CHALMERS, GÖTEBORGS UNIVERSITET

EXAM for
ARTIFICIAL NEURAL NETWORKS

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

Time:	August 17 (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 exam.
CTH >13.5 passed; >17 grade 4; >21.5 grade 5,
GU >13.5 grade G; > 19.5 grade VG.

---

1. Boltzmann machine. A deterministic update rule for a restricted Boltzmann machine with N visible neurons v_j and M hidden ones h_i reads:

$$h'_m = \mathrm{sgn}(b_m^{(\mathrm{h})})\,, \quad \text{and} \quad v'_n = \mathrm{sgn}(b_n^{(\mathrm{v})})\,, \tag{1}$$

with $b_i^{(\mathrm{h})} = \sum_{j=1}^N w_{ij}v_j - \theta_i^{(\mathrm{h})}$ and $b_j^{(\mathrm{v})} = \sum_{i=1}^M h_i w_{ij} - \theta_j^{(\mathrm{v})}$. Show that the energy function of the restricted Boltzmann machine

$$H = -\sum_{i=1}^M \sum_{j=1}^N w_{ij}h_i v_j + \sum_{j=1}^N \theta_j^{(\mathrm{v})} v_j + \sum_{i=1}^M \theta_i^{(\mathrm{h})} h_i \tag{2}$$

cannot increase under this rule (**2**p).

Solution: The deterministic update rule follows from the deterministic limit of Eq. (4.30) in the course book [where $p(b) = (1 + \exp -2\beta b)^{-1}$ with noise level β^{-1}]:

$$h'_m = \mathrm{sgn}(b_m^{(\mathrm{h})})\,, \quad \text{and} \quad v'_n = \mathrm{sgn}(b_n^{(\mathrm{v})})\,, \tag{3}$$

with $b_i^{(\mathrm{h})} = \sum_{j=1}^N w_{ij}v_j - \theta_i^{(\mathrm{h})}$ and $b_j^{(\mathrm{v})} = \sum_{i=1}^M h_i w_{ij} - \theta_j^{(\mathrm{v})}$. Consider first the changes in H [Eq. (2)] when updating the hidden neurons, keeping the states of the visible neurons unchanged (constant). We write

$$H = -\sum_{i=1}^M h_i \Big(\sum_{j=1}^N w_{ij}v_j - \theta_i^{(\mathrm{h})} \Big) + \mathrm{const}\,. \tag{4}$$

This allows us to express the change in H as

$$H' - H = -\sum_{i=1}^{M}(h_i' - h_i)\Big(\sum_{j=1}^{N} w_{ij}v_j - \theta_i^{(\mathrm{h})}\Big)\,. \tag{5}$$

Suppose that $h_i = 1$ and $h_i' = -1$, so that $h_i' - h_i < 0$. It follows from Equation (3) that the sign of $\sum_{j=1}^{N} w_{ij}v_j - \theta_i^{(\mathrm{h})}$ equals $h_i' < 0$. Therefore $H' - H < 0$. Now assume that $h_i = -1$ and $h_i' = 1$. In this case $h_i' - h_i > 0$ and $\sum_{j=1}^{N} w_{ij}v_j - \theta_i^{(\mathrm{h})} > 0$. Again $H' - H < 0$. When $h_i' = h_i$, the energy function does not change.
In summary, H cannot increase when updating the hidden neurons (keeping the states of the visible neurons fixed). Here the argument works for synchronous updates of the hidden neurons, because there are no interactions between them. For the Hopfield model, the energy function can increase under synchronous updates. In a similar fashion one can show that H cannot increase under synchronous updates of the visible neurons, if one keeps the states of the hidden neurons constant.

2. Linear unit. The Boolean AND problem (Figure 1) cannot be solved by a linear unit (a neuron with activation function $g(b) = b$) with weights $\boldsymbol{w}$ and threshold θ. To show this, solve $\partial H/\partial \boldsymbol{w} = 0$ and $\partial H/\partial\theta = 0$ for $\boldsymbol{w}$ and θ, where $H = \frac{1}{2}\sum_\mu (t^{(\mu)} - O^{(\mu)})^2$. Using these weights and thresholds, demonstrate that $O^{(\mu)} \neq t^{(\mu)}$. *Hint:* express the linear system to solve in terms of $\langle \boldsymbol{x}\boldsymbol{x}^{\mathsf{T}}\rangle$, $\langle \boldsymbol{x}\rangle$, $\langle t\boldsymbol{x}\rangle$, and $\langle t\rangle$, where $\langle\cdots\rangle$ is an average over patterns (**1.5**p).

Qualitatively sketch the contours of H as a function of the weight components w_1 and w_2 for a fixed value of θ (take the value you obtained above) (**0.5**p).

Solution: The energy function for a linear unit with threshold θ can be written as $H = \frac{1}{2}\sum_\mu (t^{(\mu)} - \boldsymbol{w}\cdot\boldsymbol{x}^{(\mu)} + \theta)^2$. The derivatives of H with respect

x_1	x_2	t
0	0	-1
0	1	-1
1	0	-1
1	1	+1

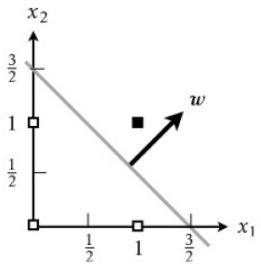

Figure 1: Boolean AND problem. Left: value table. Right: input plane, decision boundary, and weight vector $\boldsymbol{w}$. The pattern $\boldsymbol{x} = [1, 1]$ with target $t = 1$ is marked as ■, the patterns with $t = -1$ as □. Question 2.

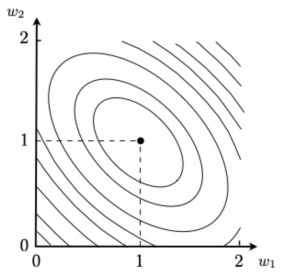

Figure 2: Energy function $H(w_1, w_2, \theta = \frac{3}{2})$ for the Boolean AND problem. Question 2.

to $\boldsymbol{w}$ and θ are

$$\frac{\partial H}{\partial \boldsymbol{w}} = p\big(\langle t\boldsymbol{x}\rangle - \langle \boldsymbol{x}\boldsymbol{x}^{\mathsf{T}}\rangle \boldsymbol{w} + \theta\langle \boldsymbol{x}\rangle\big)\,, \quad \frac{\partial H}{\partial \theta} = p\big(\langle t\rangle - \boldsymbol{w}^{\mathsf{T}}\langle \boldsymbol{x}\rangle + \theta\big)\,. \tag{6}$$

Here $\langle \cdots \rangle = p^{-1}\sum_{\mu=1}^{p} \cdots$ is the average over patterns. For the AND problem (Figure 5.7 in the course book),

$$\langle t\boldsymbol{x}\rangle = \begin{bmatrix} 0 \\ 0 \end{bmatrix}\,, \quad \langle \boldsymbol{x}\boldsymbol{x}^{\mathsf{T}}\rangle = \tfrac{1}{4}\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}\,, \quad \langle \boldsymbol{x}\rangle = \tfrac{1}{2}\begin{bmatrix} 1 \\ 1 \end{bmatrix}\,, \quad \text{and} \quad \langle t\rangle = -\tfrac{1}{2}\,. \tag{7}$$

Now set the derivatives to zero to determine $\boldsymbol{w}$ and θ. This gives $\boldsymbol{w} = [1,1]^{\mathsf{T}}$ and $\theta = \frac{3}{2}$. With these weights and threshold, we find $O^{(1)} = \boldsymbol{w}\cdot\boldsymbol{x}^{(1)} - \theta = -\frac{3}{2}$, $O^{(2)} = -\frac{1}{2}$, $O^{(3)} = -\frac{1}{2}$, and $O^{(4)} = \frac{1}{2}$. So $O^{(mu)} \neq t^{(mu)}$. In other words: H is non-zero.
However, one can check that the values obtained for $\boldsymbol{w}$ and θ correspond to a local minimum of H at $\boldsymbol{w} = [1,1]^{\mathsf{T}}$ and $\theta = \frac{3}{2}$, where $H = \frac{1}{2}$. Since H is non-zero at the minimum, the optimal solution is only approximate. Figure 2 shows how the energy function depends on w_1 and w_2 for $\theta = \frac{3}{2}$.

3. Backpropagation. Consider the residual network shown in Figure 3, where $V^{(0)} = x$ is the input and $V^{(4)} = O$ is the output. Write down the dynamical rules for all neurons $V^{(\ell)}$ for $\ell = 1, \ldots 4$. Derive the learning rules for their weights and thresholds, using gradient descent for the energy function $H = \frac{1}{2}(t - O)^2$. (**2**p).

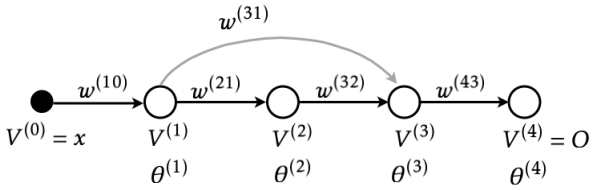

Figure 3: Residual network, chain of neurons with a skipping connection. Question 3.

Solution: Eq. (7.34) in the course book yields for the weight updates:

$$\delta w^{(43)} = \eta(t-O)g'(b^{(4)})V^{(3)} \tag{8a}$$
$$\delta w^{(32)} = \eta(t-O)g'(b^{(4)})w^{(43)}g'(b^{(3)})V^{(2)} \tag{8b}$$
$$\delta w^{(31)} = \eta(t-O)g'(b^{(4)})[w^{(43)}g'(b^{(3)}V^{(1)} \tag{8c}$$
$$\delta w^{(21)} = \eta(t-O)g'(b^{(4)})w^{(43)}g'(b^{(3)})w^{(32)}g'(b^{(2)})V^{(1)} \tag{8d}$$
$$\delta w^{(10)} = \eta(t-O)g'(b^{(4)})w^{(43)}g'(b^{(3)})[w^{(31)}+w^{(32)}g'(b^{(2)})w^{(21)}]g'(b^{(1)})V^{(0)} \tag{8e}$$

The corresponding formulae for the threshold updates are obtained by replacing $V^{(j)}$ by -1.

4. Generalised XOR function. The parity function can be viewed as a generalisation of the XOR function to $N > 2$ input dimensions, because it becomes the XOR function for $N = 2$. Another way to generalise the XOR function to $N > 2$-dimensional inputs is to define a Boolean function that gives unity if exactly one of its inputs equals unity. Otherwise the function evaluates to zero. For $N = 3$, construct a network that represents this function. Then construct a network that does the trick for $N = 4$. In both cases, explain (on max one A4 page for both) why/how your networks work (**2**p).

Solution: The input-space representation of the three-dimensional generalised XOR function is shown in Figure 4(left). The problem is not linearly separable. A network that solves this classification problem is shown on the right of Figure 4(right). The generalised XOR function with four inputs is represented in an analogous way.
An alternative is to use winning neurons. The construction outlined in Section 7.1 in the course book requires 2^N hidden neurons, so eight hidden neurons for $N = 3$, whereas the network layout shown in Figure 4 has ten hidden neurons. For $N = 4$, this approach requires 12 hidden neurons, while the construction with winning neurons requires 16 hidden neurons.

5. Linearly inseparable problem. A classification problem is given in Figure 5. Inputs $\boldsymbol{x}^{(\mu)}$ inside the gray 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 $V_j^{(\mu)} = \text{sgn}\left(-\theta_j + \sum_{k=1}^{2} w_{jk}x_k^{(\mu)}\right)$, for $j = 1, \ldots, 4$. The output is computed as $O^{(\mu)} = \text{sgn}\left(-\Theta + \sum_{j=1}^{4} W_j V_j^{(\mu)}\right)$. Find the weights w_{jk}, W_j, and thresholds θ_j, Θ that solve the classification problem, assuming that all hidden weight vectors point out of the gray region (**2**p).

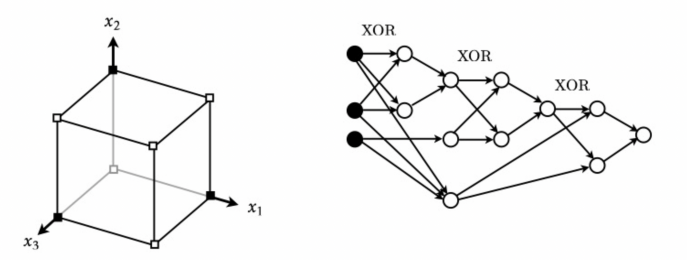

Figure 4: Input space and network layout for the three-dimensional exclusive XOR problem. Question 4.

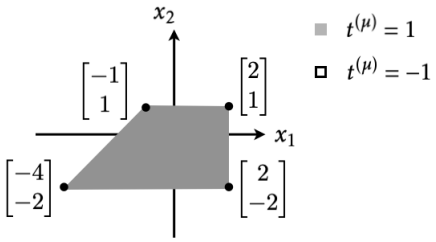

Figure 5: Classification problem for question 5.

Solution: We first compute the outward-pointing normal vectors to each decision boundary. Denoting the point $[-4, -2]^\mathsf{T}$ as A, $([1, 1]^\mathsf{T}$ as B, $[2, 1]^\mathsf{T}$ as C, and $[2, -2]^\mathsf{T}$ as D, the normal vectors of the decision boundaries are

$$\mathbf{n}_{AB} = [-1, 1]^\mathsf{T}, \quad \mathbf{n}_{BC} = [0, 1]^\mathsf{T}, \quad \mathbf{n}_{CD} = [1, 0]^\mathsf{T}, \quad \mathbf{n}_{DA} = [0, -1]^\mathsf{T}, \tag{9}$$

This gives the following weight matrix for the hidden neurons:

$$\begin{bmatrix} -1 & 1 \\ 0 & 1 \\ 1 & 0 \\ 0, & -1 \end{bmatrix}, \tag{10}$$

To obtain the threshold values θ_i, we solve the equations $w_{i1}x_1 + w_{i2}x_2 - \theta_i = 0$, where x_1 and x_2 are coordinates on the decision boundary corresponding to index i. This yields the thresholds

$$\theta = \begin{bmatrix} 2 \\ 1 \\ 2 \\ 2 \end{bmatrix}. \tag{11}$$

Now, by setting the weights leading from the hidden layer to the output to unity,

$$\boldsymbol{W} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \tag{12}$$

the output equals -4 if the coordinate lies inside all decision boundaries. Otherwise the output is larger or equal to -2. Hence we set the output threshold Θ to a value between -4 and -2.

6. Non-linear transformation of a classification problem. Table 6 describes a classification problem. Show that this problem can be solved as follows. Transform the inputs $\boldsymbol{x}$ to two-dimensional coordinates u_1, u_2 using the functions:

$$u_1 = \exp(-\tfrac{1}{4}|\boldsymbol{x} - \boldsymbol{w}_1|^2), \text{ with } \boldsymbol{w}_1 = [\text{-1,1,1}]^\mathsf{T} , \tag{13}$$

$$u_2 = \exp(-\tfrac{1}{4}|\boldsymbol{x} - \boldsymbol{w}_2|^2), \text{ with } \boldsymbol{w}_2 = [\text{1,1,-1}]^\mathsf{T} . \tag{14}$$

Plot the positions of the eight input patterns in the u_1-u_2-plane. *Hint*: to compute u_j use the following approximations: exp(-1) $\approx$ 0.37, exp(-2) $\approx$ 0.14, exp(-3) $\approx$ 0.05. The transformed data is used as input to a simple perceptron $O^{(\mu)} = \text{sgn}\big(\sum_{i=1}^{2} W_j u_j{}^{(\mu)} - \Theta\big)$. Draw a decision boundary in the u_1-u_2-plane and determine the corresponding weight vector $\boldsymbol{W}$, as well as the threshold Θ. (**2**p).

Solution: Figure 7 shows that the problem given in Figure 5 is not linearly separable. Mapping input space as described in the problem, results in the problem shown on the right of Figure 7. In the new coordinates, the problem is linearly separable with $\boldsymbol{W} = [-1, -1]^\mathsf{T}$ and $\Theta = -1$.

x_1	x_2	x_3	t
-1	-1	-1	1
-1	-1	1	1
-1	1	-1	1
-1	1	1	-1
1	-1	-1	1
1	-1	1	1
1	1	-1	-1
1	1	1	1

Figure 6: Classification problem for question 6.

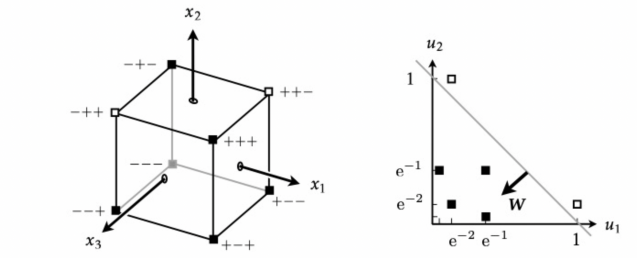

Figure 7: Left: input plane for question 6. Right: mapped problem in the u_1-u_2-plane, weight vector $\boldsymbol{W}$,and corresponding decision boundary for question 6.

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 $\boldsymbol{r} = r_1 + \mathrm{i}r_2 \ldots$'.
	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).