

Supplementary Information to the Mean-Field Theory for Randomly Connected Recurrent Networks of Threshold Gates

Thomas Natschläger

Software Competence Center Hagenberg, A-4232 Hagenberg, Austria

Thomas.Natschlaeger@scch.at

Nils Bertschinger

Max Planck Institute for Mathematics in the Sciences, D-04103 Leipzig, Germany

Nils.Bertschinger@mis.mpg.de

June 10, 2004

Abstract

In this technical report we describe the details to the mean-field theory and the synaptic scaling rule for self-organized criticality (SOC) discussed in [1].

1 The Network Model

We consider *input driven* recurrent networks consisting of N threshold gates with states $x_i \in \{0, 1\}, i = 1, \dots, N$. Each node i receives nonzero incoming weights w_{ij} from exactly K randomly chosen nodes j . Each nonzero connection weight w_{ij} is randomly drawn from a Gaussian distribution with mean μ and variance σ^2 . Furthermore, the network is *driven by an external input signal* $u_{(\cdot)}$ which is injected into each node. Hence, in summary, the update of the network state $\mathbf{x}_t = (x_{1,t}, \dots, x_{N,t})$ is given by

$$x_{i,t+1} = \Theta \left(\sum_{j=1}^N w_{ij} \cdot x_{j,t} + u_t \right) \text{ with } \Theta(h) = \begin{cases} 1 & \text{if } h \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

which is applied to all neurons in parallel.

In the following we consider a randomly drawn binary input signal $u_{(\cdot)}$: at each time step u_t assumes the value $\bar{u} + 1$ with probability r and the value \bar{u} with probability $1 - r$. Note that the $-\bar{u}$ can also be considered as the threshold of each node.

This network model is similar to the one we have considered in [2]. However it differs in two important aspects: a) By using states $x_i \in \{0, 1\}$ we emphasize the asymmetric information encoding by spikes prevalent in biological neural systems and b) it is more general in the sense that the Gaussian distribution from which the non-zero weights are drawn is allowed to have an arbitrary mean $\mu \in \mathbb{R}$. This implies that the network activity (see below) can vary

considerably for different parameters and enters all the calculations discussed in the rest of the paper.

2 Temporal Evolution of the Network activity

In this section we will give a detailed description of the function $A : [0, 1] \times \mathbb{R} \rightarrow [0, 1]$ which defines the update $a_{t+1} = A(a_t, u_t)$ of the network activity

$$a_t := \frac{1}{N} \sum_{i=1}^N x_{i,t}$$

In the limit $N \rightarrow \infty$ the network activity can be considered as the probability that a given unit i assumes the state 1, i.e. $a_t = \Pr \{x_{i,t} = 1\}$. Hence in order to calculate a_{t+1} we have to calculate

$$a_{t+1} = \Pr \{x_{i,t+1} = 1\} = \Pr \left\{ \sum_{j:w_{ij} \neq 0} w_{ij} \cdot x_{j,t} + u_t \geq 0 \right\}$$

Let us assume that the activity at time t is a_t and for the moment that there are exactly n out of the K inputs j' for which $x_{j',t-1} = 1$. Due to the Gaussian distribution (mean μ , variance σ^2) the term $\sum_{j'} w_{ij'} \cdot x_{j',t-1}$ is normally distributed with mean $n\mu$ and variance $n\sigma^2$. Hence we have for $n > 0$

$$\Pr \{x_{i,t+1} = 1 | n > 0\} = \Pr \left\{ \sum_{j:w_{ij} \neq 0} w_{ij} \cdot x_{j,t} + u_t \geq 0 \middle| n > 0 \right\} = 1 - \Phi(-u_t, n\mu, n\sigma^2)$$

where $\Phi(x, \mu, \sigma^2) = \int_{-\infty}^x \varphi(\xi, \mu, \sigma^2) d\xi = \frac{1}{2} (1 + \operatorname{erf}(\frac{-\mu+x}{\sqrt{2}\sigma}))$ denotes the Gaussian cumulative density function and $\varphi(x, \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2} \cdot \frac{(x-\mu)^2}{\sigma^2}}$ the Gaussian density function. For $n = 0$ this expression reduces to

$$\Pr \{x_{i,t+1} = 1 | 0\} = \Pr \{u_t \geq 0\} = \Theta(u_t)$$

In order to remove the condition on n we have to average over all possible values $0 \dots K$ of n . The probability of observing exactly n inputs with a value of 1 out of K possible is given by $\binom{K}{n} \cdot a_t^n \cdot (1 - a_t)^{K-n}$ since a_t is the probability of a unit at time t to have a value of 1. In summary we arrive at the following equation for the update of the network activity

$$\begin{aligned} a_{t+1} &= A(a_t, u_t) = \Pr \{x_{i,t+1} = 1\} \\ &= \sum_{n=0}^K \binom{K}{n} \cdot a_t^n \cdot (1 - a_t)^{K-n} \Pr \{x_{i,t+1} = 1 | n\} \\ A(a_t, u_t) &= (1 - a_t)^K \Theta(u_t) + \sum_{n=1}^K \binom{K}{n} \cdot a_t^n \cdot (1 - a_t)^{K-n} (1 - \Phi(-u_t, n\mu, n\sigma^2)) \quad (2) \end{aligned}$$

3 Temporal Evolution of the Hamming Distance

In this section we want to analyze the temporal evolution of the normalized Hamming distance¹

$$d_t := |\{i : x_{1,i,t} \neq x_{2,i,t}\}|/N$$

between two trajectories $\mathbf{x}_{1,(\cdot)}$ and $\mathbf{x}_{2,(\cdot)}$ which result by applying Equ. (1) to the two initial network states $\mathbf{x}_{1,0}$ and $\mathbf{x}_{2,0}$ and the input time series $u_{1,(\cdot)}$ and $u_{2,(\cdot)}$.

$$\begin{aligned} \text{trajectory 1 : } & \mathbf{x}_{1,0} \xrightarrow{u_{1,0}} \mathbf{x}_{1,1} \xrightarrow{u_{1,1}} \cdots \mathbf{x}_{1,t} \xrightarrow{u_{1,t}} \mathbf{x}_{1,t+1} \cdots \\ \text{trajectory 2 : } & \mathbf{x}_{2,0} \xrightarrow{u_{2,0}} \mathbf{x}_{2,1} \xrightarrow{u_{2,1}} \cdots \mathbf{x}_{2,t} \xrightarrow{u_{2,t}} \mathbf{x}_{2,t+1} \cdots \end{aligned}$$

As we have already shown in the previous section the temporal evolution of the network activity in each of the two trajectories $l = 1, 2$ can be described by $a_{l,t+1} = A(a_{l,t}, u_{l,t})$. In the rest of this section we will calculate an update function H of the form $d_{t+1} = H(d_t, a_{1,t}, a_{2,t}, u_{1,t}, u_{2,t})$ that describes the one-step evolution of d_t . According to the annealing approximation introduced by Derrida and others [3] the temporal evolution of d_t in the above trajectory can be obtained by applying the one-step update H successively. The basic assumption of the annealing approximation in our context is that at each time step the whole weight matrix is randomly re-generated. This has the effect that the updates at successive timesteps are independent so that correlations between units which would otherwise be built up over time must not be considered. Although this is a rather radical assumption the predictions obtained for the quenched case (weight matrix generated once) are quite accurate; see below. This is due to the fact that if the number of inputs to a unit is rather small compared to the network size ($K \ll \ln N$) then correlations between units build up very slowly and can be neglected in the limit $N \rightarrow \infty$ [4, 5].

To get started let us consider two fixed network states $\mathbf{x}_{1,t}$ and $\mathbf{x}_{2,t}$ and an arbitrary unit i (with its K nonzero input weights w_{ij}). With the help of the definitions

$$\begin{aligned} J_{11} &:= \{j : w_{ij} \neq 0 \wedge x_{1,j} = 1 \wedge x_{2,j} = 1\} \\ J_{01} &:= \{j : w_{ij} \neq 0 \wedge x_{1,j} = 0 \wedge x_{2,j} = 1\} \\ J_{10} &:= \{j : w_{ij} \neq 0 \wedge x_{1,j} = 1 \wedge x_{2,j} = 0\} \end{aligned}$$

the internal activations $h_{l,t} = \sum_j w_{ij} x_{l,j,t} + u_{l,t}$ of unit i at time t can be written as (index t omitted for brevity)

$$h_1 = \sum_j w_{ij} x_{1,j} + u_1 = \underbrace{\sum_{j \in J_{11}} w_{ij}}_{\xi} + \underbrace{\sum_{j \in J_{10}} w_{ij}}_v + u_1 = \xi + v + u_1 \quad (3)$$

$$h_2 = \sum_j w_{ij} x_{2,j} + u_2 = \underbrace{\sum_{j \in J_{11}} w_{ij}}_{\xi} + \underbrace{\sum_{j \in J_{01}} w_{ij}}_{\zeta} + u_2 = \xi + \zeta + u_2 \quad (4)$$

The probability $\Pr \{x_{1,i,t+1} \neq x_{2,i,t+1} | \mathbf{x}_{1,t}, \mathbf{x}_{2,t}, u_{1,t}, u_{2,t}\}$ that the states $x_{1,i,t+1}$ and $x_{2,i,t+1}$ of the node i differ at time $t+1$ (given $\mathbf{x}_{1,t}, \mathbf{x}_{2,t}$ which induce the sets J_{11}, J_{01}, J_{10} and the

¹ $x_{l,i,t}$ denotes the state of unit i at time t in trajectory l .

inputs $u_{1,t}, u_{2,t}$) is then given by:

$$\begin{aligned}
\Pr \{x_{1,i,t+1} \neq x_{1,i,t+1} | \mathbf{x}_{1,t}, \mathbf{x}_{2,t}, u_{1,t}, u_{2,t}\} &= \Pr \{\Theta(h_1) \neq \Theta(h_2)\} \\
&= \Pr \{h_1 \geq 0 \wedge h_2 < 0\} + \Pr \{h_1 < 0 \wedge h_2 \geq 0\} \\
&= \Pr \{\xi + v + u_1 \geq 0 \wedge \xi + \zeta + u_2 < 0\} + \\
&\quad + \Pr \{\xi + v + u_1 < 0 \wedge \xi + \zeta + u_2 \geq 0\} \\
\Pr \{x_{1,i,t+1} \neq x_{1,i,t+1} | \mathbf{x}_{1,t}, \mathbf{x}_{2,t}, u_{1,t}, u_{2,t}\} &= \Pr \{v \geq -\xi - u_1 \wedge \zeta < -\xi - u_2\} + \\
&\quad + \Pr \{v < -\xi - u_1 \wedge \zeta \geq -\xi - u_2\}
\end{aligned} \tag{5}$$

Taking into account the Gaussian distributions of the weights and defining $c := |J_{10}| + |J_{01}|$ (i.e. the number of bits which differ in the two states $\mathbf{x}_{1,t}$ and $\mathbf{x}_{2,t}$), $e := |J_{11}|$, and $f := |J_{10}|$ we know that the terms ξ , v and ζ are also Gaussian distributed with densities $\varphi(\xi, e\mu, e\sigma^2)$, $\varphi(v, f\mu, f\sigma^2)$ and $\varphi(\zeta, (c-f)\mu, (c-f)\sigma^2)$. Hence, for $c, e, f \neq 0$ we can calculate the probabilities in Equ. (5) as

$$\begin{aligned}
Q_1(c, e, f, u_1, u_2) &:= \Pr \{v \geq -\xi - u_1 \wedge \zeta < -\xi - u_2\} = \\
&\int_{-\infty}^{+\infty} (1 - \Phi(-\xi - u_1, f\mu, f\sigma^2)) \Phi(-\xi - u_2, (c-f)\mu, (c-f)\sigma^2) \varphi(\xi, e\mu, e\sigma^2) d\xi
\end{aligned} \tag{6}$$

$$\begin{aligned}
Q_2(c, e, f, u_1, u_2) &:= \Pr \{v < -\xi - u_1 \wedge \zeta \geq -\xi - u_2\} = \\
&\int_{-\infty}^{+\infty} \Phi(-\xi - u_1, f\mu, f\sigma^2) (1 - \Phi(-\xi - u_2, (c-f)\mu, (c-f)\sigma^2)) \varphi(\xi, e\mu, e\sigma^2) d\xi
\end{aligned} \tag{7}$$

and get as a result for the general case $c, e, f \neq 0$:

$$\begin{aligned}
Q(c, e, f, u_1, u_2) &:= \Pr \{x_{1,i,t+1} \neq x_{1,i,t+1} | \mathbf{x}_{1,t}, \mathbf{x}_{2,t}, u_{1,t}, u_{2,t}\} \\
&= \Pr \{x_{1,i,t+1} \neq x_{1,i,t+1} | c, e, f, u_{1,t}, u_{2,t}\} \\
&= Q_1(c, e, f, u_1, u_2) + Q_2(c, e, f, u_1, u_2)
\end{aligned} \tag{8}$$

Unfortunately there are a lot of special cases which arise at certain combinations of the values of c, e, f (mostly when one of this values is zero). For example a value of $c = 0$ implies that $f = 0$, i.e. $v = \zeta = 0$. For $e > 0$ we get

$$\begin{aligned}
Q(0, e, 0, u_1, u_2) &= \Pr \{x_{1,i,t+1} \neq x_{1,i,t+1} | 0, e, 0, u_1, u_2\} \\
&= \Pr \{\xi + u_1 \geq 0 \wedge \xi + u_2 < 0\} + \Pr \{\xi + u_1 < 0 \wedge \xi + u_2 \geq 0\} \\
&= \Pr \{-u_1 \leq \xi < -u_2\} + \Pr \{-u_2 \leq \xi < -u_1\} \\
&= \begin{cases} 0 & \text{if } u_1 = u_2 \\ \Phi(-u_2, e\mu, e\sigma^2) - \Phi(-u_1, e\mu, e\sigma^2) & \text{if } u_1 > u_2 \\ \Phi(-u_1, e\mu, e\sigma^2) - \Phi(-u_2, e\mu, e\sigma^2) & \text{if } u_2 > u_1 \end{cases}
\end{aligned} \tag{9}$$

while for $e = 0$ this reduces to

$$\begin{aligned}
Q(0, 0, 0, u_1, u_2) &= \Pr \{x_{1,i,t+1} \neq x_{1,i,t+1} | 0, 0, 0, u_1, u_2\} \\
&= \Pr \{u_1 \geq 0 \wedge u_2 < 0\} + \Pr \{u_1 < 0 \wedge u_2 \geq 0\} \\
&= \begin{cases} 1 & \text{if } \Theta(u_1) \neq \Theta(u_2) \\ 0 & \text{otherwise} \end{cases}
\end{aligned} \tag{10}$$

Other special cases are calculated in a similar manner and are detailed in [6].

In order to remove the condition on c, e, f we have to calculate the probability of a given triple (c, e, f) . This can be done using the given values of d, a_1 , and a_2 (index t omitted for brevity). Under the approximations made a Hamming distance of d can be interpreted as the probability that a certain component of two states \mathbf{x}_1 and \mathbf{x}_2 differs and $a_l = \Pr\{x_{l,i} = 1\}$. Let us first consider the relationship between the probabilities

$$\begin{aligned} p_{00} &:= \Pr\{x_{1,i} = 0 \wedge x_{2,i} = 0\} \\ p_{01} &:= \Pr\{x_{1,i} = 0 \wedge x_{2,i} = 1\} \end{aligned} \quad (11)$$

$$\begin{aligned} p_{10} &:= \Pr\{x_{1,i} = 1 \wedge x_{2,i} = 0\} \\ p_{11} &:= \Pr\{x_{1,i} = 1 \wedge x_{2,i} = 1\} \end{aligned} \quad (12)$$

and d, a_1 and a_2 . Obviously the following equations hold:

$$\begin{aligned} a_1 &= p_{11} + p_{10} \\ a_2 &= p_{11} + p_{01} \\ d &= p_{10} + p_{01} \\ 1 &= p_{00} + p_{01} + p_{10} + p_{11} \end{aligned} \quad (13)$$

Solving these equations for p_{01}, p_{10} and p_{11} gives

$$\begin{aligned} p_{01} &= \frac{-a_1 + a_2 + d}{2} \\ p_{10} &= \frac{a_1 - a_2 + d}{2} \\ p_{11} &= \frac{a_1 + a_2 - d}{2} \end{aligned}$$

With these probabilities we can calculate the probability of a triple (c, e, f) where we define

$$B(p, K, n) := \binom{K}{n} p^n (1-p)^{(K-n)}$$

as the binomial probability distribution.

To calculate the probability of a triple (c, e, f) we note the following:

- a) The (unconditional) probability of a certain value c is given by $B(d, K, c)$.
- b) The conditional probability p_e of observing a single input out of $K - c$ which has value 1 in both states given that c inputs have different values is $p_e = p_{11}/(p_{11} + p_{00}) = \frac{a_1 + a_2 - d}{2(1-d)}$. Hence the probability of observing exactly $e = |J_{11}|$, $0 \leq e \leq K - c$ inputs out of $K - c$ which have value 1 in both states given that c inputs have different values is $B(\frac{a_1 + a_2 - d}{2(1-d)}, K - c, e)$.
- c) The conditional probability p_f of observing a single input out of c which has value 1 in \mathbf{x}_1 and value 0 in \mathbf{x}_2 given that c inputs have different values is $p_f = p_{10}/(p_{10} + p_{01}) = \frac{a_1 - a_2 - d}{2d}$. Hence the probability of observing exactly $f := |J_{10}|$, $0 \leq f \leq c$ such inputs is $B(\frac{a_1 - a_2 - d}{2d}, c, f)$.

Altogether we have for the probability $P(c, e, f, d, a_1, a_2)$ of a given triple (c, e, f) :

$$P(c, e, f, d, a_1, a_2) = B(d, K, c) \cdot B\left(\frac{a_1 + a_2 - d}{2(1-d)}, K - c, e\right) \cdot B\left(\frac{a_1 - a_2 - d}{2d}, c, f\right) \quad (14)$$

Putting all the pieces together we end up with the formula

$$\begin{aligned} d_{t+1} &= \Pr\{x_{1,i,t+1} \neq x_{2,i,t+1}\} \\ &= \sum_{c=0}^K \sum_{e=0}^{K-c} \sum_{f=0}^c P(c, e, f, d_t, a_{1,t}, a_{2,t}) \Pr\{x_{1,i,t+1} \neq x_{2,i,t+1} | c, e, f\} \\ &= H(d_t, a_{1,t}, a_{2,t}, u_{1,t}, u_{2,t}) := \sum_{c=0}^K \sum_{e=0}^{K-c} \sum_{f=0}^c P(c, e, f, d_t, a_{1,t}, a_{2,t}) Q(c, e, f, u_{1,t}, u_{2,t}) \quad (15) \end{aligned}$$

for the update $d_{t+1} = H(d_t, a_{1,t}, a_{2,t}, u_{1,t}, u_{2,t})$ of the Hamming distance. Be aware that there are a lot of special cases for Q and that for $d_t = 0$ (which implies $c = 0$) and $d_t = 1$ (which implies $c = K$) this equation also assumes special forms (see [6] for details).

Note that in section 3 of [1] we considered the special case $u_{(\cdot)} = u_{1,(\cdot)} = u_{2,(\cdot)}$ when we discussed the ordered regime or equivalently fading memory. In this case we made the assumptions that the two trajectories $\mathbf{x}_{1,(\cdot)}$ and $\mathbf{x}_{2,(\cdot)}$ started from initial states with equal activity, i.e. $a_0 = a_{1,0} = a_{2,0}$, and a *consistent* initial Hamming distance d_0 . It can be calculated from Equ. (11) and Equ. (13) that only values of $d_0 \in [0, \min\{2a_0, 2(1-a_0)\}]$ are consistent with an initial value of a_0 . This leads to the formula

$$d_{t+1} = f(d_t, a_t, u_t) = H(d_t, a_t, a_t, u_t, u_t) \quad (16)$$

for the update of the Hamming distance for the “fading scenario”. In the general case for arbitrary $u_{1,(\cdot)}$ and $u_{2,(\cdot)}$ we used the notation

$$d_{t+1} = s(d_t, a_{1,t}, a_{2,t}, u_{1,t}, u_{2,t}) := H(d_t, a_{1,t}, a_{2,t}, u_{1,t}, u_{2,t}) \quad (17)$$

for the update of the Hamming distance for the “separation scenario”.

4 Calculation of a Criticality Criterion

To derive the criticality criterion we have to calculate

$$\alpha(a, u) := \left. \frac{\delta f(d, a, u)}{\delta d} \right|_{d=0}$$

The algebraic computations detailed in [6] show that $\alpha(a, u)$ is given by

$$\alpha(a, u) = \sum_{e=0}^{K-1} B(a, K-1, e) \cdot Q(1, e, 0, u, u)$$

which leads to the criticality criterion

$$\alpha^* := K \cdot \underbrace{\sum_{e=0}^{K-1} B(a^*, K-1, e) (rQ(1, e, 0, \bar{u}+1, \bar{u}+1) + (1-r)Q(1, e, 0, \bar{u}, \bar{u}))}_{P_{bf}} = 1 \quad (18)$$

or equivalently

$$P_{bf} = \frac{1}{K} \tag{19}$$

as stated in [1].

5 A Synaptic Scaling Rule for Self-Organized Criticality

Self-organized criticality would allow a network to adjust its dynamics towards criticality, i.e. the phase-transition between ordered and chaotic dynamics. Here a local rule is derived that achieves this in networks of randomly connected, input-driven threshold gates.

The criticality criterion defined by Equ. (18) and Equ. (19) shows that a local quantity, namely the one-bit-flip probability P_{bf} , can be used to predict the dynamics of a network. In the following a rule for self-organized criticality (SOC) will be derived by locally estimating this probability.

5.1 Estimating the one-bit-flip probability

The state update of a single threshold gate is given by:

$$y = \Theta \left(\sum_{i=1}^K w_i x_i + u \right)$$

where $x_i, y \in \{0, 1\}$ and Θ is defined in Equ. (1). $u \in \mathbb{R}$ is the external input. The margin m is defined as the distance to the decision boundary, i.e.

$$m = \left| \sum_{i=1}^K w_i x_i + u \right|$$

This local quantity will now be used to estimate the one-bit-flip probability P_{bf} . Intuitively the larger the margin the smaller the one-bit-flip probability since an input bit-flip has to affect the internal activation more to cross the decision boundary if the margin is large.

Consider the following setting: Let y denote the output of a node with input $x \in \{0, 1\}^K$, $u \in \mathbb{R}$ and y_{bf} the output when input x_f is flipped to $x_f^{bf} = 1 - x_f$, i.e. $x_i^{bf} = x_i, \forall i \neq f$ and $x_f^{bf} = 1 - x_f$.

To estimate P_{bf} we need to know whether $y = y_{bf}$ or not. To do this consider the internal

activations h and h_{bf} of the node given the two inputs

$$\begin{aligned}
h &= \sum_{i=1}^K w_i x_i + u \\
h_{bf} &= \sum_{i=1}^K w_i x_i^{bf} + u \\
&= \sum_{i=1, i \neq f}^K w_i x_i + w_f x_f^{bf} + u \\
&= \sum_{i=1}^K w_i x_i - w_f x_f + w_f x_f^{bf} + u \\
&= h - w_f x_f + w_f x_f^{bf} \\
&= h - w_f x_f + w_f(1 - x_f) \\
&= h + w_f(1 - 2x_f)
\end{aligned}$$

Now consider the following cases (note that $m = |h|$):

1. $y = 1$, i.e. $h \geq 0$ and $(1 - 2y) = -1$. Here we have $y_{bf} = \Theta(h_{bf}) \neq y = \Theta(h)$ iff

$$\begin{aligned}
h_{bf} &< 0 \\
h + w_f(1 - 2x_f) &< 0 \\
h &< -w_f(1 - 2x_f) \\
|h| &< -w_f(1 - 2x_f) \\
m &< -w_f(1 - 2x_f) \\
m &< (1 - 2y)w_f(1 - 2x_f)
\end{aligned}$$

2. $y = 0$, i.e. $h < 0$ and $(1 - 2y) = +1$. Here we have $y_{bf} = \Theta(h_{bf}) \neq y = \Theta(h)$ iff

$$\begin{aligned}
h_{bf} &\geq 0 \\
h + w_f(1 - 2x_f) &\geq 0 \\
h &\geq -w_f(1 - 2x_f) \\
-h &\leq w_f(1 - 2x_f) \\
|h| &\leq w_f(1 - 2x_f) \\
m &\leq w_f(1 - 2x_f) \\
m &\leq (1 - 2y)w_f(1 - 2x_f)
\end{aligned}$$

Combining these cases² we obtain:

$$y \neq y_{bf} \quad \text{if and only if} \quad w_f(1 - 2x_f)(1 - 2y) > m \quad (20)$$

²We assume $m < (1 - 2y)w_f(1 - 2x_f)$ for case 2 and ignore the pathological case $m = (1 - 2y)w_f(1 - 2x_f)$.

5.2 The SOC-rule

Equ. (20) will allow us to estimate P_{bf} as

$$\begin{aligned} P_{bf}^{MEst} &= \frac{1}{K} \sum_{i=1}^K (w_i(1 - 2x_i)(1 - 2y) > m) \\ &= \frac{1}{K} \sum_{i=1}^K \Theta(w_i(1 - 2x_i)(1 - 2y) - m) \end{aligned}$$

i.e. by summing up how often the condition Equ. (20) is true on average over all K inputs. Note that this estimate also handles the external input u implicitly by incorporating it into the margin. Therefore a time averaged version of P_{bf}^{MEst} will give the mean one-bit-flip probability for a given, stationary input signal $u(\cdot)$.

By the criticality criterion defined in Equ. (19) we get critical dynamics if

$$\begin{aligned} P_{bf}^{MEst} &= \frac{1}{K} \\ \frac{1}{K} \sum_{i=1}^K \Theta(w_i(1 - 2x_i)(1 - 2y) - m) &= \frac{1}{K} \\ \underbrace{\hspace{10em}}_{M_{est}} & \\ M_{est} &= 1 \end{aligned} \tag{21}$$

Since it is known that changing the variance σ^2 of the K incoming weights will change the network dynamics we use synaptic scaling to obtain the following SOC-rule where the time-averaged value of M_{est} is denoted by $\overline{M_{est}}$:

$$w_{ij}(k+1) = \begin{cases} (1 + \nu)w_{ij}(k) & \text{if } \overline{M_{est}} < 1 \\ \frac{1}{1+\nu}w_{ij}(k) & \text{if } \overline{M_{est}} > 1 \end{cases} \tag{22}$$

with a small learning rate $0 < \nu \ll 1$.

References

- [1] T. Natschläger, N. Bertschinger, and R. Legenstein. At the edge of chaos: Real-time computations and self-organized criticality in recurrent neural networks. submitted for publication, 2004.
- [2] N. Bertschinger and T. Natschläger. Real-time computation at the edge of chaos in recurrent neural networks. *Neural Computation*, 16(7):1413–1436, 2004.
- [3] B. Derrida and Y. Pomeau. Random networks of automata: A simple annealed approximation. *Europhys. Lett.*, 1:45–52, 1986.
- [4] B. Derrida. Dynamical phase transition in non-symmetric spin glasses. *J. Phys. A: Math. Gen.*, 20:721–725, 1987.

- [5] B. Derrida and G. Weisbuch. Evolution of overlaps between configurations in random boolean networks. *J. Physique*, 47:1297, 1986.
- [6] T. Natschläger and N. Bertschinger. A Mathematica notebook with an implementation of the mean-field theory for randomly connected recurrent networks of threshold gates, 2004. <http://www.igi.tugraz.at/tnatschl/edge-of-chaos/mean-field-notebook.nb>.