P} \xi_i^\mu \xi_j^\mu \tag{3}$$

This is also called the Hebbian prescription. Let us first check that this prescription works as desired, that is, the patterns ξ^μ are stable fixed points of the network. The condition for a particular pattern ξ^μ to be a fixed point of the network dynamics is given by:

$$\xi_i^\mu = \text{sign}(\sum_j W_{ij} \xi_j^\mu) = \text{sign}(h_i^\mu) \quad \forall i \in \{1, \ldots, N\} \tag{4}$$

where we just called the sum inside the sign function h_i^μ. We first note that:

$$h_i^\mu = \sum_j W_{ij} \xi_j^\mu = \frac{1}{N} \sum_j \sum_\nu \xi_i^\nu \xi_j^\nu \xi_j^\mu \tag{5}$$

where we plugged in the Hebbian prescription (Equation 3) for W_{ij}. We now separate h_i^μ into two terms corresponding to the pattern μ and all the other ones.

$$\begin{aligned} h_i^\mu &= \frac{1}{N} \sum_j \xi_i^\mu \xi_j^\mu \xi_j^\mu + \frac{1}{N} \sum_j \sum_{\nu \neq \mu} \xi_i^\nu \xi_j^\nu \xi_j^\mu & (6) \\ &= \frac{1}{N}(N \xi_i^\mu) + \frac{1}{N} \sum_j \sum_{\nu \neq \mu} \xi_i^\nu \xi_j^\nu \xi_j^\mu & (7) \\ &= \xi_i^\mu + \gamma & (8) \end{aligned}$$

where γ is called the crosstalk term. The crucial idea is that if $|\gamma|$ is smaller than 1, it will not affect the sign of h_i^μ and hence we can conclude that ξ^μ is a fixed point of the network dynamics. It turns out that $|\gamma|$ ***is*** smaller than 1, if P is small and N is large. To understand why this is the case, note that γ is a random variable that is equal to $1/N$ times the sum of about NP random variables that are equally likely to be $+1$ or -1. Thus, γ has a distribution with mean 0 and variance P/N. To see this, note that a single binary random variable that is equally likely to be $+1$ or -1 has a variance of $\frac{1}{2}(+1-0)^2 + \frac{1}{2}(-1-0)^2 = 1$.

γ is equal to $1/N$ times the sum of about NP such binary random variables. Since these binary variables are pairwise uncorrelated (why?), variances add up and then multiplying by $1/N$ has the effect of scaling the variance by $1/N^2$ resulting in a variance of $NP/N^2 = P/N$ for γ. Therefore, as long as the ratio of the number of stored patterns to the number of units in the network is small enough, $|\gamma|$ will be highly unlikely to be larger than 1.

This argument demonstrates that the patterns ξ^μ are fixed points of the network dynamics as long as P is small compared to N. As in the case of storing a single pattern, if the network is initialized with a pattern $S = [S_1, \ldots, S_N]$ that is close to one of the stored patterns ξ^μ, the sum $\sum_j W_{ij} S_j$ will be dominated by the bits that are identical in S and ξ^μ and thus the network will be quickly pulled toward the pattern ξ^μ. Therefore, the stored patterns ξ^μ are not only fixed points of the network dynamics, but *stable* fixed points, as desired.

Spurious attractors

We have shown that the desired patterns ξ^μ are stable fixed points of the network dynamics. But are they the only stable fixed points or are there other stable fixed points as well? It turns out that there are in fact many other stable fixed points than the patterns ξ^μ. We have previously seen that if ξ^μ is a stable fixed point of the network dynamics, so is $-\xi^\mu$. These reverse attractor states may be considered innocuous, but there are less trivial stable fixed points as well. We now show that mixtures of an odd number of stored patterns are also stable fixed points of the network. For concreteness, we consider mixtures of three patterns, but the same argument applies to any mixture of an odd number of patterns. Specifically, we show that patterns of the form:

$$\xi_i^{mix} = \text{sign}(\pm\xi_i^{\mu_1} \pm \xi_i^{\mu_2} \pm \xi_i^{\mu_3}) \tag{9}$$

are stable fixed points of the network. Again, the argument applies for all sign combinations in Equation 9, but for concreteness, let us consider the case where all signs are +. When the network is initialized with this mixture pattern, the total input for the i-th unit can be written as follows:

$$h_i^{mix} = \frac{1}{N} \sum_j W_{ij} \xi_j^{mix} = \frac{1}{N} \sum_j \sum_\mu \xi_i^\mu \xi_j^\mu \xi_j^{mix} \tag{10}$$

Now pulling out the terms corresponding to the component patterns μ_1, μ_2 and μ_3, we get:

$$h_i^{mix} = \frac{1}{N} \sum_j \xi_i^{\mu_1} \xi_j^{\mu_1} \xi_j^{mix} + \frac{1}{N} \sum_j \xi_i^{\mu_2} \xi_j^{\mu_2} \xi_j^{mix} + \frac{1}{N} \sum_j \xi_i^{\mu_3} \xi_j^{\mu_3} \xi_j^{mix} + \text{crosstalk term} \tag{11}$$

On average, $\xi_j^{\mu_1} \xi_j^{mix}$ has a value of $1/2$. This is because with probability $3/4$, $\xi_j^{\mu_1}$ and ξ_j^{mix} have the same sign ($\xi_j^{\mu_1}$ and ξ_j^{mix} have opposite signs only when both $\xi_j^{\mu_2}$ and $\xi_j^{\mu_3}$ have the opposite sign to that of $\xi_j^{\mu_1}$), in which case $\xi_j^{\mu_1} \xi_j^{mix}$ evaluates to $+1$, and with probability $1/4$ they have opposite signs, in which case $\xi_j^{\mu_1} \xi_j^{mix}$ evaluates to -1. Thus, on average $\xi_j^{\mu_1} \xi_j^{mix}$ has a value of $3/4 - 1/4 = 1/2$. Similarly for $\xi_j^{\mu_2} \xi_j^{mix}$ and $\xi_j^{\mu_3} \xi_j^{mix}$. Inserting these average values in Equation 11, we get:

$$h_i^{mix} = \frac{1}{2} \xi_i^{\mu_1} + \frac{1}{2} \xi_i^{\mu_2} + \frac{1}{2} \xi_i^{\mu_3} + \text{crosstalk term} \tag{12}$$

Again, the crosstalk term can be shown to be small. Thus, h_i^{max} has the same sign as ξ_i^{mix}. Similarly, if the network is initialized in a state close to ξ^{mix}, the aligned bits will dominate in the total input and the network will quickly converge to the mixture pattern ξ^{mix}. This establishes that the pattern ξ^{mix} will be a stable fixed point of the network.

Storage capacity (small P, deterministic dynamics)

Consider the quantity:

$$C_i^\mu = -\xi_i^\mu \frac{1}{N} \sum_j \sum_{\nu \neq \mu} \xi_i^\nu \xi_j^\nu \xi_j^\mu \tag{13}$$

which is just minus the i-th bit in the μ-th pattern times the crosstalk term. If C_i^μ is smaller than 1, from Equation 7, the crosstalk term cannot change the sign of h_i^μ and thus ξ_i^μ is stable. If C_i^μ is larger than 1, the crosstalk term has the opposite sign to that of ξ_i^μ and again from Equation 7, ξ_i^μ is unstable. Thus the probability that any bit i is unstable is given by:

$$P_{error} = P(C_i^\mu > 1) \tag{14}$$

C_i^μ is $1/N$ times the sum of about NP independent random binary variables. Thus it has a distribution with mean 0 and variance $\sigma^2 = P/N$. For large NP, this distribution can be approximated with a Gaussian distribution with the same mean and variance. Thus:

$$P_{error} = P(C_i^\mu > 1) = \frac{1}{\sqrt{2\pi}\sigma} \int_1^\infty \exp(-\frac{x^2}{2\sigma^2}) dx \tag{15}$$

$$= \frac{1}{2}[1 - \mathrm{erf}(\frac{1}{\sqrt{2\sigma^2}})] = \frac{1}{2}[1 - \mathrm{erf}(\sqrt{N/2P})] \tag{16}$$

This implies, for instance, that if we want $P_{error} < 0.01$, P cannot be larger than about $0.185N$.

The above result gives the single-bit error probability. Since there are N bits in a stored pattern, the probability of error-free recall of a stored pattern is given by $(1 - P_{error})^N$. We may require that this probability be greater than some set value, say, 0.99, i.e. $(1 - P_{error})^N > 0.99$. Using the binomial expansion of the left-hand side and keeping the two lowest-order terms with respect to P_{error} (because P_{error} is small), we get $P_{error} < 0.01/N$. Thus, $P_{error} \to 0$ as $N \to \infty$. From Equation 16, this implies $P/N \to 0$ as $N \to \infty$. Therefore, we can use the following asymptotic approximation for the erf function in Equation 16:

$$1 - \mathrm{erf}(x) \to \exp(-x^2)/\sqrt{\pi}x \quad as\ x \to \infty \tag{17}$$

This yields:

$$\log P_{error} \approx -\log 2 - N/2P - \frac{1}{2}\log \pi - \frac{1}{2}\log(N/2P) \tag{18}$$

If we now require $P_{error} < 0.01/N$:

$$-\log 2 - N/2P - \frac{1}{2}\log(\pi) - \frac{1}{2}\log(N/2P) < \log 0.01 - \log N \tag{19}$$

Keeping only terms of leading order in N:

$$N/2P > \log N \tag{20}$$

2 Stochastic (Glauber) dynamics

So far, we have assumed that the units are updated deterministically according to the update rule in Equation 1. This means that a unit i will produce the same response S_i given the same set of inputs, S_j, to the unit. In reality, neurons are noisy devices, so even when the same set of inputs are presented to the neuron, it is not guaranteed to produce the same response. To capture this important characteristic of neurons, we now assume that their responses are stochastic or noisy. In particular, we assume that the

output of the unit i is determined according to the following probabilistic update rule:

$$P(S_i = \pm 1 | h_i) = f_\beta(\pm h_i) = \frac{1}{1 + \exp(\mp 2\beta h_i)} \tag{21}$$

where we denote the total input to unit i with h_i:

$$h_i = \sum_j W_{ij} S_j \tag{22}$$

In Equation 21, $\beta = 1/T$ is an "inverse temperature" parameter that determines the noise level of the unit. Large β values correspond to low noise (or low temperature) and thus the unit behaves more like a deterministic unit as in Equation 21 for large β, whereas smaller β values correspond to higher noise (high temperature) and thus the unit behaves more randomly for smaller β values (for $\beta = 0$, the unit's response becomes independent of its input and thus its output is completely random with $P(S_i = +1) = P(S_i = -1) = 0.5$).

One should think of Equation 21 as defining conditional probability distributions $p(S_i|S_{-i})$ (where $-i$ denotes all indices other than i). The conditional distributions collectively determine the joint distribution over the units, $p(S_1, S_2, \ldots, S_N)$. Once we know this joint distribution, we know everything about the system (at least, as far as the equilibrium properties are concerned). The joint distribution corresponding to the conditional distributions in Equation 21 can be written down explicitly:

$$p(S_1, S_2, \ldots, S_N) = \frac{1}{Z} \exp(\frac{\beta}{2} \sum_{i,j} W_{ij} S_i S_j) \tag{23}$$

where Z is the normalization constant (called the partition function in statistical mechanics). It is easy to check that this joint distribution gives rise to the conditional distributions in Equation 21 (check this for yourself!). Notice that the variables S_i are coupled in the joint distribution, hence there are statistical dependencies between the variables (assuming of course that the weight matrix W is not diagonal).

The problem with the joint distribution in Equation 23 is that it is difficult to work with. For example, suppose we want to find the average configuration $\langle S_1, S_2, \ldots, S_N \rangle$. This would require taking a sum over 2^N terms.

Mean field theory

In the mean field approximation, dependencies between the stochastic variables S_i are ignored. S_i are treated as independent variables and one tries to find the best factorized approximation to the joint distribution in Equation 23. One might think that whatever interesting behavior this model exhibits arises due to the interactions between the units. So, treating them as independent might seem like a severe approximation, but sometimes it works, that is, it gives a qualitatively accurate description of the model's behavior. The exact conditions under which mean field theory "works" are rather complicated, but for Hopfield-type models mean field theory becomes exact for infinite range interactions between units, i.e. when there are infinite products of the form $S_i S_j S_k \ldots$ in the joint distribution (in Equation 23, there are only pairwise interactions between the units).

Mean field approximation to the joint distribution in Equation 23 can be derived as follows. Let's first express each variable S_i as a sum of its mean value plus some fluctuations around this mean: $S_i = m_i + \delta S_i$ where $m_i = \langle S_i \rangle$ is the mean and δS_i is the fluctuation around the mean. Note that m_i is not a random

variable any more. Equation 23 then becomes:

$$p(S_1, S_2, \ldots, S_N) = \frac{1}{Z}\exp(\frac{\beta}{2}\sum_{i,j} W_{ij}(m_i + \delta S_i)(m_j + \delta S_j)) \quad (24)$$

$$= \frac{1}{Z}\exp(\frac{\beta}{2}\sum_{i,j} W_{ij}(m_i m_j + m_i \delta S_j + m_j \delta S_i + \delta S_i \delta S_j)) \quad (25)$$

$$\approx \frac{1}{Z^*}\exp(\frac{\beta}{2}\sum_{i,j} W_{ij}(m_i \delta S_j + m_j \delta S_i)) \quad (26)$$

where in the last step, we absorbed the $m_i m_j$ terms into the new normalization constant Z^* (because they are constant with respect to S_is) and we also ignored the second order terms $\delta S_i \delta S_j$ which is the crucial step in the mean field approximation. The joint distribution now becomes separable in S_i. Equation 26 can be written as:

$$p(S_1, S_2, \ldots, S_N) = \frac{1}{Z^*}\exp(\frac{\beta}{2}\sum_{i,j} W_{ij}(m_i(S_j - m_j) + m_j(S_i - m_i))) \quad (27)$$

$$= \frac{1}{Z^{**}}\exp(\frac{\beta}{2}\sum_{i,j} W_{ij}(m_i S_j + m_j S_i)) \quad (28)$$

$$= \frac{1}{Z^{**}}\exp(\beta\sum_j S_j \sum_i W_{ij} m_i) \quad (29)$$

$$\propto \prod_{j=1}^{N}\exp(\beta S_j \sum_i W_{ij} m_i) \quad (30)$$

Thus, the variables are now decoupled. In order to find the mean configuration $\langle S_1, S_2, \ldots, S_N\rangle$, it is enough to calculate the means of individual variables separately. For a single variable S_i, we find:

$$\langle S_j\rangle = \frac{\exp(\beta\sum_i W_{ij} m_i) - \exp(-\beta\sum_i W_{ij} m_i)}{\exp(\beta\sum_i W_{ij} m_i) + \exp(-\beta\sum_i W_{ij} m_i)} = \tanh(\beta\sum_i W_{ij} m_i) = \tanh(\beta\sum_i W_{ij}\langle S_i\rangle) \quad (31)$$

When the connection weights W_{ij} are set according to the Hebbian prescription (Equation 3), Equation 31 becomes:

$$\langle S_j\rangle = \tanh(\frac{\beta}{N}\sum_{\mu}^{P}\sum_{i}^{N}\xi_j^\mu \xi_i^\mu \langle S_i\rangle) \quad (32)$$

This simplifies things a bit, but Equation 32 is still a set of N coupled nonlinear equations. To make progress in solving it, we make the ansatz (physicists' term for an initial guess) that $\langle S_i\rangle$ is proportional to one of the stored patterns:

$$\langle S_i\rangle = m\xi_i^\nu \quad (33)$$

Plugging this ansatz in Equation 32, we get:

$$m\xi_j^\nu = \tanh(\frac{m\beta}{N}\sum_{\mu}^{P}\sum_{i}^{N}\xi_j^\mu \xi_i^\mu \xi_i^\nu) \quad (34)$$

$$= \tanh(\frac{m\beta}{N}(N\xi_j^\nu + \sum_{\mu\neq\nu}^{P}\sum_{i}^{N}\xi_j^\mu \xi_i^\mu \xi_i^\nu)) \quad (35)$$

$$\approx \tanh(m\beta\xi_i^\nu) \quad (36)$$

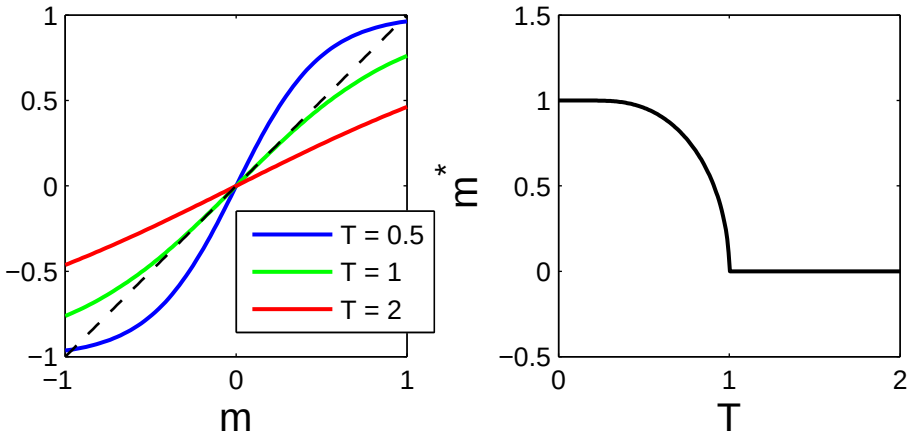

Figure 1: Solution of Equation 37. The dashed line is the left hand side of Equation 37 (identity), the solid lines are the right hand side of Equation 37 for different T values. The solutions of the equation are given by the intersection points of the dashed line with the solid lines. The plot on the right shows the non-negative solutions of Equation 37 as a function of T.

In the last step, we used the fact that the cross-talk term (the second term inside the tanh function) is small as long as $P \ll N$. Because tanh is an odd function, Equation 36 implies:

$$m \approx \tanh(m\beta) \tag{37}$$

Equation 37 tells us that the system undergoes a phase transition at $T = 1/\beta = 1$. For $T > 1$, the only solution of Equation 37 is $m^* = 0$, hence the system is useless as a memory device. For $T < 1$, there is a pair of symmetric solutions as shown in Figure 1.

Storage capacity (large P)

In Equation 36, we had to assume $P \ll N$ to make the cross-talk term negligible. If this assumption does not hold, the cross-talk term is no longer negligible and a more detailed analysis of its effect is required. To this end, let us first define:

$$m_\nu = \frac{1}{N} \sum_i \xi_i^\nu \langle S_i \rangle \tag{38}$$

which is the overlap between the mean state of the network and the ν-th pattern. If we plug in the mean-field expression for $\langle S_i \rangle$ from Equation 32, we get:

$$m_\nu = \frac{1}{N} \sum_i \xi_i^\nu \tanh(\beta \sum_\mu \xi_i^\mu m_\mu) \tag{39}$$

Without loss of generality, suppose that we are interested in the retrieval of the first pattern. We can then separate out the terms corresponding to the first pattern and the ν-th pattern from the sum inside

the tanh function:

$$m_\nu = \frac{1}{N}\sum_i \xi_i^\nu \tanh[\beta(\xi_i^1 m_1 + \xi_i^\nu m_\nu + \sum_{\mu\neq 1,\nu} \xi_i^\mu m_\mu)] \quad (40)$$

$$= \frac{1}{N}\sum_i \xi_i^\nu \xi_i^1 \tanh[\beta(\xi_i^1 \xi_i^1 m_1 + \xi_i^\nu \xi_i^1 m_\nu + \sum_{\mu\neq 1,\nu} \xi_i^\mu \xi_i^1 m_\mu)] \quad (41)$$

$$= \frac{1}{N}\sum_i \xi_i^\nu \xi_i^1 \tanh[\beta(m_1 + \xi_i^\nu \xi_i^1 m_\nu + \sum_{\mu\neq 1,\nu} \xi_i^\mu \xi_i^1 m_\mu)] \quad (42)$$

where in the second step we used the fact that $\tanh(x) = \xi_i^1 \tanh(\xi_i^1 x)$. Now we make a Taylor expansion of Equation 42 around $A = \beta(m_1 + \sum_{\mu\neq 1,\nu} \xi_i^\mu \xi_i^1 m_\mu)$ keeping only terms of order up to (and including) 1:

$$m_\nu = \frac{1}{N}\sum_i \xi_i^\nu \xi_i^1 \tanh(A) + \frac{\beta}{N}\sum_i \{1 - \tanh^2(A)\} m_\nu \quad (43)$$

where we used the facts that $\tanh' = 1 - \tanh^2$ and $\beta(m_1 + \xi_i^\nu \xi_i^1 m_\nu + \sum_{\mu\neq 1,\nu} \xi_i^\mu \xi_i^1 m_\mu) - A = \beta \xi_i^\nu \xi_i^1 m_\nu$. Note that the Taylor expansion is justified because $\beta \xi_i^\nu \xi_i^1 m_\nu$ is small.

Now we assume that the overlaps m_μ for $\mu \neq 1, \nu$ are small Gaussian random variables with zero mean and variance $\alpha r/P$ where $\alpha = P/N$ is the memory load and $r = \frac{1}{\alpha}\sum_{\nu\neq 1} m_\nu^2$. Then the sum $\sum_{\mu\neq 1,\nu} \xi_i^\mu \xi_i^1 m_\mu$ in A is Gaussian with zero mean and variance αr and thus the average over i in Equation 43 is an average over Gaussian noise with zero mean and variance αr. Thus, Equation 43 becomes:

$$m_\nu = \frac{1}{N}\sum_i \xi_i^\nu \xi_i^1 \tanh(A) + \beta m_\nu - \beta q m_\nu \quad (44)$$

$$= \frac{N^{-1}\sum_i \xi_i^\nu \xi_i^1 \tanh(A)}{1 - \beta(1-q)} \quad (45)$$

where q is the average of $\tanh^2(A)$ over Gaussian noise with zero mean and variance αr:

$$q = \int \frac{dz}{\sqrt{2\pi}} \exp(-\frac{z^2}{2}) \tanh^2[\beta(m_1 + \sqrt{\alpha r} z)] \quad (46)$$

Now we need self-consistent equations for r and m_1 so that together with q we have 3 equations and 3 unknowns. From the definition of r (i.e. $r = \frac{1}{\alpha}\sum_{\nu\neq 1} m_\nu^2$), we can obtain an equation for r by taking the square of Equation 45:

$$m_\nu^2 = [\frac{1}{1-\beta(1-q)}]^2 \frac{1}{N^2}\sum_{i,j} \xi_i^\nu \xi_i^1 \xi_j^\nu \xi_j^1 \tanh[\beta(m_1 + \sum_{\mu\neq 1,\nu} \xi_i^\mu \xi_i^1 m_\mu)] \tanh[\beta(m_1 + \sum_{\mu\neq 1,\nu} \xi_j^\mu \xi_j^1 m_\mu)] \quad (47)$$

and averaging over the patterns ν:

$$
\begin{aligned}
r &= [\frac{1}{1-\beta(1-q)}]^2 \frac{1}{N^2} \sum_{i,j} \{\frac{N}{P} \sum_{\nu}^{P} \xi_i^\nu \xi_i^1 \xi_j^\nu \xi_j^1\} \tanh[\beta(m_1 + \sum_{\mu\neq 1,\nu} \xi_i^\mu \xi_i^1 m_\mu)] \tanh[\beta(m_1 + \sum_{\mu\neq 1,\nu} \xi_j^\mu \xi_j^1 m_\mu)] \\
&= [\frac{1}{1-\beta(1-q)}]^2 \frac{1}{N} \sum_{i=j} \tanh[\beta(m_1 + \sum_{\mu\neq 1,\nu} \xi_i^\mu \xi_i^1 m_\mu)] \tanh[\beta(m_1 + \sum_{\mu\neq 1,\nu} \xi_j^\mu \xi_j^1 m_\mu)] \quad (48) \\
&= [\frac{1}{1-\beta(1-q)}]^2 \frac{1}{N} \sum_{i} \tanh^2[\beta(m_1 + \sum_{\mu\neq 1,\nu} \xi_i^\mu \xi_i^1 m_\mu)] \quad (49) \\
&= \frac{q}{[1-\beta(1-q)]^2} \quad (50)
\end{aligned}
$$

Finally, to obtain an equation for m_1, we plug in $\nu = 1$ in Equation 39, and separate out the term corresponding to the first pattern inside the tanh function:

$$
\begin{aligned}
m_1 &= \frac{1}{N} \sum_i \xi_i^1 \tanh[\beta(\xi_i^1 m_1 + \sum_{\mu\neq 1} \xi_i^\mu m_\mu)] \quad (51) \\
&= \frac{1}{N} \sum_i \xi_i^1 \xi_i^1 \tanh[\beta(\xi_i^1 \xi_i^1 m_1 + \sum_{\mu\neq 1} \xi_i^1 \xi_i^\mu m_\mu)] \quad (52) \\
&= \frac{1}{N} \sum_i \tanh[\beta(m_1 + \sum_{\mu\neq 1} \xi_i^1 \xi_i^\mu m_\mu)] \quad (53) \\
&= \int \frac{dz}{\sqrt{2\pi}} \exp(-\frac{z^2}{2}) \tanh[\beta(m_1 + \sqrt{\alpha r} z)] \quad (54)
\end{aligned}
$$

Equations 46, 50 and 54 (in magenta) can be solved simultaneously for q, r and m_1. In general, this has to be done numerically, but in some special cases, e.g. in the limit of low temperature ($T \to 0$), we can make further progress toward an analytical solution.

Let us investigate the low temperature (or low noise) limit, i.e. $T \to 0$ ($\beta \to \infty$), more closely. In this limit, we can use the following approximations:

$$
\begin{aligned}
\int \frac{dz}{\sqrt{2\pi}} \exp(-\frac{z^2}{2})(1 - \tanh^2[\beta(az+b)]) &\approx \frac{1}{\sqrt{2\pi}} \exp(-\frac{z^2}{2})|_{\tanh^2[\beta(az+b)]=0} \int dz (1 - \tanh^2[\beta(az+b)]) \\
&= \frac{1}{\sqrt{2\pi}} \exp(-\frac{b^2}{2a^2}) \frac{1}{a\beta} \int dz \frac{d}{dz} \tanh[\beta(az+b)] \quad (55) \\
&= \sqrt{\frac{2}{\pi}} \exp(-\frac{b^2}{2a^2}) \frac{1}{a\beta} \quad (56)
\end{aligned}
$$

and

$$\int \frac{dz}{\sqrt{2\pi}} \exp(-\frac{z^2}{2}) \tanh[\beta(az+b)] \approx \int \frac{dz}{\sqrt{2\pi}} \exp(-\frac{z^2}{2}) \text{sign}[\beta(az+b)] \quad (57)$$

$$= \int_{-b/a}^{\infty} \frac{dz}{\sqrt{2\pi}} \exp(-\frac{z^2}{2}) - \int_{-\infty}^{-b/a} \frac{dz}{\sqrt{2\pi}} \exp(-\frac{z^2}{2}) \quad (58)$$

$$= 2\int_{0}^{b/a} \frac{dz}{\sqrt{2\pi}} \exp(-\frac{z^2}{2}) \quad (59)$$

$$= \text{erf}(\frac{b}{a\sqrt{2}}) \quad (60)$$

Using these approximations in Equations 46, 50 and 54, we get the following three equations in the $T \to 0$ limit:

$$C \equiv \beta(1-q) = \sqrt{\frac{2}{\pi\alpha r}} \exp(-\frac{m_1^2}{2\alpha r}) \quad (61)$$

$$r = \frac{1}{(1-C)^2} \quad (62)$$

$$m_1 = \text{erf}(\frac{m_1}{\sqrt{2\alpha r}}) \quad (63)$$

Substituting $y = m_1/\sqrt{2\alpha r}$, we get:

$$y(\sqrt{2\alpha} + \frac{2}{\sqrt{\pi}} \exp(-y^2)) = \text{erf}(y) \quad (64)$$

where we used the approximation $\sqrt{r} = \frac{1}{1-C} \approx 1 + C$ in the derivation. Equation 64 can be solved graphically. Figure 2 shows the solutions of Equation 64 for different values of the memory load, α. For a given α, the solutions are the intersection points of the solid line (which represents the left hand side of Equation 64) and the dashed line (which represents the right hand side of Equation 64). It can be seen from the figure that there is a critical value of $\alpha = \alpha_c$ between 0.10 and 0.15 such that for $\alpha < \alpha_c$ the system has non-zero solutions, and for $\alpha > \alpha_c$ it has no non-zero solution. This critical value turns out to be $\alpha_c \approx 0.138$ for $T \to 0$. The transition at α_c is a very sharp one: around this point, the solution m_1^* drops abruptly and discontinuously from around 0.97 suggesting a very high-fidelity memory, to 0 meaning a completely useless memory.

For a detailed analysis of the system in the full T-α (temperature-memory load) space, see the two classic papers by Amit, Gutfreund & Sompolinsky (1985a; 1985b).

References

[1] Amit, D.J., Gutfreund, H., & Sompolinsky, H. (1985a). Spin-glass models of neural networks. *Physical Review A*, 32, 1007-1018.

[2] Amit, D.J., Gutfreund, H., & Sompolinsky, H. (1985b). Storing infinite numbers of patterns in a spin-glass model of neural networks. *Physical Review Letters*, 55, 1530-1533.

[3] Hertz, J., Krogh, A., & Palmer, R.G. (1991). *Introduction to the Theory of Neural Computation.* Addison-Wesley.

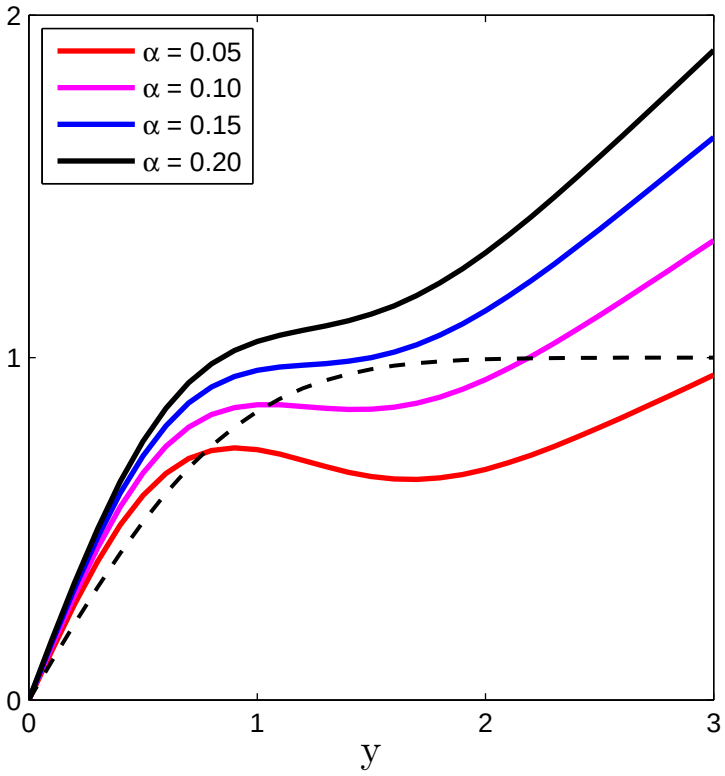

Figure 2: Solutions of Equation 64 for different values of the memory load, α. The dashed line is the right hand side of Equation 64, the solid lines are the right hand side of Equation 64 for different α values. The solutions of the equation are given by the intersection points of the dashed line with the solid lines.