CHALMERS, GÖTEBORGS UNIVERSITET

EXAM for
ARTIFICIAL NEURAL NETWORKS

COURSE CODES: **FFR 135, FIM 720 GU, PhD**

Time:	January 4 (2023), at $14^{00} - 18^{00}$
Place:	Johanneberg
Teacher:	Bernhard Mehlig, 073-420 0988 (mobile)
Allowed material:	Book B. Mehlig, *Machine Learning with Neural Networks, CUP*
Not allowed:	Any other written material, calculator

Maximum score on this exam: 12 points.
Maximum score for homework problems: 12 points.
To pass the course it is necessary to score at least 5 points on this written exam.
CTH >13.5 passed; >17 grade 4; >21.5 grade 5,
GU >13.5 grade G; > 19.5 grade VG.

---

1. One-step error probability. In this question, consider a deterministic Hopfield network with weights given by

$$w_{ij} = \frac{1}{N}\sum_{\mu=1}^{p} x_i^{(\mu)} x_j^{(\mu)}, \tag{1}$$

where the diagonal entries are *non-zero*, and the patterns $\boldsymbol{x}^{(\mu)}$ are random bits such that

$$\text{Prob}(x_i^{(\mu)} = \pm 1) = \frac{1}{2}. \tag{2}$$

The local field is given by

$$b_i = \sum_{j=1}^{N} w_{ij} s_j. \tag{3}$$

Feeding an arbitrary stored pattern $\boldsymbol{x}^{(\nu)}$ to the network (i.e. by setting $s_j = x_j^{(\nu)}$), and updating a single bit, what is the probability of the bit changing sign? This probability is explored in the following subquestions.
(a) Derive the cross-talk term $C_i^{(\nu)}$, defined such that an error occurs when $C_i^{(\nu)} > 1$. Start from $b_i = \sum_{j=1}^{N} w_{ij} x_j^{(\nu)}$ (**0.5**p).

Answer: 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:

$$C_i^{(\nu)} = -x_i^{(\nu)} \frac{1}{N} \sum_{j=1}^{N} \sum_{\mu\neq\nu} x_i^{(\mu)} x_j^{(\mu)} x_j^{(\nu)} . \tag{4}$$

(b) Assuming N and p large, compute the mean value of $C_i^{(\nu)}$ (**1**p).
Answer: Average over the independent patterns, using that $\langle x_i^{(\nu)} x_i^{(\mu)} x_j^{(\mu)} x_j^{(\nu)}\rangle = 0$ when $i \neq j$ and $\mu \neq \nu$, because the average factorises in this case, and $\langle x_k^{(\mu)}\rangle = 0$. When $i = j$, there are $p-1$ terms that average to $\langle [x_j^{(\nu)}]^2 [x_j^{(\mu)}]^2\rangle = 1$. Thus, we conclude that $\langle C_i^{(\nu)}\rangle = -(p-1)/N \approx -p/N$ for large p.
(c) Using the central-limit theorem, one can show that the distribution of $C_i^{(\nu)}$ is

$$P(C_i^{(\nu)}) = (2\pi\sigma_C^2)^{-1/2} \exp\Big[-(C_i^{(\nu)} - \langle C_i^{(\nu)}\rangle)^2/(2\sigma_C^2)\Big], \tag{5}$$

where $\langle C_i^{(\nu)}\rangle$ is the mean value computed in the previous subquestion, and σ_C^2 is the variance of the distribution of $C_i^{(\nu)}$. Using the result from (b), describe what happens to the one-step error probability in the limit where $p \gg N$. (**0.5**p).
Answer: Due to the central limit theorem, 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 the weight matrix, $\langle \mathbb{W}\rangle = \frac{p}{N}\mathbb{I}$, 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. Linearly inseparable problem. A classification problem is given in Figure 1. Inputs $\boldsymbol{x}^{(\mu)}$ inside the grey region have targets $t^\mu = 1$, inputs outside the grey region have targets $t^\mu = -1$. The problem can be solved by a perceptron with a hidden layer with four neurons $V_j^{(\mu)} = \operatorname{sgn}\left(-\theta_j + \sum_{k=1}^{2} w_{jk} x_k^{(\mu)}\right)$, for $j = 1, \ldots, 4$. The output is computed as $O^{(\mu)} = \operatorname{sgn}\left(-\Theta + \sum_{j=1}^{4} W_j V_j^{(\mu)}\right)$. Find the weights w_{jk}, W_j, and thresholds θ_j, Θ that solve the classification problem (**2**p).

Answer: We set the rows in the 4×2 weight matrix $\mathbb{W}^{(1)}$ leading from the input layer to the hidden layer to be normal vectors to the decision boundaries

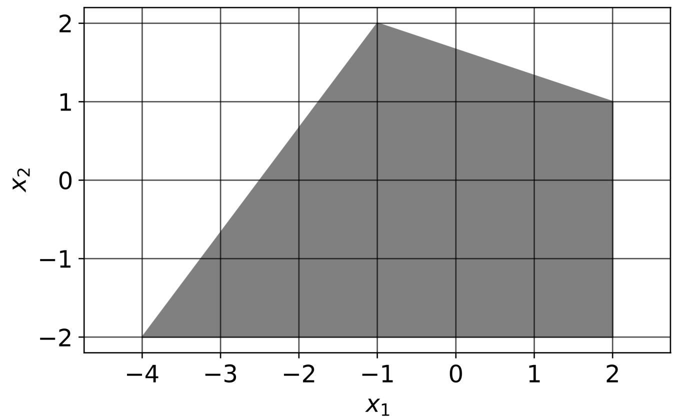

Figure 1: Classification problem for question 2.

pointing towards the origin:

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

Using that the decision boundary is parametrized by $w_{i1}^{(1)}x_1 + w_{i2}^{(1)}x_2 = \theta_i^{(1)}$, we pick a point on the i:th decision boundary to find the threshold. This gives $\theta_1^{(1)} = -2$, $\theta_2^{(1)} = -2$, $\theta_3^{(1)} = -\frac{5}{3}$, $\theta_4^{(1)} = -\frac{5}{6}$. Setting all elements in the 1×4 weight matrix $\mathbb{W}^{(2)}$ connecting the hidden layer to the output layer to 1, we know that the sum $\sum_{i=1}^{4} w_i^{(2)} V_i$, where V_i is the output from the i:th hidden neuron, will only take its maximal value of 4 when the input coordinate is inside the grey region. Otherwise, it will be less than or equal to 2. Thus, we pick the threshold $\theta^{(2)}$ to be a value between 2 and 4, say 3.

3. Backpropagation. Derive the update rules for the weights and thresholds of a one-layer perceptron with two input neurons, M hidden neurons, and three output neurons. The activation function $g(b)$ is used for all neurons. The outputs from the input layer, hidden layers, and output layers, are denoted x_k, V_j, and O_i respectively. The weights leading from the input layer to the hidden layer are denoted w_{jk} and the weights leading from the hidden layer to the output layer are denoted W_{ij}. The thresholds for the hidden and output layer are denoted θ_i and Θ_i respectively. Consider the energy function $H = \sum_{\mu=1}^{p} E(\mathbf{t}^{(\mu)}, \mathbf{O}^{(\mu)})$, where $E(\mathbf{t}^{(\mu)}, \mathbf{O}^{(\mu)})$ is a differentiable scalar function that depends on the targets $\mathbf{t}^{(\mu)}$ and outputs $\mathbf{O}^{(\mu)}$, and which reaches its minimum when $\mathbf{t}^{(\mu)} = \mathbf{O}^{(\mu)}$. (**2**p).

Answer: We start by deriving the update rules for the output weights. The update rule is

$$W'_{mn} = W_{mn} + \delta W_{mn}, \quad \delta W_{mn} = -\eta \frac{\partial H}{\partial W_{mn}}. \tag{7}$$

Performing the differentiation, we have

$$\frac{\partial H}{\partial W_{mn}} = \sum_{\mu=1}^{p} \frac{\partial E(\mathbf{t}^{(\mu)}, \mathbf{O}^{(\mu)})}{\partial W_{mn}} = \sum_{\mu=1}^{p} \sum_{i=1}^{3} \frac{\mathrm{d}E}{\mathrm{d}O_i^{(\mu)}} g'(B_i^{(\mu)}) \sum_{j=1}^{M} \frac{\partial W_{ij}}{\partial W_{mn}} V_j^{(\mu)} \tag{8}$$

where $B_i^{(\mu)}$ is the local field of the i:th output neuron. Using that $\frac{\partial W_{ij}}{\partial_{mn}} = \delta_{im}\delta_{jn}$, we obtain

$$\frac{\partial H}{\partial W_{mn}} = \sum_{\mu=1}^{p} \frac{\mathrm{d}E}{\mathrm{d}O_m^{(\mu)}} g'(B_m^{(\mu)}) V_n^{(\mu)} = \sum_{\mu=1}^{p} \Delta_m^{(\mu)} V_n^{(\mu)} \tag{9}$$

where $\Delta_m^{(\mu)} = \frac{\mathrm{d}E}{\mathrm{d}O_m^{(\mu)}} g'(B_m^{(\mu)})$. Hence, the update rule for the output weights is

$$\delta W_{mn} = -\eta \sum_{\mu=1}^{p} \Delta_m^{(\mu)} V_n^{(\mu)}. \tag{10}$$

Similarly, the update rule for the output thresholds is calculated to be

$$\delta \Theta_m = \eta \sum_{\mu=1}^{p} \Delta_m^{(\mu)}. \tag{11}$$

The update rule for the weights leading from the input to the hidden layer is given by

$$w'_{mn} = w_{mn} + \delta w_{mn}, \quad \delta w_{mn} = -\eta \frac{\partial H}{\partial w_{mn}}. \tag{12}$$

Performing the derivative, we obtain

$$\frac{\partial H}{\partial w_{mn}} = \sum_{\mu=1}^{p} \sum_{i=1}^{3} \Delta_i^{(\mu)} \sum_{j=1}^{M} W_{ij} \frac{\partial V_j^{(\mu)}}{\partial w_{mn}} = \sum_{\mu=1}^{p} \sum_{i=1}^{3} \Delta_i^{(\mu)} \sum_{j=1}^{M} W_{ij} g'(b_j^{(\mu)}) \sum_{k=1}^{2} \frac{\partial w_{jk}}{\partial w_{mn}} x_k^{(\mu)} \tag{13}$$

which simplifies to

$$\frac{\partial H}{\partial w_{mn}} = \sum_{\mu=1}^{p} \sum_{i=1}^{3} \Delta_i^{(\mu)} W_{im} g'(b_m^{(\mu)}) x_n^{(\mu)} = \sum_{\mu=1}^{p} \delta_m^{(\mu)} x_n^{(\mu)} \tag{14}$$

where $\delta_m^{(\mu)} = \sum_{i=1}^{3} \Delta_i^{(\mu)} W_{im} g'(b_m^{(\mu)})$. Thus, the update rule for the input weights is

$$\delta w_{mn} = -\eta \sum_{\mu=1}^{p} \delta_m^{(\mu)} x_n^{(\mu)}. \tag{15}$$

The update rule for the hidden thresholds take a similar form:

$$\delta\theta_m = \eta \sum_{\mu=1}^{p} \delta_m^{(\mu)}. \tag{16}$$

4. Convolutional network. The two patterns shown in Figure 2(a) are processed by a simple convolutional neural network that has one convolution layer with one single 3×3 kernel with ReLU units, zero threshold, and weights as given in Figure 2(b). Stride $(1, 1)$. The resulting feature map is fed into a 3×3 max-pooling layer with stride $(1, 1)$. Finally, there is a fully connected classification layer with two output units with Heaviside activation functions.
(a) For both patterns determine the resulting feature map and the output of the max-pooling layer (**1**p).
Answer: The feature map for patterns (a) and (b) are

$$(a): \begin{bmatrix} 1 & 3 & 1 \\ 3 & 4 & 3 \\ 2 & 7 & 2 \\ 2 & 6 & 2 \\ 1 & 4 & 1 \end{bmatrix}, \quad (b): \begin{bmatrix} 2 & 4 & 1 \\ 3 & 4 & 3 \\ 2 & 6 & 1 \\ 3 & 4 & 3 \\ 2 & 4 & 1 \end{bmatrix}$$

and the outputs of the max-pooling layers are

$$(a): \begin{bmatrix} 7 \\ 7 \\ 7 \end{bmatrix}, \quad (b): \begin{bmatrix} 6 \\ 6 \\ 6 \end{bmatrix}$$

(b) Determine weights and thresholds of the classification layer that allow to classify the two patterns into different classes (**1**p).
Answer: By picking output weights as $W = [1, 1, 1]$, the outputs for patterns (a) and (b) will be 21 and 18 respectively. Thus, it suffices to choose a threshold between 21 and 18 to successfully classify the different patterns, say $\theta = 20$.

5. Oja's rule. The aim of unsupervised learning is to construct a network that learns the properties of a distribution $P(\mathbf{x})$ of input patterns $\mathbf{x} = (x_1, \ldots, x_N)^\top$. Consider a network with one linear output function $y = \sum_{j=1}^{N} w_j x_j$. Under Oja's learning rule $\delta w_i = \eta y(x_i - y w_i)$ the weight vector $\mathbf{w}$ converges to a steady state $\mathbf{w}^*$ with components w_j^*.
(a) Show that the steady state $\mathbf{w}^*$ is an eigenvector of the matrix $\mathbb{C}'$ with elements $C'_{ij} = \langle x_i x_j \rangle$. Here $\langle \ldots \rangle$ denotes the average over $P(\mathbf{x})$ (**1**p).
(a) Show that the steady state $\mathbf{w}^*$ is an eigenvector of the matrix $\mathbb{C}'$ with elements $C'_{ij} = \langle x_i x_j \rangle$. Here $\langle \ldots \rangle$ denotes the average over $P(\mathbf{x})$. (**1**p).

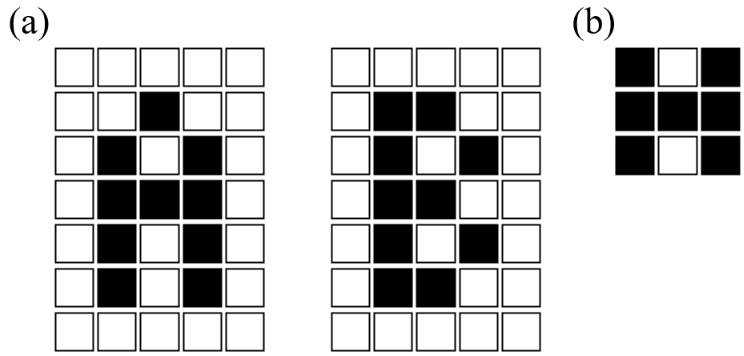

Figure 2: Patterns for question 4.

Answer: We start with the given learning rule written in vector notation:

$$\begin{aligned}\delta\mathbf{w} &= \eta y(\mathbf{x} - y\mathbf{w})\\ &= \eta(\mathbf{x}y - y^2\mathbf{w})\\ &= \eta[\mathbf{x}\mathbf{x}^\top\mathbf{w} - (\mathbf{w}^\top\mathbf{x}\mathbf{x}^\top\mathbf{w})\mathbf{w}]\end{aligned}$$

where in the last line we have used $y = \mathbf{w}^\top\mathbf{x} = \mathbf{x}^\top\mathbf{w}$, which yields $y^2 = yy = \mathbf{w}^\top\mathbf{x}\mathbf{x}^\top\mathbf{w}$. Now, by averaging $\delta\mathbf{w}$ over the data distribution, we get

$$\langle\delta\mathbf{w}\rangle = \eta[\langle\mathbf{x}\mathbf{x}^\top\rangle\mathbf{w} - (\mathbf{w}^\top\langle\mathbf{x}\mathbf{x}^\top\rangle\mathbf{w})\mathbf{w}]. \tag{17}$$

Let $\mathbb{C}' = \langle\mathbf{x}\mathbf{x}^\top\rangle$. Then, using the above equation, we have

$$\langle\delta\mathbf{w}\rangle = \eta[\mathbb{C}'\mathbf{w} - (\mathbf{w}^\top\mathbb{C}'\mathbf{w})\mathbf{w}]. \tag{18}$$

Now assume that $\mathbf{w} = \mathbf{w}^*$ is the normalized maximal eigenvector of the matrix $\mathbb{C}'$; that is, $\mathbb{C}'\mathbf{w}^* = \lambda_1\mathbf{w}^*$ where $(\mathbf{w}^*)^\top\mathbf{w} = 1$ and λ_1 is the maximal eigenvalue. Then

$$\begin{aligned}\langle\delta\mathbf{w}\rangle &= \eta[\mathbb{C}'\mathbf{w}^* - ((\mathbf{w}^*)^\top\mathbb{C}'\mathbf{w}^*)\mathbf{w}^*]\\ &= \eta[\lambda_1\mathbf{w}^* - \lambda_1((\mathbf{w}^*)^\top\mathbf{w}^*)\mathbf{w}^*]\\ &= \eta[\lambda_1\mathbf{w}^* - \lambda_1\mathbf{w}^*]\\ &= 0\end{aligned}$$

which proves that the eigenvector $\mathbf{w}^*$ is a steady state of the learning dynamics.

(b) Show that the matrix $\mathbb{C}'$ has non-negative eigenvalues (**1**p).

Answer: Given an eigenvector $\mathbf{v}$ of $\mathbb{C}'$ we have

$$\mathbf{v}^\top\mathbb{C}'\mathbf{v} = \mathbf{v}^\top\langle\mathbf{x}\mathbf{x}^\top\rangle\mathbf{v} = \lambda\mathbf{v}^\top\mathbf{v}. \tag{19}$$

This can be rewritten as

$$\langle\mathbf{v}^\top\mathbf{x}\mathbf{x}^\top\mathbf{v}\rangle = \langle(\mathbf{v}^\top\mathbf{x})^2\rangle = \lambda||\mathbf{v}||^2. \tag{20}$$

Table 1: Three-point probabilities for the data set shown in Figure 3(**a**,**b**).

x_1	x_2	x_3	$P(x_1, x_2, x_3)$
1	1	1	$\frac{4}{14}$
1	1	-1	$\frac{1}{14}$
1	-1	1	$\frac{1}{14}$
-1	1	1	$\frac{1}{14}$
1	-1	-1	$\frac{1}{14}$
-1	1	-1	$\frac{1}{14}$
-1	-1	1	$\frac{1}{14}$
-1	-1	-1	$\frac{4}{14}$

Hence, the eigenvalues are given by

$$\lambda = \frac{\langle (\mathbf{v}^\top \mathbf{x})^2 \rangle}{||\mathbf{v}||^2} \geq 0. \tag{21}$$

6. Restricted Boltzmann machine. Demonstrate that a Boltzmann machine requires hidden units to learn the 3×3 data set shown in Figure 3(**a**). Evaluate all eight three-point probabilities $P(x_1 = \pm 1, x_2 = \pm 1, x_3 = \pm 1)$ for x_1, x_2, and x_3 as shown in panel (**b**). Here $x_j = +1$ represents $\blacksquare$, and $x_j = -1$ stands for $\square$. Check whether these three-point probabilities factorise. For example, does $P(x_1 = 1, x_2 = 1, x_3 = -1) = P(x_1 = 1, x_2 = 1) P(x_3 = -1)$ hold, or not? Use your results to explain why a Boltzmann machine needs hidden units to learn the data set (**a**). Now consider the data set in Figure 3(**c**), only stripes. Explain why no hidden units are needed for (**c**) (**2p**).

Answer: The eight three-point probabilities $P(x_1 = 1, x_2 = 1, x_3 = -1)$ for the data set are listed in Table 1. Since $P(x_1 = 1, x_2 = 1) = \frac{5}{14}$ and $P(x_3 = -1) = \frac{7}{14}$, we see that $P(x_1 = 1, x_2 = 1, x_3 = -1) \neq P(x_1 = 1, x_2 = 1) P(x_3 = -1)$, this three-point probability does not factorise. This means that a Boltzmann machine requires hidden units to represent the data set (**a**). For the data set (**c**), by contrast, the three-point probabilities do factorise. For example, $P(x_1 = 1, x_2 = 1, x_3 = -1) = \frac{1}{8}$, $P(x_1 = 1, x_2 = 1) = \frac{1}{4}$, and $P(x_3 = -1) = \frac{1}{2}$. Since the three-point correlations can be expressed in terms of two-point correlations, no hidden units are needed to represent this data set with a Boltzmann machine.

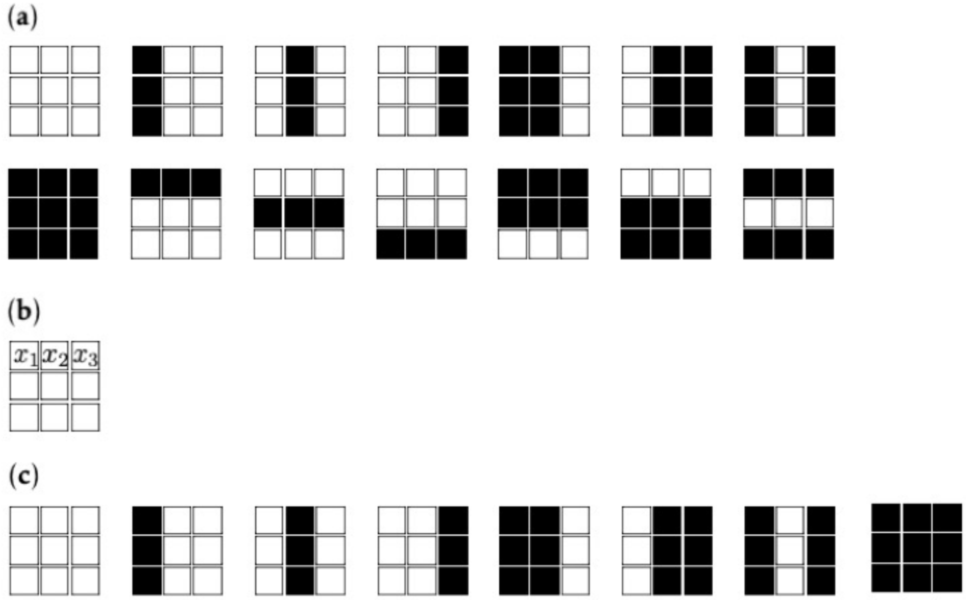

Figure 3: (**a**) 3×3 bars-and-stripes data set. The shown patterns occur with probability $P_{\text{data}} = \frac{1}{14}$, all other patterns have $P_{\text{data}} = 0$. (**b**) Definition of the bits x_1, x_2, and x_3. (**c**) Data set with stripes only. The shown patterns occur with probability $P_{\text{data}} = \frac{1}{8}$, all other patterns have $P_{\text{data}} = 0$. Question 6.

Errata for "Machine learning with neural networks" Bernhard Mehlig, Cambridge University Press (2021)

p. 32	l. 11	'$w_{ii} > 0$' should be replaced by '$w_{ii} = 0$'.
p. 32	l. 21	should read: '$H = -\frac{1}{2}\sum_{ij} w_{ij} g(b_i) g(b_j) - \int_0^{b_i} \mathrm{d}b b g'(b)$, with $b_i = \sum_j w_{ij} n_j - \theta_i$, cannot increase...'.
p. 37	l. 16	replace '$\sqrt{N}$' by '$N^{-1/2}$'.
	l. 17	replace '$\langle b_i(t)\rangle \sim N$' by '$\langle b_i(t)\rangle = O(1)$'.
p. 54	eq. (4.5c)	replace '$-\beta b_m$' by '$2\beta b_m$'.
p. 55	eq. (4.5d)	replace 'βb_m' by '$-2\beta b_m$'.
p. 67	alg. 3	add superscripts '(μ) ' to 'δw_{mn}, '$\delta\theta_n^{(\mathrm{v})}$', and '$\delta\theta_n^{(\mathrm{h})}$'.
p. 72	l. 12	the list should read '$1, 2, 4$, and 8'.
p. 85	fig. 5.11	switch the labels '10' and '50'.
p. 93	fig. 5.22	switch the labels '1111' and '1101' in the right panel.
p. 97	eq. (6.6a)	insert '$V_n^{(\mu)}$' before the '$\equiv$' sign.
p. 106	l. 18	should read 'a compromise, reducing the tendency of the network to overfit at the expense of training accuracy'.
p. 117	fig. 7.5	the hidden neurons should be labeled '$j = 0, 1, 2, 3$' from bottom to top.
p. 118	fig. 7.6	exchange labels '1' and '2'.
	eq. (7.9)	should read '$O_1 = \mathrm{sgn}(-V_0 + V_1 + V_2 - V_3)$'.
p. 121	fig. 7.10	change '$w^{(L-2)}$' to '$w^{(L)}$'.
p. 122	eq. (7.17)	replace '$\mathbb{J}$' by '$\mathbb{J}'$', also in the two lines above the equation.
p. 123	eq. (7.19)	should read '$\boldsymbol{\delta}^{(\ell)} = \boldsymbol{\delta}^{(L)} \mathbb{J}_{L-\ell}$ with $\mathbb{J}_{L-\ell} = [\mathbb{D}^{(L)}]^{-1} \mathbb{J}'_{L-\ell} \mathbb{D}^{(\ell)}$'.
p. 131	eq. (7.45)	replace 'O_l' by 'O_i'.
p. 139	l. 33	replace 'the Lagrangian (7.57)' by '$\frac{1}{2}\delta\boldsymbol{w} \cdot \mathbb{M}\delta\boldsymbol{w}$'.
p. 160	l. 15	delete 'then $L_{ij} = \delta_{ij}$. In this case'.
p. 161	l. 19	replace 'negative real parts' by 'positive real parts', and 'positive' by 'negative' in the next line.
p. 171	l. 23	the upper limit of the second summation should be 'M'.
p. 197	alg. 10	replace '$s_j = 0$' by '$s_j = 1$' in line 2 of Algorithm 10.
p. 202	l. 37	replace 'positive' by 'non-negative'.
p. 203	l. 21	should read 'Alternatively, assume that $\boldsymbol{w}^* = u + \mathrm{i}v$ can be written as an analytic function of $\boldsymbol{r} = r_1 + \mathrm{i}r_2$...'.
	l. 27	add 'See Ref. [2]'.
p. 225	l. 5,6	replace 'two' by 'two (three)' and 'lost' by 'lost (drew)'.

Gothenburg, October 18 (2022).