Solutions to exercises in Chapter 2

2.3 Cross-talk term. When we use Hebb's rule (2.25), the local field is obtained as $b_i^{(\nu)} = x_i^{(\nu)} + \frac{1}{N}\sum_{j=1}^{N}\sum_{\mu\neq\nu} x_i^{(\mu)} x_j^{(\mu)} x_j^{(\nu)}$, instead of Equation (2.28). This implies a slightly different definition of the cross-talk term. Equation (2.33) is replaced by:

$$C_i^{(\nu)} = -x_i^{(\nu)} \frac{1}{N} \sum_{j=1}^{N} \sum_{\mu\neq\nu} x_i^{(\mu)} x_j^{(\mu)} x_j^{(\nu)} . \tag{1}$$

Averaging over random patterns shows that the mean of $C_i^{(\nu)}$ is non-zero, $\langle C_i^{(\nu)}\rangle = -(p-1)/N \approx -p/N$ for large p. This means that the distribution of C is a shifted Gaussian, $P(C) = (2\pi\sigma_C)^{1/2} \exp[-(C-\langle C\rangle)^2/(2\sigma_C^2)]$, instead of Equation (2.36). For small $\alpha = p/N$, the mean tends to zero, so that the new distribution approaches Equation (2.36). For large values of α, the mean $\langle C\rangle$ dominates the error probability. In the limit $\alpha \to \infty$, the mean of $\langle \mathbb{W}\rangle = \frac{p}{N}\mathbb{I}$ dominates the network dynamics. The one-step error probability tends to zero in this limit because all states are reproduced, but the network cannot learn anything meaningful.

2.9 Energy function and synchronous dynamics. Consider the effect of one update following the synchronous rule (1.2). Divide the neurons into two sets: for all neurons in $\mathcal{S}$, $s_i' = s_i$, but for those in the complement $\mathcal{S}^c$ the state changed $s_i' = -s_i$. The corresponding change in the energy function (2.44) reads

$$H' - H = 2 \sum_{\substack{i\in\mathcal{S} \\ j\in\mathcal{S}^c}} w_{ij} s_i s_j . \tag{2}$$

Here it was assumed that the weights are symmetric, and that the diagonal weights vanish. Equation (2) simplifies to Equation (2.48) when $\mathcal{S}^c$ contains only one neuron, number m. Now assume that all neurons except the first one change, that is $\mathcal{S} = \{1\}$ and $\mathcal{S}^c = \{2,\ldots,N\}$. Note that this assumption fails for $N = 2$, where it corresponds to $\mathcal{S} = \{1\}$, $\mathcal{S}^c = \{2\}$. For $N = 2$, the update rules are $s_1' = \mathrm{sgn}(w_{12}s_2)$ and $s_2' = \mathrm{sgn}(w_{21}s_1)$. Since the weights are symmetric, either both neurons are updated, or none. As consequence we must require that $N > 2$. At any rate, the change in H is

$$H' - H = 2\sum_{j=2}^{N} w_{1j} s_1 s_j = 2\sum_{j=1}^{N} w_{1j} s_1 s_j = 2 s_1 b_1 . \tag{3}$$

In the second equality we used that $w_{11} = 0$. Since the state of the first neuron was assumed not to change, $\mathrm{sgn}(b_1) = s_1$. This means that $s_1 b_1 > 0$. We conclude that the energy function can increase under synchronous updates. Contrast this conclusion to the change of H under asynchronous McCulloc-Pitts updates. In this case $H' - H \leq 0$, as explained in Section 2.5.

Table 1: Distances and overlaps between the patterns shown in Figure 2.12. Exercise 2.12.

μ	ν	$d_{\mu\nu}$	$Q_{\mu\nu}$
1	1	0	1
1	2	13	$\frac{6}{32}$
1	3	2	$\frac{28}{32}$
1	4	32	-1
1	5	16	0
2	2	0	1
2	3	15	$\frac{2}{32}$
2	4	32	$-\frac{6}{32}$
2	5	19	$-\frac{6}{32}$

2.12 Recognising letters with a Hopfield network. Define

$$Q_{\mu\nu} = \frac{1}{N}\sum_{j=1}^{N} x_j^{(\mu)} x_j^{(\nu)}, \tag{4}$$

as in Equation (3.50). The bit j contributes with $+1$ to $Q_{\mu\nu}$ if $x_j^{(\mu)} = x_j^{(\nu)}$, and with -1 if $x_j^{(\mu)} \neq x_j^{(\nu)}$. Since there are $N = 32$ bits, we have $Q_{\mu\nu} = (32 - 2d_{\mu\nu})/32$, where $d_{\mu\nu}$ is Hamming distance, the number of bits by which the patterns $\boldsymbol{x}^{(\mu)}$ and $\boldsymbol{x}^{(\nu)}$ differ [Equation (2.2)]. Table 1 lists the values of $Q_{\mu\nu}$ extracted from Figure 2.12. Hebb's rule implies that

$$b_i^{(\nu)} = \sum_j w_{ij} x^{(\nu)} = x_i^{(1)} Q^{(1,\nu)} + x_i^{(2)} Q^{(2,\nu)}. \tag{5}$$

Applying the signum function, one finds

$$\begin{aligned} &\operatorname{sgn}(b_i^{(1)}) = x_i^{(1)}, \quad \operatorname{sgn}(b_i^{(2)}) = x_i^{(2)}, \quad \operatorname{sgn}(b_i^{(3)}) = x_i^{(1)}, \\ &\operatorname{sgn}(b_i^{(4)}) = -x_i^{(1)} = x_i^{(4)}, \quad \operatorname{sgn}(b_i^{(5)}) = -x_i^{(2)}. \end{aligned}$$

We conclude that patterns $\boldsymbol{x}^{(1)}$, $\boldsymbol{x}^{(2)}$ and $\boldsymbol{x}^{(4)}$ remain unchanged.

2.13 XOR function. Hebb's rule $w_{ij} = \frac{1}{N}\sum_\mu x_i^{(\mu)} x_j^{(\mu)}$ gives for the weight matrix

$$\mathbb{W} = \frac{4}{3}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{6}$$

Since the weight matrix is proportional to the unit matrix, all patterns are reproduced, not only the stored ones. The network cannot single out the XOR patterns because Hebb's rule is based on two-point correlations of the stored patterns, but three-point correlations are needed to learn the XOR patterns (Chapter **4**).

Solutions to exercises in Chapter 4

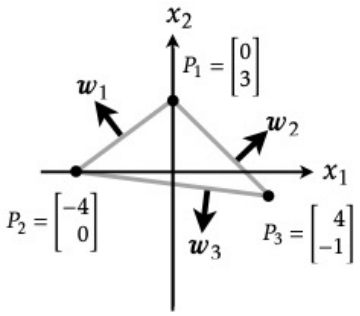
4.6 Kullback-Leibler divergence. We start from Equation (4.18)

$$\begin{aligned} D_{\mathrm{KL}} &= -\sum_{\boldsymbol{x}_\mu} P_{\mathrm{data}}(\boldsymbol{x}^{(\mu)}) \log[P_{\mathrm{B}}(\boldsymbol{s}=\boldsymbol{x}^{(\mu)})/P_{\mathrm{data}}(\boldsymbol{x}^{(\mu)})] \\ &\geq -\sum_{\boldsymbol{x}_\mu} P_{\mathrm{data}}(\boldsymbol{x}^{(\mu)}) [P_{\mathrm{B}}(\boldsymbol{s}=\boldsymbol{x}^{(\mu)})/P_{\mathrm{data}}(\boldsymbol{x}^{(\mu)}) - 1], \\ &= \sum_{\boldsymbol{x}_\mu} [P_{\mathrm{data}}(\boldsymbol{x}^{(\mu)}) - P_{\mathrm{B}}(\boldsymbol{s}=\boldsymbol{x}^{(\mu)})]. \end{aligned} \tag{7}$$

For the second step we used the inequality $\log z \leq z-1$. Using normalisation of the distributions, $1 = \sum_\mu P_{\mathrm{data}}(\boldsymbol{x}^{(\mu)}) = \sum_\mu P_{\mathrm{data}}(\boldsymbol{s}=\boldsymbol{x}^{(\mu)})$, it follows that $D_{\mathrm{KL}} \geq 0$. One verifies that D_{KL} attains the global minimum $D_{\mathrm{KL}}=0$ by setting $P_{\mathrm{data}}(\boldsymbol{x}^{(\mu)}) = P_{\mathrm{B}}(\boldsymbol{s}=\boldsymbol{x}^{(\mu)})$ in Equation (7).
To show that minimusing D_{KL} corresponds to maximising the log-likelihood $\mathscr{L}$, Equation (4.17), one starts from

$$D_{\mathrm{KL}} = \sum_{\boldsymbol{x}_\mu} P_{\mathrm{data}}(\boldsymbol{x}^{(\mu)}) \log[P_{\mathrm{data}}(\boldsymbol{x}^{(\mu)})] - \langle \log P_{\mathrm{B}}(\boldsymbol{s}=\boldsymbol{x}^{(\mu)}) \rangle_{P_{\mathrm{data}}}. \tag{8}$$

The first term is a constant, it does not depend on the weights. The second term equals $-p^{-1}\mathscr{L}$. So minimising D_{KL} corresponds to maximising $\mathscr{L}$.



V_1	V_2	V_3	target
0	0	0	1
0	0	1	0
0	1	0	0
1	0	0	0
0	1	1	0
1	0	1	0
1	1	0	0
1	1	1	-

Figure 1: (**a**) Input plane for Exercise 5.6 (schematic). (**b**) Value table for Exercise 5.6.

Solutions to exercises in Chapter 5

5.6 Linearly inseparable problem. A possible solution for the hidden neurons is shown in Figure 1. The weight vectors are

$$\boldsymbol{w}_1 = \begin{bmatrix} -3 \\ 4 \end{bmatrix}, \quad \boldsymbol{w}_2 = \begin{bmatrix} 4 \\ 4 \end{bmatrix}, \quad \boldsymbol{w}_3 = \begin{bmatrix} -1 \\ -8 \end{bmatrix}. \tag{9}$$

The thresholds are read off from the intersections of the decision boundary with the x_2-axis [Equation (**5.13**)]: $\theta_1 = 12$, $\theta_2 = 12$, $\theta_3 = 4$. A possible choice for the weights and threshold of the output neuron can be read off from the value table in Figure 1: $\boldsymbol{W} = [-1,-1,-1]^\mathsf{T}$ and $\Theta = \frac{1}{2}$, so that $O = \theta_\mathrm{H}(-V_1 - V_2 - V_3 + 1)$.

Solutions to exercises in Chapter 6

6.1 Covariance matrix. This is demonstrated on page 105. Since matrix $\mathbb{C} = \langle \delta\boldsymbol{x}\delta\boldsymbol{x}^\mathsf{T}\rangle$ is symmetric, it has a complete orthonormal basis of eigenvectors $\boldsymbol{u}_\alpha$. This allows us to write $\mathbb{C} = \sum_\alpha \boldsymbol{u}_\alpha \boldsymbol{u}_\alpha^\mathsf{T}$ with real eigenvalues λ_α. To show that the eigenvalues are non-negative, we use $\lambda_\beta = \boldsymbol{u}_\beta^\mathsf{T}\mathbb{C}\boldsymbol{u}_\beta$ to show that $\lambda_\beta = \langle(\delta\boldsymbol{x}^\mathsf{T}\boldsymbol{u}_\beta)^2\rangle \geq 0$. Here we used that the scalar product (2.14) is symmetric, $\delta\boldsymbol{x}^\mathsf{T}\boldsymbol{u}_\beta = \boldsymbol{u}_\beta^\mathsf{T}\delta\boldsymbol{x}$.