An introduction to stochastic neural networks

H.J. Kappen

SNN University of Nijmegen
Geert Grooteplein Noord 21, 6525 EZ Nijmegen, The Netherlands

bert@mbfys.kun.nl

§1 Introduction

How does the brain compute? Particularly in the last hundred years have we gathered an enormous amount of experimental findings that shed some light on this question. The picture that has emerged is that the neuron is the central computing element of the brain which performs a non-linear input to output mapping between its synaptic inputs and its spiky output. The neurons are connected by synaptic junctions, thus forming a neural network.

A central question is how such a neural network implements brain functions such as vision, audition and motor control. These questions are to a certain extend premature, because our knowledge of the functioning of the neuron and the synaptic process itself is only partial and much remains to be discovered. Nevertheless, it is interesting to see what emergent behavior arises in a network of very simple neurons.

The pioneering work in this direction was done by McCulloch and Pitts [1)] in the '40s. Taking the thresholding property of neurons to the extreme, they proposed that neurons perform logical operations on their inputs, such as AND and OR. One can show that a network of such neurons, when properly wired, can perform any logical function and is equivalent to a Turing machine.

When considering neural networks, an important distinction is between feed-forward networks and recurrent networks. In *feed-forward networks*, the neurons can be labeled such that each neuron only receives input from neurons with lower label. Thus, one can identify *input neurons*, which receive no input from other neurons and whose activity depends only on the sensory stimulus,

and *output neurons* whose output does not affect other neurons. When in addition the neurons themselves are assumed to have no internal dynamics, the dynamics of feed-forward networks is trivial in the sense that the output is a time-independent function of the input: $y(t) = F(x(t))$, where F is a concatenation of the individual neuron transfer functions and x and y are input and output activity, respectively. Examples of such networks are the perceptron [2] and the multi-layered perceptron [3, 4].

In *recurrent networks* one typically defines a subset of neurons as input neurons and another subset as output neurons. Even when individual neurons have no internal dynamics, the network as a whole does, and the input-output mapping depends explicitly on time: $y(t) = F(x(t), t)$, Examples of such networks are attractor neural networks [5], topological maps [6] (see chapter by Flanagan in this book), sequence generators [7] and Boltzmann Machines [8].

Unlike the logical McCullock-Pitts neurons, real neurons are noisy and the output of the neuron is a probabilistic function of its input. The dynamics of a network of such neurons is characterized by transient and stationary behavior. The stationary behavior of the network is obtained for large time when the input to the network is time-independent (or when it is described by a time-independent probability distribution). This behaviour is then described in terms of a time-independent probability distribution over the states of the network. The transient behavior is described by the characteristic time(s) to approach stationarity and by its dependence on initial values.

In section 2 we begin with a very brief description of the behavior of the biological neuron and some properties of the synapses and discuss under which assumptions the description by a probabilistic binary threshold device is appropriate. In section 3 we discuss stochastic neural networks with parallel and sequential dynamics. This dynamics is given by a Markov process, and in section 4 we discuss some of the properties of Markov processes, such as ergodicity and periodicity.

An exact description of transient and stationary behavior for stochastic neural networks is not possible in general. In some special cases, however, one can compute the generic behavior of stochastic networks using mean field theory. One averages over many random instances of the network (quenched average) and describes the properties of the network with a small number of order parameters. The classical example is the attractor neural network, as proposed by Hopfield [5]. The mean field analysis was presented in a series of papers

by Amit, Gutfreund and Sompolinsky [9, 10]. Due to the symmetric connectivity of the Hebb rule, the asymptotic behavior of the network can be computed in closed form. The patterns that are stored with the Hebb rule become stable attractors of the dynamics when the number of patterns is sufficiently small and the noise in the dynamics is sufficiently low. Thus the network operates as a distributed memory. When the noise is too high, all attractors become unstable and the firing of the neurons becomes more or less uncorrelated (paramagnetic phase). When the number of patterns is too large, the network behaves as a *spin glass* whose minima are uncorrelated with the stored patterns. In section 5 we will introduce the quenched average approach for a simpler problem, the Sherrington-Kirkpatrick model. It will show us the generic behavior that can be expected from symmetrically connected neural networks. For a more thorough treatment of this topic see the chapters by Coolen in this volume.

Clearly, biological neural networks do not have symmetric connectivity. For non-symmetric networks the theoretical analysis is much harder and fewer results are known. Most of the results have been obtained with numerical simulations. It appears that when a sufficient amount of asymmetry is introduced, the network dynamics is dominated by periodic orbits of exponential length. Thus asymmetric networks are radically different from symmetric networks. The differences between symmetric and asymmetric networks are discussed in section 6.1.

In many instances we are not satisfied with the generic behavior of networks as given by the quenched average approach, but we would like to say something about one individual network. An example is when we consider learning. It has been well established experimentally, that synapses change their strength as a function of the firing of the pre and post synaptic neuron. In order to compute these changes, one needs estimates of the mean firing rates and the correlations of the pre and post synaptic neuron. In the case of symmetric connectivity, this approach was pioneered by Hinton with the introduction of Boltzmann Machines [8]. Due to the *intractability* of the Boltzmann Machine learning rule, it has not been used widely. In section 6.2 we therefore consider a form of mean field theory that does not involve the quenched average. We derive mean field approximations for the mean firing rates and the correlations for stochastic networks with arbitrary connectivity. A drawback of this approach is that it is only valid for small values of the weights. However, as we will see in section 2, due to their noisiness synapses are expected to be small.

Subsequently, we will discuss learning in stochastic networks in section 7. We briefly discuss Hebbian learning in the attractor neural network, as proposed by Hopfield. Then, we discuss the Boltzmann Machine proposed by Hinton. We show that learning in Boltzmann Machines is intractable and how mean field theory can be applied to obtain fast learning algorithms. We illustrate Boltzmann Machine learning on a digit classification task.

§2 Stochastic binary neurons

The effect of a presynaptic spike on the post-synaptic neuron is a local change in the membrane potential. This change can be either positive or negative and is called the excitatory or inhibitory postsynaptic potential (PSP). The PSP is of the order of 0.05-2 mV [11] and is a stochastic event [12]: it either happens or it does not. The probability of the PSP is experimentally observed anywhere between 0.1 and 0.9 (see [13] and references there) and depends also on recent pre- and post-synaptic cell activity [14, 15].

How these local changes in the membrane potential at synaptic junctions contribute to the spike generation process can be computed by compartmental modeling of the geometry of the cell. The dynamics is a complex spatio-temporal process involving many thousand synaptic inputs which are distributed over the dendrites and soma. A main complicating factor is that such simulations require the setting of many parameter values, many of which are not experimentally accessible. The general picture that emerges is, however, that the local PSP at the synapse propagates to the cell body with a delay of 1-2 msec and shows a temporal dispersion of about 10 msec. In addition, the dendrite acts as a low pass filter. It attenuates the frequency components of the PSP below 50 Hz by a factor of 2-4 depending on the frequency of stimulation and location of the synapse on the dendrite and effectively blocks all high frequency components [16].

In order to study the behavior of networks of neurons we may try to find a more compact description of a neuron which ignores its internal details but retains some of its input-output behavior. Let us define the synaptic response function $W_{ij}(t)$ as the temporal response of a presynaptic spike of neuron j on the membrane potential of the soma of neuron i. This function incorporates the effects of delay, attenuation and dispersion mentioned above. This response occurs with probability p_{ij}.

We describe the activity of neuron j as a train of spikes with each spike

a delta peak:

$$x_j(t) = \sum_k \delta(t - t_k^j),$$

where $t_k^j, k = 1, \ldots$ are the times at which neuron j fires. We assume that the PSPs from different synapses combine linearly and therefore the soma potential is given by

$$\begin{aligned} v_i(t) &= \sum_j \int_{-\infty}^{t} dt' W_{ij}(t - t') x_j(t') \\ &= \sum_{j,k} W_{ij}(t - t_k^j) \end{aligned} \tag{1}$$

This potential is to be compared with the threshold Θ_i. If $v_i(t)$ exceeds the threshold, neuron i emits a spike and is forced to remain quiet during the subsequent refractory period τ_r (2-4 msec).

We approximate the neuron dynamics described above, by assuming that the maximal firing rate of the neurons is lower than $\frac{1}{\tau}$, with $\tau \approx 10$ msec the characteristic width of $W_{ij}(t)$. In this case, the presynaptic neuron j is likely to fire zero or one time and unlikely to fire more than one time in the period $[t - \tau, t]$. Indeed, when the spikes from the presynaptic neuron are given by a Poisson process with mean firing rate f, the probability that it fires exactly k times in the period $[t - \tau, t]$ is given by

$$p_k(\tau) = \frac{(f\tau)^k}{k!} e^{-f\tau}$$

When $f\tau << 1$, it is easy to verify that the probability of the neuron to fire more than one time $\sum_{k=2}^{\infty} p_k(\tau) = \mathcal{O}(f^2\tau^2)$ and will be ignored. We associate the binary variable $y_j(t) = 0, 1$ with the firing of neuron j in the following way:

$$y_j(t) = 1 \Leftrightarrow \text{neuron } j \text{ fires in } [t - \tau, t]$$

We discretize time in chunks of length τ and thus at any time t, the state of a network of n neurons is described by the vector $\mathbf{y}(t) = (y_1(t), \ldots, y_n(t))$. In addition, we assume that $W_{ij}(t)$ is block shaped: $W_{ij}(t) = W_{ij}$ for $0 < t < \tau$ and zero otherwise. Thus, the potential becomes

$$v_i(t) = \sum_j W_{ij} y_j(t).$$

It is well known experimentally, that the PSPs W_{ij} do not give the same response every time the presynaptic neuron fires. In fact, the synaptic processes are very noisy and give largely varying postsynaptic responses [12]. We will therefore consider the W_{ij} as independent stochastic variables. Let $\bar{W}_{ij}$ denote their mean value and $\sigma_{ij}^2 = \frac{1-p_{ij}}{p_{ij}}\bar{W}_{ij}^2$ their variance. Since the membrane potential consists of a typically large sum of PSPs, it becomes a Gaussian variable with mean and variance given by

$$\begin{aligned}\bar{v}_i(t) &= \sum_j \bar{W}_{ij} y_j(t) \\ \sigma_i^2(t) &= \sum_j \sigma_{ij}^2 y_j(t).\end{aligned} \tag{2}$$

The neuron fires, when the post synaptic potential exceeds a threshold Θ_i. Therefore, the probability of a post synaptic spike is given by

$$\begin{aligned}p(y_i(t+\tau)=1|\mathbf{y}(t)) &= \int_{\Theta_i}^{\infty} dv_i \frac{1}{\sqrt{2\pi}\sigma_i}\exp\left(-\frac{(v_i-\bar{v}_i)^2}{2\sigma_i^2}\right) \\ &= \frac{1}{2}\left(1+\mathrm{erf}\left(\frac{\bar{v}_i-\Theta_i}{\sigma_i\sqrt{2}}\right)\right)\end{aligned} \tag{3}$$

In this equation, erf is the error function, defined as

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x dy \exp(-y^2).$$

In our derivation of this stochastic neuron model, we have assumed that each of the synapses participates with a contribution W_{ij} and that the membrane potential is an instantaneous function of the total input (see Eq. 2). In reality, the dynamics is much more complex. The membrane integrates incoming stochastic activity and the time needed to reach the threshold is known as a first passage problem. The analytical solution of the first passage time problem is not known in general. This problem is well approximated by the above treatment when the membrane time constant is small compared to the rate of change of the presynaptic input signal.

Note, that the probability to generate a spike depends on the input activity $\mathbf{y}(t)$ both in the numerator and in the denominator of Eq. 3. The former dependence is well known and states that the probability of firing of the cell is a function of the overlap between the input pattern $\mathbf{y}$ and the vector of synapses, i.e. the mean membrane potential. This dependence is the basis of

coincidence detection: if between $t-\tau$ and t a large enough number of afferent cells fire, each of which has an excitatory connection to cell i, cell i will fire.

The dependence in the denominator is weaker and is usually ignored. σ_i^2 is a sum of random positive quantities and therefore its mean value is of $\mathcal{O}(n)$ and its fluctuations of order $\mathcal{O}(\sqrt{n})$. For large n we can therefore ignore the fluctuations in σ_i^2 so that

$$\sigma_i^2 \approx n\sigma^2 r. \tag{4}$$

Here $\sigma^2 = \frac{1}{n}\sum_j \sigma_{ij}^2$ denotes the mean noise in the synapses and $r = \frac{1}{n}\sum_j \bar{y}_j$ denotes the mean firing rate. erf is an increasing function of its argument, and σ_i affects the slope of this function. We see that this slope decreases with increasing overall firing rate. This effect can be easily understood as follows. When the firing rate increases, the mean membrane potential will not be affected because of the balance of excitatory and inhibitory synapses. However, total noise in the input as given by Eq. 2 increases. This will broaden the distribution of v_i and thus increase (decrease) the probability to fire when $\bar{v}_i$ is less (larger) then Θ_i, respectively. Thus, because the mean membrane potential is usually lower than the threshold, an increase in the overall input firing rate will increase the probability of the cell to fire (without affecting the mean membrane potential).

The effect is illustrated in Fig. 1. We consider a model neuron with n=10000 synaptic inputs. The synaptic strength is uniformly distributed between -1 and 1. The synaptic probability is uniformly distributed between 0 and 1. The threshold is set to zero. For firing rates 0.1, 0.5 and 0.9, respectively, we generated 500 binary input vectors and plot the probability of firing, as given by Eq. 3 versus the mean membrane potential $\bar{v}_i$.

In the case that the neuron only receives excitatory input, the neuron is virtually deterministic and the above effects are absent: The membrane potential is a sum of n positive quantities and therefore of $\mathcal{O}(n)$. The membrane potential will display large fluctuations also of $\mathcal{O}(n)$ due to the stochastic nature of the synapses and the variable input. Therefore, the threshold must also be of $\mathcal{O}(n)$ because otherwise the neuron will be either always firing or always quiet. Therefore, $\bar{v}_i - \Theta_i$ is of $\mathcal{O}(n)$, whereas the denominator is of $\mathcal{O}(\sqrt{n})$. The erf will always be driven to saturation, which makes its output either zero or one and it is insensitive to the particular value in the denominator.

The error function is numerically very similar to the hyperbolic tangent,

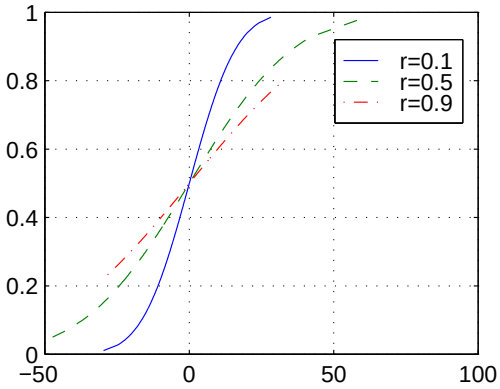

Fig. 1 Spike probability as a function of mean membrane potential for different values of overall firing rate. See main text for details

in the following way: [*1]

$$\mathrm{erf}(x) \approx \tanh(\frac{2x}{\sqrt{\pi}}).$$

In addition we define $s_i = 2y_i - 1 = \pm 1$ to denote whether a neuron is firing or not. The state of the whole network will be simply denoted by $\mathbf{s} = (s_1, \ldots, s_n)$. Thus, we can rewrite Eq. 3 in the following way:

$$p(s_i', t+\tau|\mathbf{s}, t) = \frac{1}{2}\left(1 + \tanh(h_i(\mathbf{s})s_i')\right), \tag{5}$$

with

$$\begin{aligned} h_i &= \sum_{j \neq i} w_{ij} s_j + \theta_i \\ w_{ij} &= \frac{\bar{W}_{ij}}{\sqrt{2\pi n r \sigma}} \\ \theta_i &= \frac{\sum_j \bar{W}_{ij} - 2\Theta_i}{\sqrt{2\pi n r \sigma}} \end{aligned}$$

[*1] The choice of the factor $\frac{2}{\sqrt{\pi}}$ is such that the derivatives of both functions in $x = 0$ are equal and their maximal difference is 0.0352. One can optimize the prefactor such that the maximal difference is minimized. The resulting prefactor is slightly higher and the maximal difference reduces to 0.0189

§3 Stochastic network dynamics

3.1 Parallel dynamics: Little model

Eq. 5 describes the probability for a single neuron to emit a spike between t and $t+\tau$, given an input activity $\mathbf{s}$. In a network of neurons, this equation must be updated in parallel for all neurons. Thus, the transition probability from a state $\mathbf{s}$ at time t to a state $\mathbf{s}'$ at time $t' = t+\tau$ is given by

$$T(\mathbf{s}', t'|\mathbf{s}, t) = \prod_i p(s_i', t+\tau|\mathbf{s}, t) \tag{6}$$

with $p(s_i', t+\tau|\mathbf{s}, t)$ given by Eq. 5. T denotes the probability to observe the network in state $\mathbf{s}'$, given the fact that it was in state $\mathbf{s}$ at the previous time step. Since the dynamics is stochastic, the network will in general not be found in any one state but instead in a superposition of states. Therefore, the fundamental quantity to consider is $p_t(\mathbf{s})$, denoting the probability that the network is in state $\mathbf{s}$ at time t. The dynamics of the network is therefore defined as

$$p_{t+\tau}(\mathbf{s}') = \sum_{\mathbf{s}} T(\mathbf{s}'|\mathbf{s}) p_t(\mathbf{s}). \tag{7}$$

Eq 7 is known as a first order homogeneous Markov process. The first order refers to the fact that the probability of the new state only depends on the current state and not on any past history. Homogeneous means that the transition probability is not an explicit function of time, as can be verified by Eq. 5. This Markov process was first considered by Little [17].

3.2 Sequential dynamics

One of the drawbacks of parallel dynamics is that due to the strict discretization of time in intervals of length τ, an external clock is implicitly assumed which dictates the updating of all the neurons. There exists another stochastic dynamics which has been used extensively in the neural network literature which is called sequential Glauber dynamics. Instead of updating all neuron in parallel, one neuron is selected at random and is updated. The neurobiological motivation that is sometimes given for this dynamics is that neurons are connected with random delays and that the membrane integration time is negligible [18] or that integration times have random duration [19]. The main reason for the popularity of sequential dynamics is that the stationary distribution is a Boltzmann-Gibbs distribution when the connectivity in the network is symmetric. This makes the connection to statistical physics immediate and allows for all the powerfull

machinery of mean field theory to be applied. Also, the parameters (weights and thresholds) in the Boltzmann-Gibbs distribution can be adapted with a learning algorithm which is known as the Boltzmann Machine [8].

The sequential dynamics is defined as follows. At every iteration t, choose a neuron i at random. Update the state of neuron i using Eq. 5. Let $\mathbf{s}$ denote the current state of the network and let F_i denote a flip operator that flips the value of the ith neuron: $\mathbf{s}' = F_i\mathbf{s} \Leftrightarrow s'_i = -s_i$ and $s'_j = s_j$ for all $j \neq i$. Thus, the network can make a transition to state $\mathbf{s}' = F_i\mathbf{s}$ with probability

$$T(\mathbf{s}', t'|\mathbf{s}, t) = \frac{1}{n} p(s'_i, t+\tau|\mathbf{s}, t) \tag{8}$$

with $p(s'_i, t+\tau|\mathbf{s}, t)$ again given by Eq. 5. The factor $\frac{1}{n}$ is a consequence of the random choice of the neurons at each iteration. The probability to remain in state s is given by the equality $\sum_{\mathbf{s}'} T(\mathbf{s}'|\mathbf{s}) = 1$, so that

$$T(\mathbf{s}, t'|\mathbf{s}, t) = 1 - \frac{1}{n}\sum_i p(s'_i, t+\tau|\mathbf{s}, t). \tag{9}$$

Eqs. 8 and 9 together with Eq. 7 define the sequential dynamics. Note, that this dynamics allows only transitions between states $\mathbf{s}$ and $\mathbf{s}'$ that differ at most at one location, whereas the Little model allows transitions between all states.

§4 Some properties of Markov processes

In this section, we review some of the basic properties of first order Markov processes. For a more thorough treatment see [20].

4.1 Eigenvalue spectrum of T

Let $\mathcal{S}$ denote the set of all state vectors $\mathbf{s}$. $\mathbf{s} \in \mathcal{S}$ is a binary vector of length n and thus $\mathbf{s}$ can take on 2^n different values. Therefore, $p_t(\mathbf{s})$ in Eq. 7 is a vector of length 2^n and $T(\mathbf{s}'|\mathbf{s})$ is a $2^n \times 2^n$ matrix. Since $p_t(\mathbf{s})$ denotes a probability vector, it must satisfy $\sum_{\mathbf{s}} p_t(\mathbf{s}) = 1$. In addition, $T(\mathbf{s}'|\mathbf{s})$ is a probability vector in $\mathbf{s}'$ for each value of $\mathbf{s}$ and therefore each column must add up to one:

$$\sum_{\mathbf{s}'} T(\mathbf{s}'|\mathbf{s}) = 1. \tag{10}$$

Matrices with this property are called stochastic matrices.

Let us denote the eigenvalues and left and right eigenvectors of T by $\lambda_\alpha, l_\alpha, r_\alpha, \alpha = 1, \ldots, 2^n$, respectively [*2]. In matrix notation we have

$$\begin{aligned} Tr_\alpha &= \lambda_\alpha r_\alpha \\ l_\alpha^\dagger T &= \lambda_\alpha l_\alpha^\dagger \end{aligned}$$

Since T is a non-symmetric matrix, the left and right eigenvectors are different, non-orthogonal and complex valued. $\dagger$ denotes complex conjugation and transpose. The eigenvalues are complex valued. Under rather general conditions each set of eigenvectors spans a non-orthogonal basis of C^{2^n}. These two bases are dual in the sense that:

$$l_\alpha^\dagger r_\beta = \delta_{\alpha\beta}. \tag{11}$$

δ_{ab} denotes the Kronecker delta: $\delta_{ab} = 1$ if $a = b$ and 0 otherwise. a and b can be real scalars or vectors. We can therefore expand T on the basis of its eigenvectors:

$$T = \sum_{\alpha=1}^{2^n} \lambda_\alpha r_\alpha l_\alpha^\dagger$$

If at $t = 0$ the network is in a state $\mathbf{s}^0$ then $p_0(\mathbf{s}) = p_{t=0}(\mathbf{s}) = \delta_{\mathbf{s},\mathbf{s}^0}$. Let us set $\tau = 1$ for convenience. The probability vector p_t at some later time t is obtained by repeated application of Eq. 7:

$$p_t = T^t p_0 = \sum_\alpha \lambda_\alpha^t r_\alpha (l_\alpha^\dagger p_0) \tag{12}$$

The stationary probability distribution of the stochastic dynamics T is given by p_∞ which is invariant under the operation of T and therefore satisfies

$$Tp_\infty = p_\infty. \tag{13}$$

Thus, the stationary distribution is a right eigenvector of T with eigenvalue 1.

4.2 Ergodicity and ergodicity breaking

A Markov process is called *irreducible*, or *ergodic*, on a subset of states $\mathcal{C} \subset \mathcal{S}$ if for any state $\mathbf{s} \in \mathcal{C}$ there is a finite probability to visit any other state $\mathbf{s}' \in \mathcal{C}$. This means that for any two states $\mathbf{s}, \mathbf{s}' \in \mathcal{C}$, there exists a number k and

[*2] In general, the number of eigenvalues of T can be less than 2^n. However, for our purposes we can ignore this case

a set of intermediate states $\mathbf{s} = \mathbf{s}^0, \mathbf{s}^1, \ldots, \mathbf{s}^k = \mathbf{s}'$ such that $\prod_{i=1}^{k} T(\mathbf{s}^i|\mathbf{s}^{i-1}) > 0$. A subset of states $\mathcal{C} \subset \mathcal{S}$ is called *closed* when the Markov process can never escape from $\mathcal{C}$, once entered: $T(\mathbf{s}'|\mathbf{s}) = 0$ for all $\mathbf{s} \in \mathcal{C}, \mathbf{s}' \neg \in \mathcal{C}$. In general, we can decompose the state space $\mathcal{S}$ uniquely into closed irreducible subsets $\mathcal{S} = \mathcal{T} \cup \mathcal{C}_1 \cup \mathcal{C}_2 \ldots$, where $\mathcal{T}$ is a set of *transient states* and the $\mathcal{C}_i$ are closed irreducible sets.

For an irreducible Markov process of *periodicity* d the Perron-Frobenius theorem states that T has d eigenvalues given by

$$\lambda_m = \exp(2\pi i m/d), m = 0, \ldots, d-1,$$

and all remaining eigenvalues of T are inside the unit circle in the complex plane: $|\lambda_\alpha| < 1$ [*3]. In particular, T has exactly one eigenvalue 1. Its corresponding right eigenvector is equal to the (unique) stationary distribution. Note, that the left eigenvector with eigenvalue 1 is $\propto (1, \ldots, 1)$ as is immediately seen from Eq. 10. The right eigen vector, in contrast, is in general difficult to compute, as will be seen later.

A non-irreducible or non-ergodic Markov process has more than one eigenvalue 1 and therefore more than one left and right eigenvector with eigenvalue 1. Let us denote these eigenvectors by $l_1, \ldots, l_k$ and $r_1, \ldots, r_k$, respectively. Any linear combination of the right eigenvectors

$$p_\infty = \sum_{\alpha=1}^{k} \rho_\alpha r_\alpha \tag{14}$$

is therefore a stationary distribution, assuming proper normalization: $p_\infty(\mathbf{s}) \geq 0$ for all $\mathbf{s}$ and $\sum_{\mathbf{s}} p_\infty(\mathbf{s}) = 1$. Thus, there exists a manifold of dimension $k-1$ of

[*3] The fact that all eigenvalues are within the unit circle in the complex plane can be easily demonstrated in the following way. Let λ be an eigenvalue of T and l its corresponding left eigenvector. Then for all s,

$$(\lambda - T(s|s))l(s) = \sum_{s' \neq s} l(s')T(s'|s).$$

Choose s such that $|l(s)|$ is maximal. Then

$$|\lambda - T(s|s)| = \frac{1}{|l(s)|}|\sum_{s' \neq s} l(s')T(s'|s)| \leq \sum_{s' \neq s} T(s'|s) = 1 - T(s|s).$$

This statement is known as Gershgoren's Theorem. Thus, λ is within a circle of radius $1 - T(s|s)$ centered at $T(s|s)$. We do not know which s maximizes $|l(s)|$ and therefore we do not know the value of $T(s|s)$. However, since circles with smaller $T(s|s)$ contain circles with larger $T(s|s)$, λ is in the largest circle: $|\lambda| < 1$. This completes the proof.

stationary distributions.

In addition, the k left eigenvectors with eigenvalue 1 encode *invariants* of the Markov process in the following way. Let the state of the network at time t be given by p_t. Define $L_\alpha(p_t) = l_\alpha^\dagger p_t, \alpha = 1, \ldots, k$. Then L_α is invariant under the Markov dynamics:

$$L_\alpha(p_{t+1}) = l_\alpha^\dagger p_{t+1} = l_\alpha^\dagger T p_t = l_\alpha^\dagger p_t = L_\alpha(p_t).$$

One of these invariants is the left eigenvector $l_1 \propto (1, \ldots, 1)$ which ensures that the normalization of the probability vector is conserved under the Markov process. The value of the remaining $k-1$ invariants are determined by the initial distribution p_0. Since their value is unchanged during the dynamics they parametrize the stationary manifold and determine uniquely the stationary distribution. We can thus compute the dependence of the stationary distribution on the initial state. Because of the orthogonality relation Eq. 11, we obtain $L_\alpha(p_\infty) = l_\alpha^\dagger p_\infty = \rho_\alpha$. Because L_α is invariant, we also have $L_\alpha(p_0) = L_\alpha(p_\infty)$. Thus, $\rho_\alpha = L_\alpha(p_0)$ and the stationary state depends on the initial state as

$$p_\infty = \sum_{\alpha=1}^{k} (l_\alpha^\dagger p_0) r_\alpha. \tag{15}$$

Note, that in the ergodic case ($k = 1$) the dependence on the initial state disappears, as it should, since $l_1^\dagger p_0 = 1$ for any initial distribution.

The time to approach stationarity is also given by the eigenvalues of T. In particular, each eigenvalue whose norm $|\lambda_\alpha| < 1$ corresponds to a transient mode in Eq. 12 with *relaxation time* $\tau_\alpha = \dfrac{-1}{\log \lambda_\alpha}$.

Both concepts of irreducibility and periodicity are important for neural networks and we therefore illustrate them with a number of simple examples.

Consider a network of two neurons connected symmetrically by a synaptic weight $w = w_{12} = w_{21}$. First consider sequential dynamics. The transition matrix T has 4 eigenvalues. Their values as a function of w are plotted in Fig. 2a. We observe, that for small w there exists only one eigenvalue 1. Its corresponding right eigenvector is the Boltzmann-Gibbs distribution $p(s_1, s_2) = \dfrac{\exp(w s_1 s_2)}{Z}$ as will be shown below. For small weigths, the dynamics is ergodic: for any initialization of the network the asymptotic stationary distribution is the Boltzmann-Gibbs distribution. The dominant relaxation time is given by the largest eigenvalue that is smaller than 1. For larger w, we observe that the relaxation time

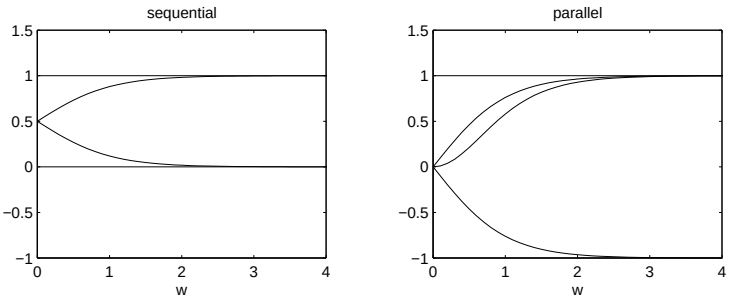

Fig. 2 Eigenvalues of T as a function of w under sequential and parallel dynamics. For large w, multiple eigenvalues 1 signal ergodicity breaking.

becomes infinite because a second eigenvalue approaches 1. This means that some transitions in the state space require infinite time and therefore ergodicity is broken. The large weight prohibits the two neurons to have opposite value and therefore only the states $(1,1)$ and $(-1,-1)$ have positive probability in the stationary distribution. Let us denote the 4 states $(1,1),(1,-1),(-1,1),(-1,-1)$ by $\mathbf{s}^\mu, \mu = 1,\ldots,4$. The right eigenvectors with eigenvalue 1 are the Boltzmann-Gibbs distribution

$$r_1(\mathbf{s}) = \frac{1}{2}(\delta_{\mathbf{s},\mathbf{s}^1} + \delta_{\mathbf{s},\mathbf{s}^4})$$

and the vector

$$r_2(\mathbf{s}) = \frac{1}{2}(\delta_{\mathbf{s},\mathbf{s}^1} - \delta_{\mathbf{s},\mathbf{s}^4})$$

The stationary distribution is no longer unique and consists of any linear combination of r_1 and r_2 that is normalized and positive: $p_\infty = r_1 + \rho_2 r_2$, with $-1 < \rho_2 < 1$. The left eigenvectors with eigenvalue 1 are

$$\begin{aligned} l_1(\mathbf{s}) &= 1 \\ l_2(\mathbf{s}) &= \delta_{\mathbf{s},\mathbf{s}^1} - \delta_{\mathbf{s},\mathbf{s}^4} \end{aligned}$$

and the corresponding quantities L_1 and L_2 are conserved. The dependence of the stationary distribution on the initial distribution is given by Eq. 15 with $k = 2$. In particular, the 4 pure states are mapped onto:

$$\begin{aligned} \mathbf{s}^1 : L_2 = 1 \quad &\to p_\infty(\mathbf{s}) = r_1(\mathbf{s}) + r_2(\mathbf{s}) = \delta_{\mathbf{s},\mathbf{s}^1} \\ \mathbf{s}^{2,3} : L_2 = 0 \ &\to p_\infty(\mathbf{s}) = r_1(\mathbf{s}) \\ \mathbf{s}^4 : L_2 = -1 \ &\to p_\infty(\mathbf{s}) = r_1(\mathbf{s}) - r_2(\mathbf{s}) = \delta_{\mathbf{s},\mathbf{s}^4} \end{aligned}$$

For the same network with parallel dynamics, the eigenvalues are depicted in Fig. 2b. For small weights the network is again ergodic. The stationary distribution is given by Eq. 20 and is flat: independent of w and $\mathbf{s}$. For large weights ergodicity breaking occurs together with the occurence of a cycle of period 2 and two additional eigenvalues 1. Using the invariants, it is easy to show that the 4 pure states are mapped onto the stationary distributions:

$$\begin{aligned}
\mathbf{s}^1 &\rightarrow p_\infty(\mathbf{s}) = \delta_{\mathbf{s},\mathbf{s}^1} \\
\mathbf{s}^{2,3} &\rightarrow p_\infty(\mathbf{s}) = \frac{1}{2}(\delta_{\mathbf{s},\mathbf{s}^2} + \delta_{\mathbf{s},\mathbf{s}^3}) \\
\mathbf{s}^4 &\rightarrow p_\infty(\mathbf{s}) = \delta_{\mathbf{s},\mathbf{s}^4}
\end{aligned}$$

States $\mathbf{s}^1$ and $\mathbf{s}^4$ are two attractors: $T\mathbf{s}^{1,4} = \mathbf{s}^{1,4}$. States $\mathbf{s}^2$ and $\mathbf{s}^3$ form a limit cycle of length 2: $T^2\mathbf{s}^2 = T\mathbf{s}^3 = \mathbf{s}^2$. Note in particular, that none of the states is mapped onto the ergodic stationary distribution Eq. 20 when ergodicity is broken.

In our examples we have seen that for symmetric networks, all eigen values of T are real. This is indeed in general true for both parallel and sequential dynamics: $-1 \leq \lambda_\alpha \leq 1$. In addition, one can show for sequential dynamics (symmetric or asymmetric) that all eigenvalues are within the circle centered at $\frac{1}{2} + 0i$ with radius $\frac{1}{2}$ [18]. The proof of this last statement again uses Gershgoren's Theorem and the special property of sequential dynamics that $T(F_i s|s) + T(s|F_i s) = \frac{1}{n}$. As a consequence, sequential dynamics has always periodicity 1 since other eigenvalues with $|\lambda| = 1$ are excluded. Note, that this property holds regardless of whether the network has symmetric or asymmetric connectivity. It also follows that for parallel dynamics with symmetric weights one can have at most periodicity 2. Parallel dynamics with asymmetric weights can have arbitrary periodicity and will be discussed in section 6.1.

§5 Boltzmann-Gibbs distributions

If we consider a stochastic neural network with a random connectivity matrix, what will the behavior of the network be? This is a rather difficult question to answer in general, but in some specific cases quite a lot is known. In particular for symmetrically connected networks with sequential dynamics, the equilibrium distribution is the Boltzmann-Gibbs distribution which plays a central role in statistical physics (see also the chapter by Coolen in this book).

In this section we derive the Boltzmann-Gibbs distribution. Then we indicate the computational problems associated with the computation of statistics of the Boltzmann-Gibbs distribution. Subsequently, we will use the cavity method to describe the behavior of an ensemble of randomly generated networks. It is shown that depending on the type of connectivity in the network, it can be in one of three possible phases: a paramagnetic phase where neural firing is weakly correlated, a ferromagnetic phase where large groups of neurons assume either maximal or minimal firing rates, and a spin-glass phase where neurons are frozen in a random disordered state. Subsequently, we briefly discuss the Hopfield attractor neural network and explain how the above phases arise in this model and affect its storage capacity.

5.1 The stationary distribution

In the case that the synaptic connectivity is symmetric, $w_{ij} = w_{ji}$ one can compute the stationary probability distribution for the parallel and sequential dynamics explicitly. In both cases the derivation uses the argument of detailed balance, which states that for the dynamics $T(s'|s)$ there exists a function $p(s)$ such that

$$T(s|s')p(s') = T(s'|s)p(s) \text{ for all } s, s'. \tag{16}$$

If detailed balance holds, it implies that $p(s)$ is a stationary distribution of T, which is easily verified by summing both sides of Eq. 16 over all states s' and using Eq. 10. However, the reverse is not true: many stochastic dynamics do not satisfy detailed balance and a solution to Eq. 13 is then typically not available in analytical form, although its existence is dictated by the Perron-Frobenius theorem [20].

For random sequential dynamics, T is given by Eqs. 8 and 5 and the detailed balance equation reads $T(F_i s|s)p(s) = T(s|F_i s)p(F_i s)$ for all states s and all neighbor states $F_i s$. It is easy to show that

$$\frac{T(s|F_i s)}{T(F_i s|s)} = \exp(2(\sum_j w_{ij} s_j + \theta_i)s_i).$$

Consider the distribution

$$p(s) = \frac{1}{Z} \exp(\frac{1}{2} \sum_{ij} w_{ij} s_i s_j + \sum_i \theta_i s_i). \tag{17}$$

$p(s)$ is called a Boltzmann-Gibbs distribution and plays a central role in statistical physics. For this reason, the expression in the exponent is often referred to as the energy:

$$-E(s) = \frac{1}{2}\sum_{ij} w_{ij}s_i s_j + \sum_i \theta_i s_i. \tag{18}$$

States of low energy have high probability. Z is a normalization constant,

$$Z = \sum_s \exp(-E(s)) \tag{19}$$

and is called the partition function. $p(s)$ only depends on the symmetric part of the weights w^s_{ij} and

$$\frac{p(s)}{p(F_i s)} = \exp(2(\sum_j w^s_{ij}s_j + \theta_i)s_i).$$

Thus for symmetric weights, detailed balance is satisfied between all neighboring states. Since all values of T are zero for non-neighboring states this proves that $p(s)$ is the equilibrium distribution. [*4]

5.2 Computing statistics

$p(s)$ in Eq. 17 and 20 give an analytical expression of the stationary probability distribution of an arbitrary network with symmetric connectivity and sequential and parallel dynamics, respectively. From these equations we can compute any interesting *statistics*, such as for instance the mean firing rate of each of the neurons:

$$m_i = \langle s_i \rangle = \sum_s s_i p(s), \tag{21}$$

[*4] When all neurons are updated in parallel, the transition matrix is given by Eq. 6. As in the case of sequential dynamics, we can again compute the stationary distribution for symmetric weights. We use again detailed balance:

$$\frac{T(s'|s)}{T(s|s')} = \frac{\exp(\sum_{ij} w_{ij}s_j s'_i + \sum_i \theta_i s'_i)}{\exp(\sum_{ij} w_{ij}s'_j s_i + \sum_i \theta_i s_i)} \prod_i \frac{\cosh(h_i(s'))}{\cosh(h_i(s))}.$$

When the weights are symmetric, the term involving the double sum over i and and j cancels and the remainder is of the form $\frac{p(s')}{p(s)}$, with

$$p(s) = \frac{1}{Z}\exp(\sum_i \log\cosh(\sum_j w_{ij}s_j + \theta_i) + \sum_i \theta_i s_i). \tag{20}$$

This is the equilibrium distribution for parallel dynamics [17].

and correlations between neurons:

$$\chi_{ij} = \langle s_i s_j \rangle - \langle s_i \rangle \langle s_j \rangle = \sum_s s_i s_j p(s) - m_i m_j. \tag{22}$$

However, these computations are in general too time consuming due to the sum over all states, which involves 2^n terms.

For some distributions, the sum can be performed efficiently. For Boltzmann-Gibbs distributions, the subset of probability distributions for which the sum over states can be performed efficiently are called decimatable distributions [21]. These include factorized distributions, trees and some other special graphs as sub sets. For factorized distributions, $p(s) = \prod_i p_i(s_i)$, the energy only depends linearly on s_i and the sum over states can be performed by factorization:

$$\sum_s \exp(\sum_i \alpha_i s_i) = \prod_i \left(\sum_{s_i} \exp(\alpha_i s_i) \right) = \prod_i 2\cosh(\alpha_i).$$

From Eqs. 17 and 20 we infer that this corresponds to the rather uninteresting case of a network without synaptic connections. [*5]

In general, the sum over states can not be computed in any simple way. In this case we call the the probability distribution *intractable* and one needs to apply approximation methods to compute the partition function and statistics such as Eq. 21 and 22.

For specific models, i.e. specific realizations of the connections and thresholds, one can obtain a generic description of the network behavior by using mean field theory. Such an approach typically considers not one network, but an *ensemble of networks* and the limit of $n \to \infty$. One can then compute the average behavior of such networks. This approach has been successfully applied to many neural network models, such as the attractor neural network proposed by Hopfield [5]. In this section we will briefly outline this approach and give some

[*5] The probability distribution $p(s)$ is called a *tree* when between any two neurons in the network there exists only one path, where a path is a sequence of connections. Alternatively, one can order the neurons in the graph with labels $1, \ldots, n$ such that neuron i is connected to any number of neurons with higher label but only to at most one neuron with lower label. For Boltzmann-Gibbs distributions which are trees:

$$\sum_s \exp(\sum_{(ij)} w_{ij} s_i s_j) = \sum_s \exp(\sum_i w_{ip_i} s_i s_{p_i}) = \prod_i 2\cosh(w_{ip_i}),$$

where p_i labels the parent of neuron i. For parallel dynamics, such non-trivial decimatable structures do not exist.

of the most well-known results. For a more complete review see the contribution by Coolen in this volume.

5.3 The cavity method

There are various ways to derive mean field results. The most well-known approach is to use the replica method. Here we will consider a somewhat simpler approach called the cavity method, which dates back to [22] [*6].

If we want to compute the mean firing rate of neuron i we must compute the exponential sum in Eq. 21. Due to the dependence of $p(s)$ on Z, we must also compute the exponential sum in Eq. 19. The idea of the cavity method is to separate these sums in a contribution from neuron i and a contribution from all other neurons in the following way:

$$\begin{aligned} Z &= \sum_{s\setminus i}\sum_{s_i} \exp(s_i h_i)\exp(-E_{\setminus i}) = 2\,\langle \cosh h_i\rangle_{\setminus i}\, Z_{\setminus i} \\ Z\,\langle s_i\rangle &= \sum_{s\setminus i}\sum_{s_i} s_i \exp(s_i h_i)\exp(-E_{\setminus i}) = 2\,\langle \sinh h_i\rangle_{\setminus i}\, Z_{\setminus i} \end{aligned}$$

where $E_{\setminus i}$ denotes all contributions to the energy excluding dependencies on s_i and $\langle\cdot\rangle_{\setminus i}$ denotes ensemble average with respect to the Boltzmann-Gibbs distribution $p_{\setminus i} = \dfrac{\exp(-E_{\setminus i})}{Z_{\setminus i}}$. Thus we obtain

$$\langle s_i\rangle = \frac{\langle \sinh h_i\rangle_{\setminus i}}{\langle \cosh h_i\rangle_{\setminus i}} \tag{23}$$

h_i is the local field defined previously: $h_i = \sum_{j\neq i} w_{ij}s_j + \theta_i$. It is a stochastic quantity consisting of a sum of contributions from all other neurons. In particular, h_i does not depend on s_i. However, $\langle h_i\rangle$ does depend on s_i, because s_i affects the mean firing rates of all neurons in the network. For instance, if all connections are positive, $s_i = \pm 1$ will increase (decrease) all firing rates $\langle s_j\rangle$ slightly.

Instead we consider the restricted averages and write

$$h_i = \langle h_i\rangle_{\setminus i} + u_i. \tag{24}$$

[*6] Here we use a particularly transparent formulation which was communicated to me by Manfred Opper

u_i is a stochastic quantity that we assume to be symmetrically distributed under $p_{\setminus i}$: $p_{\setminus i}(-u_i) = p_{\setminus i}(u_i)$. In particular, $\langle u_i \rangle_{\setminus i} = 0$. Substituting Eq. 24 in Eq. 23 we obtain

$$\begin{aligned} \langle \sinh(h_i) \rangle_{\setminus i} &= \sinh(\langle h_i \rangle_{\setminus i}) \langle \exp u_i \rangle_{\setminus i} \\ \langle \cosh(h_i) \rangle_{\setminus i} &= \cosh(\langle h_i \rangle_{\setminus i}) \langle \exp u_i \rangle_{\setminus i} \end{aligned}$$

and thus

$$\langle s_i \rangle = \tanh(\langle h_i \rangle_{\setminus i}). \tag{25}$$

This is the main result of the cavity method. It states that the expected firing rate of neuron i only depends on the expected value of the local field computed *in the absence of neuron i.*

5.4 Quenched average solution for the SK model

We can use Eq. 25 to compute the typical behavior of an ensemble of networks in the limit of large n. Before we consider the attractor neural network, we will first analyze the behavior of a simpler model, the so-called Sherrington-Kirkpatrick (SK) model. In this model one assumes that w_{ij} are drawn independently at random from a Gaussian distribution with mean value $\frac{J_0}{n-1}$ and variance $\frac{J^2}{n-1}$. θ_i is drawn independently at random from a Gaussian distribution with mean value I_0 and variance I^2.

For one realization of the weights, $\langle h_i \rangle_{\setminus i}$ is not a random quantity, but just a number given by

$$\langle h_i \rangle_{\setminus i} = \sum_j w_{ij} \langle s_j \rangle_{\setminus i} + \theta_i$$

We compute the distribution of $\langle h_i \rangle_{\setminus i}$ in the ensemble of networks. This is called a *quenched* average, where quenched means fixed and refers to the fact that we compute the Boltzmann-Gibbs statistics for a fixed realization of the weights and thresholds and, after that, average the resulting firing rates over all realizations of weights and thresholds.

The first term in $\langle h_i \rangle_{\setminus i}$ depends on the connections to neuron i. It is multiplied by a term which is an expectation value computed in the network from which neuron i is removed and therefore *does not depend on the connections to neuron i:* $w_{ij}, j = 1, \ldots, n$. We can therefore easily compute the statistics with respect to w_{ij} and θ_i. Since $\langle h_i \rangle_{\setminus i}$ is a large sum of random contributions, it has

a Gaussian distribution. Its mean value and variance are

$$\left\langle \langle h_i \rangle_{\setminus i} \right\rangle_w = \frac{J_0}{n-1} \sum_{j \neq i} \langle s_j \rangle_{\setminus i} + I_0 = J_0 m + I_0$$

$$\left\langle (\langle h_i \rangle_{\setminus i})^2 \right\rangle_w - \left(\left\langle \langle h_i \rangle_{\setminus i} \right\rangle_w \right)^2 = J^2 q + I^2$$

where we have defined $m = \frac{1}{n-1} \sum_{j \neq i} \langle s_j \rangle_{\setminus i}$ and $q = \frac{1}{n-1} \sum_{j \neq i} \langle s_j \rangle_{\setminus i}^2$ and $\langle \cdot \rangle_w$ denotes average with respect to w_{ij}. Note, that m and q are independent of i in the limit $n \to \infty$:

$$m = \frac{1}{n-1} \sum_{j \neq i} \langle s_j \rangle_{\setminus i} \approx \frac{1}{n} \sum_j \langle s_j \rangle ,$$

$$q = \frac{1}{n-1} \sum_{j \neq i} \langle s_j \rangle_{\setminus i}^2 \approx \frac{1}{n} \sum_j \langle s_j \rangle^2 .$$

Thus the quenched average of Eq. 25 becomes

$$m = \frac{1}{\sqrt{2\pi}} \int dz e^{-\frac{z^2}{2}} \tanh(\sqrt{qJ^2 + I^2} z + J_0 m + I_0) \qquad (26)$$

and the quenched average of the square of Eq. 25 becomes

$$q = \frac{1}{\sqrt{2\pi}} \int dz e^{-\frac{z^2}{2}} \tanh^2(\sqrt{qJ^2 + I^2} z + J_0 m + I_0) \qquad (27)$$

These equations are identical to the replica symmetric solution as given in [23].

The solutions for m and q of Eqs. 26 and 27 can be computed for different values of J_0, J, I_0 and I. Here we restrict ourselves to the case $I_0 = I = 0$, ie. $\theta_i = 0$. For both J_0 and J small, the only solution is $m = q = 0$. From the definitions of m and q we see that this means that $\langle s_i \rangle = 0$. This is the regime where the couplings between neurons are small and can in fact be ignored. The mean firing rate is given by the threshold value: $\langle s_i \rangle \approx \tanh(\theta_i)$. This is called the *paramagnetic phase*.

One can perform a linear stability analysis of the solution $m = q = 0$ and one finds that the solution is stable as long as $0 < J_0 < 1$ and $0 < J^2 < 1$. For $J_0 > 1$ and $J < J_0$, most of the weights are positive. The neurons will excite each other and the net result is that all neurons will align. The solution that is obtained is $q = m = 1$ which means $\langle s_i \rangle = 1$. This is the *ferromagnetic phase*. On the other hand, for $J > 1$ and $J_0 < J$ the weights are large but of opposite sign. As a result, the network is *frustrated* [24]. When for instance three

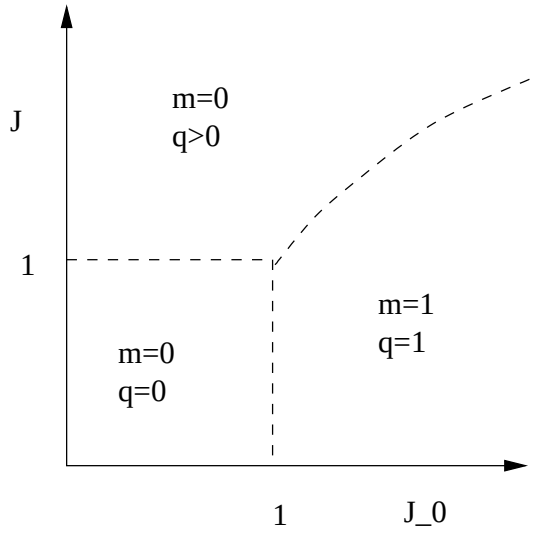

Fig. 3

neurons are connected by two positive connections and one negative connection, there is no configuration s such that $w_{ij}s_is_j > 0$ for all pairs. As a result of frustration, the energy contains (often exponentially) many local minima that have approximately the same value. In contrast, in the absence of frustration the energy contains only one (or maybe a few, due to symmetries) minimum. The solution is $m = 0$ and $q > 0$. It means that the each of the neurons has mean value $\langle s_i \rangle = \pm\sqrt{q}$ and is frozen in this state. This frozen disorder is called the *spin glass* phase. The results are summarized in Fig. 3.

When thresholds are present the behavior of the network remains qualitatively the same. For instance, there is still a paramagnetic phase where neurons fire more or less independently, but the mean firing rates are now not zero but given by $\langle s_i \rangle \approx \tanh(\theta_i)$. For larger weights there are transitions to ferromagnetic and spin glass phases. The transitions between phases is less abrupt for non-zero thresholds. When I_0 and I are large compared to J_0 and J, respectively, the collective behavior of the network breaks down and each neuron aligns according to its threshold.

§6 Asymmetric networks

6.1 The differences between symmetric and asymmetric networks

In the previous section, we have studied the typical behavior of symmetric networks. However, the assumption of symmetry is rather unrealistic. What differences should we expect when we consider asymmetric networks?

There are several ways to introduce asymmetry in neural networks. One approach uses temporal associations of neural activity of the form $w_{ij} \propto s_i(t)s_j(t+1)$. In this way, a *sequence* of patterns is memorized by the network. Such a type of memory may be needed to represent various temporal behaviors, such as motor control tasks or generation of speech. The simplest way to memorize such sequences is by using parallel dynamics. One simply assumes $w_{ij} \propto \sum_{\mu}^{p} \xi_i^{\mu+1}\xi_j^{\mu}$, with $\xi^{\mu}, \mu = \mathbf{1}, \ldots, \mathbf{p}$ the sequence of patterns. The effect is that the dynamics of the network contains a limit cycle formed by the sequence of patterns. Regarding the storage capacity of such a network, one can show a similar result as in the symmetric case, i.e. that for low noise level and for small enough p the sequence is recalled [25]. See for instance [26] for a review of these models.

Another frequently studied model is the asymmetric SK model. The weights are drawn at random from a mean zero Gaussian distribution. The asymmetry is controlled by a parameter

$$\eta = \frac{\langle w_{ij} w_{ji} \rangle_w}{\langle w_{ij}^2 \rangle_w}.$$

$\eta = 1, 0, -1$ describe the symmetric, asymmetric and anti-symmetric case, respectively.

The behavior of these networks have been studied extensively in the noiseless limit [27, 28, 29, 30]. For symmetric and anti-symmetric networks it is easy to show that they have fixed points and limit cycles of length 2 (symmetric

case) and limit cycles of length 4 (anti-symmetric) [31, 18]. [*7] As we saw in the previous section, the symmetric noiseless network is a spin glass. Ergodicity is severely broken: Any initial state will converge to one of the exponentially many local minima or limit cycles of length 2, for sequential or parallel dynamics respectively. The relaxation time is polynomial in the size of the network [29]. Ergodicity is also broken in the antisymmetric network, leading to exponentially many limit cycles of length 4, for sequential or parallel dynamics respectively.

For asymmetric networks, with $-\eta_c < \eta < \eta_c$ and $\eta_c \approx 0.5$, the behavior is radically different. In the noiseless limit, there exist exponentially long limit cycles that dominate the network dynamics [27]. Thus, to compute statistics in the stationary state requires a simulation time that is exponential in the network size [29]. This latter feature is particularly important, because it means that the behavior of the network for any finite time is transient and its stationary statistics become in a sense irrelevant. The behavior of the network can be interpreted as chaotic: divergent trajectories and exponential size of transients [32].

We illustrate the effect of asymmetry in Fig. 4, where we show the eigenvalue spectrum of a fully connected symmetric, asymmetric and antisymmetric network of 8 neurons with parallel dynamics. The symmetric network has positive and negative real eigenvalues indicating the possibility of fixed point solu-

[*7] We give the derivation for $\theta_i = 0$. Consider the two-time Lyapunov function

$$L(t, t+1) = -\sum_{i,j} s_i(t+1) w_{ij} s_j(t).$$

Under parallel dynamics the change of L in one time step is

$$\begin{aligned} \Delta L &= L(t+1, t+2) - L(t, t+1) \\ &= -\sum_i \left(s_i(t+2) \sum_j w_{ij} s_j(t+1) - s_i(t) \sum_j w_{ji} s_j(t+1) \right). \end{aligned}$$

We consider the symmetric and anti-symmetric case: $w_{ij} = k w_{ji}$ with $k = \pm 1$. In addition, define $\kappa_i(t) = \pm 1$ such that $s_i(t+2) = \kappa_i(t) s_i(t)$. Then

$$\Delta L = \sum_i (k\kappa_i - 1) s_i(t+2) h_i(t+1).$$

Due to the parallel dynamics, $s_i(t+2)h_i(t+1)$ is always positive for all i. Since $k\kappa_i = \pm 1$, we have $\Delta L \leq 0$. Since L is bounded from below the dynamics converges to a state where $\Delta L = 0$ and therefore

$$k\kappa_i = 1 \text{ for all } i.$$

For symmetric networks, $k = 1$ and thus $s_i(t+2) = s_i(t)$: the network has limit cycles of length 1 and 2. For anti-symmetric networks, $k = -1$ and thus $s_i(t+2) = -s_i(t)$. This excludes fixed points and limit cycles of length 2 but allows limit cycles of length 4.

tions and limit cycles of length 1 and 2. The anti-symmetric network has eigenvalues at multiples of i and therefore displays period 4 cycles. Both symmetric and anti-symmetric networks display ergodicity breaking. The asymmetric network has eigenvalues all over the complex circle. In the noiseless case we discern a periodic orbit of length 14. No ergodicity breaking occurs.

6.2 Mean field theory in the absence of detailed balance

In section 5 we have seen how a quenched averaged approach is capable of describing the typical behavior of neural networks. However, in many instances we are not satisfied with such average results but would like to say something about an individual network. An example is when we consider learning. It has been well established experimentally, that synapses change their strength as a function of the firing of the pre and post synaptic neuron. In order to compute these changes, one need therefore estimates of the mean firing rates and the (time-delayed) correlations of the pre and post synaptic neuron. However, as we have seen these quantities are difficult to compute.

In this section we therefore consider a form of mean field theory that was previously proposed by Plefka [33] for Boltzmann-Gibbs distributions. It turns out that the restriction to Boltzmann-Gibbs distributions is not necessary and one can derive results that are valid also for asymmetric networks as well as for parallel dynamics. We therefore consider the general case. A drawback of this approach is that it is only valid for small values of the weights. However, as we have seen in section 2 this is to be expected for biological networks, because due to noise the effective synaptic strength scales with $\frac{1}{\sqrt{n}}$. We use this method to compute the mean field equations and correlations for asymmetric stochastic networks with sequential dynamics. Subsequently, we will illustrate the approach for learning in Boltzmann Machines.

Our argument uses an information geometric viewpoint. For an introduction to this approach see for instance [34]. In section 4 we have seen that when the stochastic neuron dynamics is ergodic, it has a unique stationary probability distribution. We will assume ergodicity and denote the stationary distribution by $p(\mathbf{s}|\theta, w)$, which is a probability distribution over $\mathbf{s}$ and depends on the weigths and thresholds of the network. Unless the connectivity is symmetric, we do not know its functional form explicitly.

Let $\mathcal{P} = \{p(s|\theta, w)\}$ be the manifold of all the probability distributions

over the state space $\mathcal{S}$ that can be obtained by considering different values of θ, w. $\mathcal{P}$ contains a submanifold $\mathcal{M} \subset \mathcal{P}$ of factorized probability distributions. This submanifold is described by

$$\mathcal{M} = \{q(s|\theta, w) \in \mathcal{P} | w = 0\}.$$

$\theta = (\theta_1, \ldots, \theta_n)$ parametrizes the manifold $\mathcal{M}$, and w parametrizes the remainder of the manifold $\mathcal{P}$. For $q \in \mathcal{M}$ we can write the stationary distribution explicitly:

$$q(s|\theta^q) = \prod_i \frac{\exp(\theta_i^q s_i)}{2\cosh(\theta_i^q)} = \prod_i \frac{1}{2}(1 + m_i^q s_i),$$

with $m_i^q = \langle s_i \rangle_q = \tanh(\theta_i^q)$. Here, $\langle \cdot \rangle_q$ denotes expectation value with respect to the distribution q. The submanifold $\mathcal{M}$ describes the factorized stationary distributions for networks with all synaptic connections zero.

Consider a network, whose weights and thresholds are given by θ, w. This network has a stationary distribution $p(s|\theta, w) \in \mathcal{S}$. We want to find its *mean field approximation* which we define as the factorized distribution $q \in \mathcal{M}$ that we obtain by orthogonal projection of p onto $\mathcal{M}$. It can then be shown [34, 35], that the orthogonal projection onto $\mathcal{M}$ is found by minimizing the relative entropy

$$D(p, q) = \sum_s p(s|\theta, w) \log\left(\frac{p(s|\theta, w)}{q(s|\theta^q)}\right)$$

with respect to the coordinates of θ^q of the factorized distribution q. We find

$$\frac{dD(p, q)}{d\theta_i^q} = m_i^q - m_i^p = 0, \tag{28}$$

with $m_i^p = \langle s_i \rangle_p$. This equation states that the closest factorized model has its first moments equal to the first moments of the target distribution p. This is illustrated in Fig. 5.

We need to solve Eq. 28 for $\theta_i^q = \tanh^{-1}(m_i^q)$. However, we can not compute m_i^p since we do not know the stationary distribution p. Even if we knew p (for instance Boltzmann-Gibbs distribution) it would be of little help, since computation of m_i^p is intractable. In order to proceed, we assume that the distribution p is somehow close to the submanifold $\mathcal{M}$. Define $d\theta_i = \theta_i - \theta_i^q$ and

$dw_{ij} = w_{ij} - 0 = w_{ij}$. Expanding $dm_i = m_i^p - m_i^q$ to second order we obtain

$$0 = dm_i \approx \sum_J \frac{\partial m_i}{\partial \Theta_J}|_q d\Theta_J + \frac{1}{2}\sum_{J,K} \frac{\partial^2 m_i}{\partial \Theta_J \partial \Theta_K}|_q d\Theta_J d\theta_K, \quad (29)$$

where $\Theta_I = (\theta_i, w_{ij})$ is the vector of all weights and thresholds.

In order to proceed we need to compute the dependence of m_i on θ, w *in the factorized point* q. We can use Eqs. 13 and 21 and the definitions of the transition matrices T for sequential and parallel dynamics Eqs. 8, 9 and 6 to get the implicit relations

$$\langle s_i \rangle = \langle \tanh(h_i(s)) \rangle \,. \quad (30)$$

This equation holds for both sequential and parallel dynamics. The computation of the derivatives is tedious but straightforward. It is presented in the Appendix. The result is

$$m_i = \tanh(\sum_j w_{ij} m_j + \theta_i - m_i \sum_j w_{ij}^2 (1 - m_j^2)). \quad (31)$$

Eq. 31 is our main result and gives the approximate mean firing rates for arbitrary dynamics and arbitrary (but small) synaptic connections. In the case of symmetric connections $w_{ij} = w_{ji}$, Eq. 31 were first derived by Thouless, Anderson and Palmer and are referred to as the TAP equations [36].

The correlations can be computed in a similar manner, but depend on the type of dynamics. We restrict ourselves to sequential dynamics and equal time correlations. From Eq. 22 we obtain

$$\langle s_i s_j \rangle = \frac{1}{2} \langle s_i \tanh(h_j(s)) \rangle + (i \leftrightarrow j). \quad (32)$$

When we expand χ_{ij} around the factorized solution $\chi_{ij}^q = 0$, we obtain

$$\begin{aligned} \chi_{ij} \;=\; & \frac{1}{2}(1 - m_i^2)(1 - m_j^2) w_{ij} \\ & + \frac{1}{2}(1 - m_i^2)(1 - m_j^2) \left(\sum_{k \neq i} w_{ik} w_{ik}^s (1 - m_k^2) + 2 m_i m_j (w_{ji})^2 \right) \\ & + (i \leftrightarrow j) \end{aligned} \quad (33)$$

To evaluate the quality of our mean-field approximations, we compare them to results of Monte Carlo simulations. We consider networks of $n = 100$

neurons. We choose $w_{ij}^0, i \neq j$ random and independently from a normal distribution with mean zero and variance $\frac{1}{\sqrt{n}}$. We consider two different types of weights: symmetric weights $w_{ij}^0 = w_{ji}^0$ and asymmetric weights, where w_{ij}^0 and w_{ji}^0 are drawn independently. We consider two types of thresholds: $\theta_i^0 = 0$ and θ_i^0 random and independently from a normal distribution with mean zero and variance 1. Since the approximation is expected to deteriorate with increasing weights size, we consider networks with $(w_{ij}, \theta_i) = \beta(w_{ij}^0, \theta_i^0)$ and vary $0 \leq \beta \leq 1$.

We use Monte Carlo simulations to estimate the mean firing rates $\langle s_i \rangle$ and correlations χ_{ij}. The states are generated using sequential Glauber dynamics. To minimize the initialization (burn in) effect, we start the network in a random state and do not include the first t_0 iterations. We compute the average over the subsequent τ states:

$$\langle s_i \rangle^{mc} = \frac{1}{\tau} \sum_{t=t_0}^{t=t_0+\tau} s_i(t) \tag{34}$$

$$\chi_{ij}^{mc} = \frac{1}{\tau} \sum_{t=t_0}^{t=t_0+\tau} s_i(t) s_j(t) - \langle s_i \rangle^{mc} \langle s_j \rangle^{mc} \tag{35}$$

The results are rather dependent on a proper choice of t_0 and τ. We obtained stable results by chosing $t_0 = 10^5 n$ and $\tau = 10^6 n$. These values are rather large, but necessary to get results accurate enough to compute the small χ_{ij}'s. (The χ_{ij}'s are small because to lowest order $\chi_{ij} \propto w_{ij} \propto \frac{1}{\sqrt{n}}$).

From Eq. 31 we compute the mean field approximation of the mean firing rates. In order to assess the importance of the second order (TAP) contribution, we also compute these approximate values taking only the terms of $\mathcal{O}(w)$ into account (MF). In Fig. 6, we show the root mean square (RMS) values of the mean firing rates as a function of β for the Monte Carlo solution (MC), the mean field solution (MF) solution and the TAP solution. The statistical errors in the Monte Carlo results for m_i are of the order $\delta m_i \approx 0.002$. In addition, we show the RMS values of the difference between the MF and MC solution and between the TAP and MC solution. We conclude that the second order approximation is significantly better than the first order approximation when $\beta < 1$, both for symmetric and asymmetric networks.

The results for the correlations are presented in Fig. 7. The statistical

errors in the Monte Carlo results for χ_{ij} are of the order $\delta\chi_{ij} \approx 0.002$. We compute the TAP-values for the mean firing rates and insert these in Eq. 33 to compute the correlations. We consider again separately the $\mathcal{O}(w)$ approximation and the $\mathcal{O}(w^2)$ approximation. We conclude that the second order approximation is significantly better than the first order approximation when $\beta < 0.5$, both for symmetric and asymmetric networks.

§7 Learning in neural networks

7.1 Attractor neural networks

In 1982, John Hopfield wrote a seminal paper, where he proposed a stochastic neural network in which the connections are the result of Hebbian learning [5]. Hebbian learning is the mechanism that neurons increase their connection strength when they are both active at the same time. The rationale is that when a presynaptic spike contributions to the firing of the post synaptic neuron, it is likely that its contribution is of some functional importance to the animal and therefore the efficacy of the responsible synapse should be increased. If however, the pre synaptic spike does not result in the firing of the post synaptic cell, or vise versa, that the post synaptic cell fires in the absence of the pre synaptic spike, the synapse is probably not very important and its strength is decreased. One could summarize this behavior as

$$\Delta w_{ij} = \eta(y_i(t)y_j(t) - \lambda w_{ij}),$$

where $y_i(t) = 0, 1$ denote the firing of neuron i between time t and $t+\tau$ as defined in section 2. η is the learning rate and λ is a small positive constant. Although the mechanism of Hebbian learning has been confirmed in various experiments [37], the picture is considered to be too simple. In particular, synapses display an interesting history dependent dynamics with characteristic time scales of several msec to hours.

The analysis of stochastic networks with Hebbian connectivity was performed in a series of papers by Amit, Gutfreund and Sompolinsky [9, 10]. They considered various 'Hebbian' learning rules which are similar, but not quite identical to the Hebbian mechanism discussed above. Nevertheless, one expects that the behavior of this model is qualitatively the same as for biological networks.

Due to the symmetric connectivity, the stationary behavior of the network can be computed and is given by Eq. 17. The patterns ξ^μ become stable attractors of the dynamics when the number of patterns is sufficiently small

and β is sufficiently large. Thus the network operates as a distributed memory. When β is too small, all attractors become unstable and the firing of the neurons becomes more or less uncorrelated. This behavior is similar to the paramagnetic phase discussed in the SK model. When the number of patterns is too large, the network behaves as a *spin glass* whose minima are uncorrelated with the stored patterns. This behavior is to a large extend independent of whether the neuron dynamics is *sequential* or *parallel* (see section 3 for the definition of these terms).

7.2 Boltzmann Machines

Another well-known application of the Boltzmann-Gibbs distribution are Boltzmann Machines [8]. The basic idea is to treat the distribution Eq. 17 as a statistical model, and to use standard statistical tools to estimate its parameters w_{ij} and θ_i.

Let us partition the neurons in a set of n_v visible units and n_h hidden units ($n_v + n_h = n$). Let α and β label the 2^{n_v} visible and 2^{n_h} hidden states of the network, respectively. Thus, every state s is uniquely described by a tuple $\alpha\beta$. Learning consists of adjusting the weights and thresholds in such a way that the Boltzmann-Gibbs distribution on the visible units $p_\alpha = \sum_\beta p_{\alpha\beta}$ approximates a target distribution q_α as closely as possible.

A suitable measure for the difference between the distributions p_α and q_α is the relative entropy [38]

$$K = \sum_\alpha q_\alpha \log \frac{q_\alpha}{p_\alpha}. \tag{36}$$

It is easy to show that $K \geq 0$ for all distributions p_α and $K = 0$ iff $p_\alpha = q_\alpha$ for all α.

Therefore, learning consists of minimizing K with respect to w_{ij} and θ_i using gradient descent and the learning rules are given by [8, 39]

$$\begin{aligned} \Delta\theta_i &= -\eta \frac{\partial K}{\partial \theta_i} = \eta\Big(\langle s_i \rangle_c - \langle s_i \rangle \Big), \\ \Delta w_{ij} &= -\eta \frac{\partial K}{\partial w_{ij}} = \eta\Big(\langle s_i s_j \rangle_c - \langle s_i s_j \rangle \Big) \quad i \neq j. \end{aligned} \tag{37}$$

The parameter η is the learning rate. The brackets $\langle \cdot \rangle$ and $\langle \cdot \rangle_c$ denote the 'free' and 'clamped' expectation values, respectively. The 'free' expectation values are

defined as usual:

$$\begin{aligned} \langle s_i \rangle &= \sum_{\alpha\beta} s_i^{\alpha\beta} p_{\alpha\beta} \\ \langle s_i s_j \rangle &= \sum_{\mathbf{s}} s_i^{\alpha\beta} s_j^{\alpha\beta} p_{\alpha\beta}. \end{aligned} \tag{38}$$

The 'clamped' expectation values are obtained by clamping the visible units in a state α and taking the expectation value with respect to q_α:

$$\begin{aligned} \langle s_i \rangle_c &= \sum_{\alpha\beta} s_i^{\alpha\beta} q_\alpha p_{\beta|\alpha} \\ \langle s_i s_j \rangle_c &= \sum_{\alpha\beta} s_i^{\alpha\beta} s_j^{\alpha\beta} q_\alpha p_{\beta|\alpha} \end{aligned} \tag{39}$$

$s_i^{\alpha\beta}$ is the value of neuron i when the network is in state $\alpha\beta$. $p_{\beta|\alpha}$ is the conditional probability to observe hidden state β given that the visible state is α. Note that in Eqs. 37–39, i and j run over both visible and hidden units.

Thus, the BM learning rules contain clamped and free expectation values of the Boltzmann-Gibbs distribution. The computation of the free expectation values is intractable, because the sums in Eqs. 38 consist of 2^n terms. If q_α is given in the form of a training set of p patterns, the computation of the clamped expectation values, Eqs. 39, contains $p2^{n_h}$ terms. This is intractable as well, but usually less expensive than the free expectation values. As a result, the exact version of the BM learning algorithm can not be applied to practical problems.

We therefore apply the mean field approximation as discussed in the previous section. Due to the symmetric weights, the Boltzmann Machine is an equilibrium system and we can improve on our estimates of the correlations between neurons, Eq. 33, using the linear response theorem [40]. The starting point is to observe the exact relations

$$\langle s_i \rangle = \frac{\partial \log Z}{\partial \theta_i} \tag{40}$$

$$\chi_{ij} = \frac{\partial^2 \log Z}{\partial \theta_i \partial \theta_j}, \tag{41}$$

which follow immediately from the definition of Z. We can combine these equations and obtain

$$\chi_{ij} = \frac{\partial \langle s_i \rangle}{\partial \theta_j} \approx \frac{\partial m_i}{\partial \theta_j} \tag{42}$$

Thus, the correlations are given by the derivative of the equilibrium firing rates with respect to the thresholds. In the last step we have replaced these firing rates by their mean field estimates, Eq. 31. We can compute the right hand side of Eq. 42 from Eq. 31.

Having obtained estimates for the statistics, this basically solves the learning problem. For arbitrary w_{ij} and θ_i we can compute the mean firing rates and correlations (both clamped and free) and insert these values into the learning rule Eq. 37.

The situation is particularly simple in the absence of hidden units[*8]. In this case, $\langle\cdot\rangle_c$ does not depend on w_{ij} and θ_i and are simply given by the statistics of the data: If the data consists of p patterns with equal probability, $s_i^\mu, \mu = 1, \ldots, p$, then $\langle s_i\rangle_c = \frac{1}{p}\sum_\mu s_i^\mu$ and $\langle s_i s_j\rangle_c = \frac{1}{p}\sum_\mu s_i^\mu s_j^\mu$. Thus our task is to find w_{ij} and θ_i such that the (mean field approximations of the) free mean firing rates and correlations are equal to $\langle s_i\rangle_c$ and $\langle s_i s_j\rangle_c$, respectively:

$$m_i = \langle s_i\rangle_c \tag{43}$$

$$\chi_{ij} = \langle s_i s_j\rangle_c - m_i m_j, i \neq j. \tag{44}$$

Eqs. 43 and 44 are $n + \frac{1}{2}n(n-1)$ equations with an equal number of unknowns w_{ij} and θ_i and can be solved using standard numerical routines.

We can however, make a significant improvement in the learning procedure when we observe that the TAP term in Eq. 31 represents a self coupling to neuron i. Instead of using the TAP approximation to relate this self-coupling to the off-diagonal weights w_{ij}, we propose to introduce additional parameters, diagnonal weights w_{ii}, which we estimate in the learning process. We therefore need n additional equations for learning, for which we propose $\chi_{ii} = 1 - m_i^2$. This equation is true by definition for the exact χ, but becomes an additional constraint on w_{ij} and θ_i when χ is the linear response approximation Eq. 42. Thus our basic equations become

$$m_i = \tanh(\sum_{j=1}^n w_{ij} m_j + \theta_i) \tag{45}$$

$$\chi^{-1}_{ij} = \frac{\partial\theta_j}{\partial m_i} = \frac{\delta_{ij}}{1 - m_i^2} - w_{ij}. \tag{46}$$

[*8] The following discussion can be extended to hidden units using an EM-type of iteration procedure.

Note, that the sum over j in the equation for m_i now also includes a contribution $w_{ii}m_i$. From Eq. 43-46 we can compute the solution for w_{ij} and θ_i in closed form:

$$m_i = \langle s_i \rangle_c \tag{47}$$

$$c_{ij} = \langle s_i s_j \rangle_c - \langle s_i \rangle_c \langle s_j \rangle_c \tag{48}$$

$$w_{ij} = \frac{\delta_{ij}}{1 - m_i^2} - \left(c^{-1}\right)_{ij} \tag{49}$$

$$\theta_i = \tanh^{-1}(m_i) - \sum_{j=1}^{n} w_{ij} m_j \tag{50}$$

7.3 Classification of digits

We demonstrate the quality of the above mean field approximation for Boltzmann Machine learning on a digit recognition problem. The data consists of 11000 examples of handwritten digits (0-9) compiled by the U.S. Postal Service Office of Advanced Technology. The examples are preprocessed to produce 8×8 binary images. Some examples are shown in Fig. 8

Our approach is to model each of the digits with a separate Boltzmann Machine. For each digit, we use 700 patterns for training using the approach outlined above. We thus obtain 10 Boltzmann distributions

$$\log p(s|W^\alpha) = -E(s|W^\alpha) - \log Z(W^\alpha), \ \alpha = 0, \ldots, 9,$$

where $W^\alpha = (w_{ij}^\alpha, \theta_i^\alpha)$ are the weights and thresholds for digit α. We then test the performance of these models on a classification task using the same 700 training patterns per digit as well as the 400 test patterns per digit. We classify each pattern to the model α with the highest probability. The normalization $\log Z(W^\alpha)$ is intractable and depends on α and therefore affects classification. We use its mean field approximation given by [41, 42]

$$\log Z = -\frac{1}{2} \sum_{ij} w_{ij} m_i m_j - \sum_i \theta_i m_i$$

$$-\frac{1}{2} \sum_i \left((1 + m_i) \log(1 + m_i) + (1 - m_i) \log(1 - m_i) \right)$$

The correlation matrix c_{ij} in Eq. 48 is (close to) singular. This results in very large weights in Eq. 49 and we should question the validity of the mean field

nearest neighbor	6.7 %
back-propagation	5.6 %
wake-sleep	4.8 %
sigmoid belief	4.6 %
Boltzmann Machine	4.6 %

Table 1 Classification error rates for the test data set of handwritten digits. The first tree were reported by [43], the fourth was reported in [44].

approximation. We propose to solve this problem by adding a flat distribution to the training data:

$$q_\alpha \rightarrow (1-\lambda)q_\alpha + \lambda\frac{1}{2^n} \tag{51}$$

$$\langle s_i\rangle_c \rightarrow (1-\lambda)\langle s_i\rangle_c \tag{52}$$

$$\langle s_i s_j\rangle_c \rightarrow (1-\lambda)\langle s_i s_j\rangle_c + \lambda\delta_{ij} \tag{53}$$

In Fig. 9 we show the result of the Boltzmann Machine classifier as a function of λ. We see that the classification error depends strongly on the value of λ. However, there is no overfitting effect in the sense that a value that is optimal on the training set is also optimal on the test set. The optimal λ on the training set is $\lambda = 0.24$. The classification error on the test set for this value of λ is 4.62%. In [43, 44] this classification problem is used on the same data to compare a number of algorithms. The reported error rates on the test set are summarized in Table 1. The result obtained with the backpropagation method is rather competitive: I tried to reproduce it and it requires extensive training times and the result is not so good in all runs. The three best methods in Table 1 are all unsupervised methods. They do density estimation on each of the classes separately and are not optimized for classification. Therefore, it is encouraging that these methods are capable of outperforming the multi-layered perceptron. The Boltzmann Machine yields as good performance as the best unsupervised method known on this data. The main advantage of the Boltzmann Machine is that no hidden structure is needed in contrast to all the other methods in Table 1 except for the nearest neighbor method. As a result, the Boltzmann Machine solution is trained and tested in several minutes, whereas the other

methods require several hours [*9].

§8 Appendix: TAP equations

In this appendix we present the main steps to derive the TAP equations Eqs. 31. We start with the computation of the derivatives in Eq. 29:

$$\frac{\partial \langle s_i \rangle}{\partial \theta_j}|_q = \sum_s \frac{\partial p(s)}{\partial \theta_j}|_q \tanh(\theta_i^q) + q(s)(1 - m_{i,q}^2)\delta_{ij} = (1 - m_{i,q}^2)\delta_{ij}.$$

where $m_{i,q} = \tanh(\theta_i^q)$ is the mean firing rate of neuron i in the factorized model q. Similarly,

$$\frac{\partial \langle s_i \rangle}{\partial w_{jk}}|_q = (1 - m_{i,q}^2)\delta_{ij} m_{k,q}.$$

Using $m_i = m_{i,p} = m_{i,q}$ because of Eq. 28 we obtain to lowest order

$$0 = dm_i = (1 - m_i^2)(d\theta_i + \sum_j m_j dw_{ij}). \tag{54}$$

This is equivalent to $m_i = \tanh(\sum_j w_{ij} m_j + \theta_i^p)$.

In a similar way one computes the second order derivatives and the result is

$$\sum_{jk} \frac{\partial^2 \langle s_i \rangle}{\partial \theta_j \partial \theta_k}|_q d\theta_j d\theta_k = -2m_i(1 - m_i^2)(d\theta)_i^2$$

$$\sum_{jkl} \frac{\partial^2 \langle s_i \rangle}{\partial \theta_j \partial w_{kl}}|_q d\theta_j dw_{kl} = (1 - m_i^2) \sum_j \left((1 - m_j^2) d\theta_j - 2m_i m_j d\theta_i \right) dw_{ij}$$

$$\sum_{jklm} \frac{\partial^2 \langle s_i \rangle}{\partial w_{jk} \partial w_{lm}}|_q dw_{jk} dw_{lm} = (1 - m_i^2) \sum_{jk} ((1 - m_k^2) m_j dw_{kj} dw_{ik}$$

$$+(1 - m_j^2) m_k dw_{jk} dw_{ij} - 2m_i \langle s_j s_k \rangle dw_{ij} dw_{ik})$$

Substituting this into Eq. 29 we obtain

$$0 = dm_i = (1 - m_i^2) \left(A_i - m_i A_i^2 + \sum_j (1 - m_j^2) w_{ij} A_j - m_i \sum_j w_{ij}^2 (1 - m_j^2) \right),$$

[*9] A comparison on a larger OCR problem was done in [45] which yields the same conclusion regarding the unsupervised methods. In this case, however, significant improvements have been reported using supervised methods (see `http://www.research.att.com/ yann/ocr/mnist/index.html`).

where we have defined $A_i = d\theta_i + \sum_j dw_{ij} m_j$. Since $A_i = 0 + \mathcal{O}(w^2)$, according to Eq. 54, we obtain

$$A_i = m_i \sum_j w_{ij}^2 (1 - m_j^2) + \mathcal{O}(w^3)$$

which is equivalent to Eq. 31.

§9 Acknowledgements

I would like to thank Wim Wiegerinck and Tom Heskes for useful discussions. This research was funded in part by the Dutch Technology Foundation (STW).

References

1) W.S. McCulloch and W. Pitts. *Bulletin of Mathematical biophysics*, 5:114–133, 1943.

2) F. Rosenblatt. *Psychological Review*, 65:386–408, 1958.

3) S. Amari. *IEEE Transactions on Electronic Computers*, 16:299–307, 1967.

4) D. Rumelhart, G. Hinton, and R. Williams. *Nature*, 323:533–536, 1986.

5) J. Hopfield. *Proceedings Nat. Acad. Sci. USA*, 79:2554–2558, 1982.

6) T. Kohonen. *Biological Cybernetics*, 43:59–69, 1982.

7) H. Sompolinsky and I. Kanter. *Physical Review Letters*, 57:2861–2864, 1986.

8) D. Ackley, G. Hinton, and T. Sejnowski. *Cognitive Science*, 9:147–169, 1985.

9) D.J. Amit, H. Gutfreund, and H. Sompolinsky. *Physical Review A*, 32:1007, 1985.

10) D.J. Amit, H. Gutfreund, and H. Sompolinsky. *Physical Review Letters*, 55:1530–1533, 1985.

11) A. Mason, A. Nicoll, and K. Stratford. *Journal of Neuroscience*, 11:72–84, 1990.

12) B. Katz. Nerve, muscle and synapse McGraw-Hill, 1966.

13) Chr. Koch. Biophysics of computation Oxford University Press, 1999.

14) L.F. Abbott, J.A. Varela, K. Sen, and S.B. Nelson. *Science*, pages 220–224, 1997.

15) H. Markram and M. Tsodyks. *Nature*, pages 807–810, 1996.

16) W. Rall and J. Rinzel. *Biophysics Journal*, pages 648–688, 1973.

17) W.A. Little. *Math. Biosci.*, 19:101–120, 1974.

18) P. Peretto. *An introduction to the modeling of neural networks.* Cambridge University Press, 1992.

19) H.J. Kappen. *Physical Review E*, 55:5849–5858, 1997. SNN–96–044, F–96–094.

20) G.R. Grimmett and D.R. Stirzaker. *Probability and random processes.* Clarendon Press, Oxford, 1992.

21) L. Saul and M.I. Jordan. *Neural Computation*, 6:1174–1184, 1994.

22) L. Onsager. *Journal of the American Chemical Society*, 58:1486–1493, 1936.

23) D. Sherrington and S. Kirkpatrick. *Physical review letters*, 35:1792–1796, 1975.

24) G. Toulouse. *Comm. Phys.*, 2:115–119, 1977.

25) A. Düring, A.C.C. Coolen, and D. Sherrington. *Journal of Physics A: Mathematics General*, 31:8607–8621, 1998.

26) D. Amit. *Modeling brain function*. Cambridge University Press, Cambridge, 1989.

27) J.D. Gutfreund, H. Reger and A.P. Young. *Journal of Physics A: Mathematical General*, 21:2775–2797, 1988.

28) A. Crisanti and H. Sompolinsky. *Physical Review A*, 37:4865–4874, 1988.

29) K. Nützel and U. Krey. *Journal of Physics A: Mathematical General*, 26:L591–L597, 1993.

30) H. Eissfeller and M. Opper. *Physical Review E*, 50:709–720, 1994.

31) E. Goles and G.Y. Vichniac. In J.S. Denker, editor, *Proceedings AIP conference*, pages 165–181. American Institute of Physics, 1986.

32) A. Crisanti, M. Falcioni, and A. Vulpiani. *Journal of Physics A: Mathematical General*, 26:3441–3453, 1993.

33) T. Plefka. *Journal of Physics A*, 15:1971–1978, 1982.

34) S.-I. Amari. *IEEE Transactions Neural Networks*, 3:260–271, 1992.

35) T. Tanaka. In M.S. Kearns, S.A. Solla, and D.A. Cohn, editors, *Advances in Neural Information Processing Systems 11*, pages 351–357. MIT Press, 1999.

36) D.J. Thouless, P.W. Anderson, and R.G. Palmer. *Phil. Mag.*, 35:593–601, 1977.

37) S.R. Kelso, A.H. Ganong, and T.H. Brouwn. *Proceedings National Academy of Science*, 83:5326–5330, 1986.

38) S. Kullback. *Information Theory and Statistics*. Wiley, New York, 1959.

39) J. Hertz, A. Krogh, and R. Palmer. *Introduction to the theory of neural computation*, volume 1 of *Santa Fe Institute*. Addison-Wesley, Redwood City, 1991.

40) G. Parisi. *Statistical Field Theory*. Frontiers in Physics. Addison-Wesley, 1988.

41) H.J. Kappen and F.B. Rodríguez. In M.S. Kearns, S.A. Solla, and D.A. Cohn, editors, *Advances in Neural Information Processing Systems 11*, pages 280–286. MIT Press, 1999.

42) H.J. Kappen and F.B. Rodríguez. *Neural Computation*, 10:1137–1156, 1998.

43) G.E. Hinton, P. Dayan, B.J. Frey, and R.M. Neal. *Science*, 268:1158–1161, 1995.

44) L.K. Saul, T. Jaakkola, and M.I. Jordan. *Journal of artificial intelligence research*, 4:61–76, 1996.

45) M. Leisink and H.J. Kappen. In *Proceedings IJCNN*, 2000. Submitted.

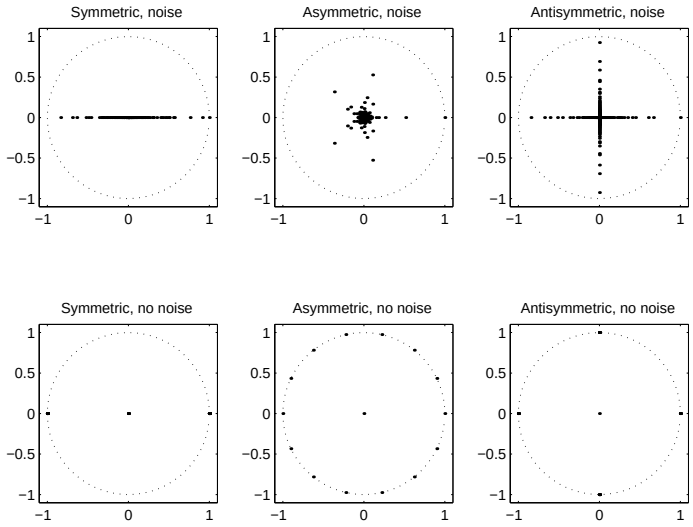

Fig. 4 Eigenvalues of the transition matrix T for the fully connected network with random symmetric, asymmetric and antisymmetric connections and $\theta_i = 0$. The eigenvalues are complex numbers λ, with $|\lambda| \leq 1$. There is always at least one eigenvalue $\lambda = 1$ (the Perron-Frobenius (PF) eigenvalue) and the corresponding right eigenvector of T is the stationary distribution. The Markov process is called periodic with periodicity d when T has eigenvalues $\lambda = \exp(2\pi i n/d), n = 0, \ldots, d-1$. See section 4 for additional details. Top row: Weights are drawn from a Gaussian distribution with mean zero and variance $1/n$. Due to the small weights the dynamics is rather noisy. All eigenvalues except for the PF eigenvalue are in the interior of the unit circle, which means that these modes do not survive asympotically. Bottom row: same weights as in the top row, but scaled $w_{ij} \to \beta w_{ij}$, with $\beta \to \infty$. In this case the dynamics is deterministic. We clearly see the limit cycles of period 2 for the symmetric case. Ergodicity is broken: in this example, the number of eigenvalues 1, 0 and -1 are 22, 216 and 18 respectively. Thus, there are 22 independent stationary distributions, of which 4 are fixed points and 18 are limit cycles of length 2. Also in the antisymmetric case ergodicity is broken: in this example, there are 204 eigenvalues 0 and 13 cycles of length 4 with eigenvalues $(1, i, -1, -i)$. In contrast, the asymmetric case is ergodic: there is only one eigenvalue 1, forming with the other 13 non-zero eigenvalues a limit cycle of length 14. These cycles persist forever.

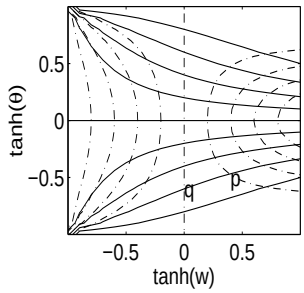

Fig. 5 Manifold of probability distributions $\mathcal{P}$ is computed for a Boltzmann-Gibbs distribution on two variables $p(s_1, s_2|w, \theta) = \dfrac{\exp(ws_1s_2 + \theta(s_1 + s_2))}{Z}$. Solid lines are lines of constant $\langle s_1 \rangle = \langle s_2 \rangle$. Broken lines are lines of constant $\langle s_1 s_2 \rangle$. Both (w, θ) and $(\langle s_1 \rangle, \langle s_1 s_2 \rangle)$ are coordinates systems of $\mathcal{P}$. $\mathcal{M}$ is given by the line $w = 0$. For any $p \in \mathcal{P}$, the closest $q \in \mathcal{M}$ satisfies $\langle s \rangle_q = \langle s \rangle_p$.

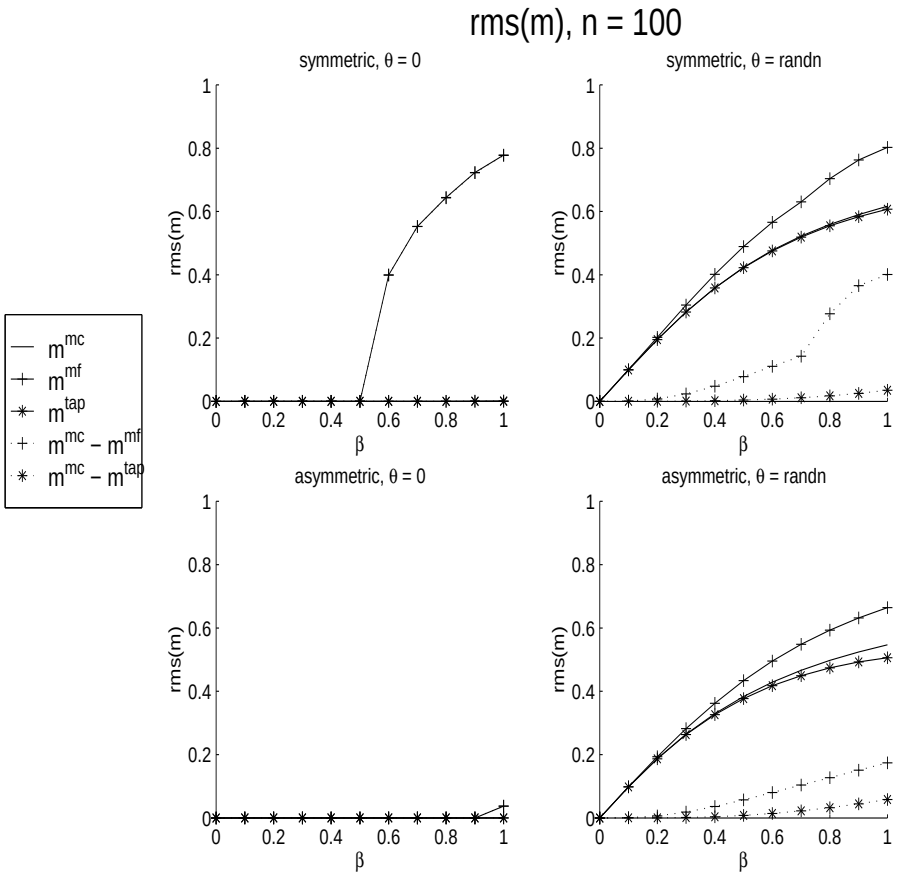

Fig. 6 Mean firing rates as a function of the strength of the connections for sequential dynamics, $n = 100$. RMS values of Monte Carlo results (—), first order approximation (+–), second order approximation (*–). RMS values of difference between first order approximation and MC value (+..) and difference between second order approximation and MC value (*..). We define $RMS^2(m) = \frac{1}{n}\sum_i^n m_i^2$.

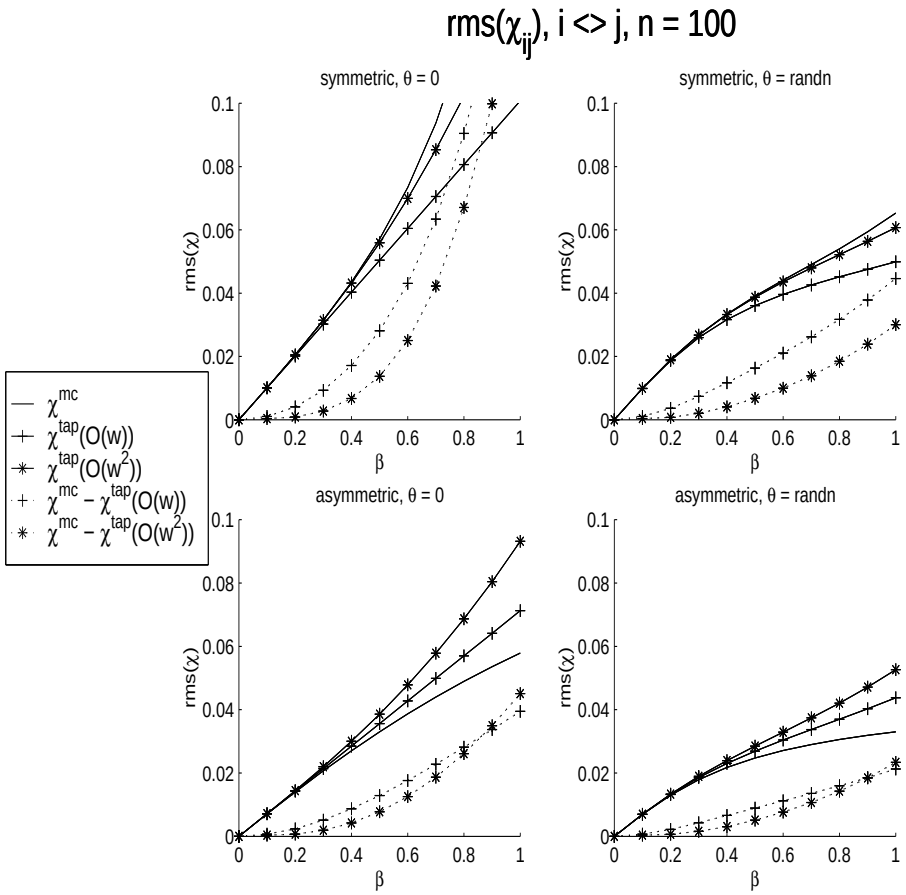

Fig. 7 Correlations as a function of the strength of the connections for sequential dynamics, $n = 100$. RMS values of Monte Carlo results (—), first order approximation (+–), second order approximation (*–). RMS values of difference between first order approximation and MC value (+..) and difference between second order approximation and MC value (*..). We define $RMS^2(\chi) = \frac{2}{n(n-1)} \sum_{i>j}^{n} \chi_{ij}^2$.

Fig. 8 Sample of patterns of the 8×8 handwritten digits of the U.S. Postal Service Office of Advanced Technology. In each row from left to right: the mean digit per class, a nice example and two rather bad examples.

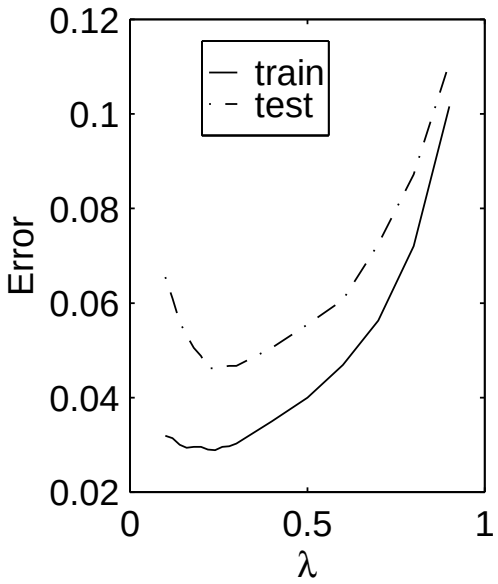

Fig. 9 Classification error of the Boltzmann Machine on the handwritten digits as a function of λ.