# Neural Networks and Biological Modeling

Professeur Wulfram Gerstner
Laboratory of Computational Neuroscience

## Correction Question Set 6

### Exercise 1: Synaptic Plasticity: the BCM rule

**1.1** First consider a presynaptic neuron $j_1$ taken from the first population. The dynamics of the strength of the synapse with neuron $i$ is such that

1. during the firing of the first population

$$\frac{d}{dt}w_{ij_1} = a_2^{\text{corr}}\Phi(\nu_i^{\text{post}} - 20\,\text{Hz})\nu_{j_1}^{\text{pre}}, \tag{1}$$

with

$$\nu_i^{\text{post}} = \sum_{1^{\text{st}}\,\text{population}} w_{ij}\nu_j^{\text{pre}} = 30\,\text{Hz}. \tag{2}$$

According to the graph in fig 1b we see that $\frac{d}{dt}w_{ij_1}$ is strictly positive. Since increased weights lead to increasing postsynaptic rate $\nu_i^{\text{post}}$ which in turn leads to increasing $\frac{d}{dt}w_{ij_1}$, the weights keep on increasing.

2. During the firing of the second population $\nu_{j_1}^{\text{pre}} = 0$ and hence $\frac{d}{dt}w_{ij_1} = 0$. The synaptic strengths stay fixed.

Therefore we conclude that the synapses of the first population of neurons become stronger with time.

Now consider a neuron $j_2$ belonging to the second population. During the firing of the first population, since $\nu_{j_2}^{\text{pre}} = 0$, we have

$$\frac{d}{dt}w_{ij_2} = 0 \tag{3}$$

During the firing of the second group, since initially we have $\nu_i^{\text{post}} = 10\,\text{Hz}$ which is below threshold, $\frac{d}{dt}w_{ij_2}$ is negative and will remain negative (see fig 1b) until the weights reach their minimum value (say zero – afterwards, weights are artificially kept null). The neuron $i$ has become selective to one of the two groups, thus achieving group discrimination (which is good for memory).

**1.2** The post-synaptic frequencies during the firing of the two groups are respectively 25 Hz and 30 Hz, which is above $\vartheta$. We are back to the situation described above, and the weights of both groups will grow indefinitely. No discrimination is made. One could argue that weights of group 1 will grow faster than those of group 2. This is true, but physically "growing without bound" actually means "growing until a limit is reached", and eventually the weights of both groups will end up being similar.

**1.3** The idea is to place the threshold $\vartheta$ (which is the border between depression and potentiation) always at the right point to allow discrimination. If the postsynaptic rate increases (in average), then the threshold should follow quickly. To set things, let us assume that $\theta = \langle\nu_i\rangle$ (averaged post-synaptic rate, which can be implemented with a standard low pass filter with long time constant). In the situation given in 1.2, the threshold would slide until it reaches $\frac{25+30}{2} = 27.5$, in which case the situation becomes analog to 1.1, allowing group selection. In fact, this explanation is simplistic

in the sense that the weights also change with time: as the threshold slides towards the average rate, the average rate also increases because weights increase in both groups. It can be shown that the threshold must vary more rapidly than $\langle \nu_i \rangle$. In practice, one usually takes $\theta = \langle \nu_i^2 \rangle$ (average *squared* post-synaptic rate).

## Exercise 2: Hopfield networks and Hebbian learning

**2.1** We begin by calculating the change of weights $\Delta w_{ij}^{\mu}$ induced by presenting pattern $\mu$ to the network for $0.5s$:

$$\Delta w_{ij}^{\mu} = \int_0^{0.5} \frac{d}{dt} w_{ij}^{\mu} dt = \int_0^{0.5} a_2(\nu_i^{\mu} - 10)(\nu_j^{\mu} - 10)dt = 0.5\ a_2(\nu_i^{\mu} - 10)(\nu_j^{\mu} - 10) = 0.5\ a_2\ 10^2\ p_i^{\mu} p_j^{\mu}. \tag{4}$$

The last equality is easily explained: for all $i$ and $\mu$ we see that if $p_i^{\mu} = 1$ then $\nu_i^{\mu} - 10 = 20 - 10 = 10 = 10p_i^{\mu}$, and similarly if $p_i^{\mu} = -1$ then $\nu_i^{\mu} - 10 = 0 - 10 = 10p_i^{\mu}$.

Thus, by choosing $a_2 = \frac{1}{50}$ and summing over all prototype presentations, we have $w_{ij}^{final} = \sum_{\mu} p_i^{\mu} p_j^{\mu}$, as we wanted. This exercise is intended to convince you that it is possible to learn memories in a fully interconnected network using a simple Hebbian learning rule.

**2.2** Just expanding the given learning rule, we have $\frac{d}{dt} w_{ij} = a_2\vartheta^2 - a_2\vartheta\nu_i^{\text{post}} - a_2\vartheta\nu_i^{\text{pre}} + a_2\nu_i^{\text{post}}\nu_i^{\text{pre}}$. To map this into the general formulation we choose: $a_0 = a_2\vartheta^2$, $a_1^{\text{pre}} = -a_2\vartheta$, $a_1^{\text{post}} = -a_2\vartheta$, $a_2^{\text{corr}} = a_2$.

**2.3** The learning is unsupervised, since the network learns implicit associations present in the input without any additional teaching signal, e.g. when the prototype being presented changed. We should note however, that learning should be limited to the first presentation of each pattern. During retrieval, we do not want the weights to change. Such a distinction of "learn a new pattern" and "retrieve a known pattern" could be triggered by a novelty related neuromodulator. In that sense, learning is not purely unsupervised.

## Exercise 3: Hopfield network with probabilistic upate

**3.1** We first split the neurons into two groups: those that *should be active*, ($p_i^3 = +1$) and those that should not be active. From $h_i(t_0) = p_i^3 m^3(t_0))$ we see that all neurons in the same group get the same input potential $h$.

We then get the following four expressions for the update dynamics:
Those that *should* be active and are active:
$P\{S_i(t+1) = +1 | h_i(t_0), p_i^3 = +1\} = g(h_i(t)) = g(p_i^3 m^3(t_0)) = g(+1 m^3(t_0)) = g(m^3(t))$
Those that *should* be active and are not active:
$P\{S_i(t+1) = -1 | h_i(t_0), p_i^3 = +1\} = 1 - g(m^3(t))$
Those that *should not* be active and are active:
$P\{S_i(t+1) = +1 | h_i(t_0), p_i^3 = -1\} = g(h_i(t)) = g(p_i^3 m^3(t_0)) = g(-1 m^3(t_0)) = g(-m^3(t))$
Those that *should not* be active and are not active:
$P\{S_i(t+1) = -1 | h_i(t_0), p_i^3 = -1\} = 1 - g(-m^3(t))$

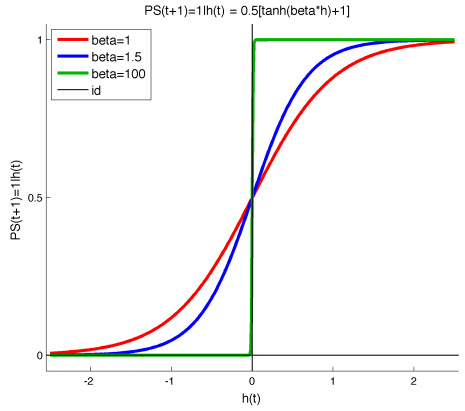
The expected number of neurons in each of the four groups is $N_+^3$ (or $N_-^3$) times the probabilities from above.

We start from
$$m(t+1) = \frac{1}{N} \sum_i^N p_i^3 S_i(t+1)$$
and split the sum into the four groups. Given $p \in \{-1, +1\}$ and $S_i(t+1) \in \{-1, +1\}$ note that we are just "counting". We can get an estimate of that sum by splitting it and replacing those "counts" by the expected number of neurons in each of the four groups (large N):

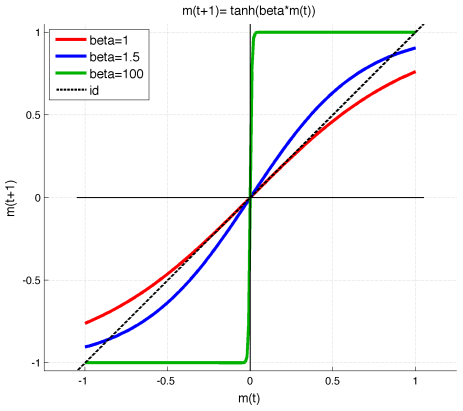
$$\begin{aligned} m(t+1) &= \frac{1}{N}\left[N_+^3 g(m^3(t)) \; - \; N_+^3(1 - g(m^3(t))\right] - \frac{1}{N}\left[\, N_-^3 g(-m^3(t)) \; - \; N_-^3(1 - g(-m^3(t))\right] \\ &= \tfrac{N_+^3}{N}\left[2g(m^3(t)) - 1\right] \; - \tfrac{N_-^3}{N}\left[2g(-m^3(t)) - 1\right] \\ &= g(m^3(t)) - g(-m^3(t)) \end{aligned}$$

Where we have used the assumption $P\{p_i^3 = 1\} = 0.5$ and that for a large network $N \to \infty$ we have $N_+^3 = N_-^3$ and $\frac{N_+^3}{N} = \frac{1}{2}$.

**3.2** g maps the input potential h onto a probability.
g: $\mathbb{R} \to [0, 1]$, monotonically increasing, symmetric around $g(0) = 0.5$
We plug $g(h) = \frac{1}{2}(tanh(\beta h) + 1)$ into $g(m^3(t)) - g(-m^3(t))$ and simplify it:
$m(t+1) = ... = tanh(\beta m^3(t))$.
See figure 1 to see the effect of $\beta$.




(a) **Stochastic update dynamics** for different levels of the inverse temperature $\beta$




(b) Evolution of the initial **overlap** in one time step. For $\beta \leq 1$ the fixed point is at 0 and any initial overlap will decrease. For $\beta = 1.5$ the overlap does not go beyond $\approx 0.85$: using a stochastic update, the network can only retrieve noisy versions of the pattern.

Figure 1: Update dynamics and overlap. Note the different domain and range of each graph. The lower the temperature, the more deterministic the update becomes and the fixed point of $m(t) \mapsto m(t+1)$ goes to 1 as $\beta \to \infty$.

## Exercise 4: Hopfield, the energy picture

In each time step only one neuron is updated (asynchronous dynamics). Let us assume that neuron $k$ has changed. The energy is given by:

$$E := -\sum_{i}^{N}\sum_{j}^{N} w_{ij}S_iS_j \tag{5}$$

We split that sum such that the contribution of the neuron $k$ to the energy is isolated from the other neurons:

$$\begin{aligned} E(t) &= \underbrace{-\sum_{j} w_{kj}S_k(t)S_j(t)}_{\text{i=k}} \underbrace{-\sum_{i} w_{ik}S_i(t)S_k(t)}_{\text{j=k}} - \textstyle\sum_{i\neq k}^{N}\sum_{j\neq k}^{N} w_{ij}S_i(t)S_j(t) \\ &= -2S_k(t)\textstyle\sum_j w_{kj}S_j(t) - \sum_{i\neq k}^{N}\sum_{j\neq k}^{N} w_{ij}S_i(t)S_j(t) \end{aligned}$$

The last equation comes from the symmetry of the weights and the fact that the first two sums run over the same range (1 to N).
For $E(t+1)$ we get the same expression but with the neuron k having a different state $S'_k = -S_k$ (all other neurons keep their value $S'_j = S_j$ for $j \neq k$.). When we look at the change in energy, all terms that do not depend on k cancel out. Therefore we have:

$$\begin{aligned} \Delta E = E(t+1) - E(t) &= -2S'_k\textstyle\sum_j w_{kj}S_j - (-2S_k\sum_j w_{kj}S_j) \\ &= -2(S'_k - S_k)\textstyle\sum_j w_{kj}S_j \end{aligned}$$

Because of the update of neuron $k$, we have $S'_k - S_k = 2S'_k$. Also, $\sum_j w_{kj}S_j \equiv h_k$. Thus, so far we have $\Delta E = -4S'_k h_k$.

Finally, due to the dynamics of the network, $S'_k = sign(h_k)$, the change in energy is
$\Delta E = -4\ h_k\ sign(h_k) < 0$.

In other words, the energy $E$ is a Liapunov function of the deterministic Hopfield network which decreases along trajectories. This yields the valuable insight that the network dynamics necessarily converge towards the minima of the energy function E.

## Exercise 5: Binary codes and spikes

**5.1** We do a change of variable and specify the Hopfield model:

- The **state** of a neuron $i$ is $\sigma_i \in \{0, 1\}$. It relates to $S_i$ by $S_i = 2\sigma_i - 1 \Leftrightarrow \sigma_i = \frac{1}{2}(S_i + 1)$.
- The **weights** are the same. They depend on the patterns, not on the state.
- For the **update dynamics** we rewrite eq. 6 in terms of $\sigma$:

$$S_i(t+1) = g(h_i(t)) = sign\left(\sum_{j=1}^{N} w_{ij} S_j(t)\right) \tag{6}$$

$$\begin{aligned} 2\sigma_i(t+1) - 1 &= sign\left(\sum_{j=1}^{N} w_{ij}(2\sigma_i(t) - 1)\right) \\ \sigma_i(t+1) &= \frac{1}{2}\left[sign\left(\sum_{j=1}^{N} w_{ij}(2\sigma_i(t) - 1)\right) + 1\right] \end{aligned}$$

**5.2** The property $\sum_{i=1}^{N} p_i = 0$ means that the patterns are balanced: they have the same number of active and inactive pixels. When patterns have specific statistical properties, these properties may translate into statistical properties of the weights. For that reason, we take the expression from the previous question, insert the definition of the weights and try to simplify it using the given property:

$$\begin{aligned} h_i(t) &= \sum_{j=1}^{N} w_{ij}(2\sigma_i(t) - 1) \\ &= 2\sum_{j=1}^{N} w_{ij}\sigma_i(t) - \sum_{j=1}^{N} w_{ij} \end{aligned}$$

$$\begin{aligned} \sum_{j=1}^{N} w_{ij} &= \frac{1}{N}\sum_{j=1}^{N}\sum_{\mu}^{M} p_i^\mu p_j^\mu \\ &= \tfrac{1}{N}\textstyle\sum_{\mu}^{M} p_i^\mu \sum_{j=1}^{N} p_j^\mu \\ &= \tfrac{1}{N}\textstyle\sum_{\mu}^{M} p_i^\mu 0 \\ &= 0 \end{aligned}$$

$$h_i(t) = 2\sum_{j=1}^{N} w_{ij}\sigma_i(t)$$

$$\begin{aligned} \sigma_i(t+1) &= \tfrac{1}{2}\left[sign\left(2\textstyle\sum_{j=1}^{N} w_{ij}\sigma_i(t)\right) + 1\right] \\ &= \frac{1}{2}\left[sign\left(\sum_{j=1}^{N} w_{ij}\sigma_i(t)\right) + 1\right] \end{aligned}$$

It's interesting to note that for balanced patterns, the weights also sum up to 0.