CHALMERS, GÖTEBORGS UNIVERSITET

EXAM for
ARTIFICIAL NEURAL NETWORKS

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

Time:	October 28, 2022, at $14^{00} - 18^{00}$
Place:	Johanneberg
Teachers:	Bernhard Mehlig, 073-420 0988 (mobile)
	Teacher visits at 14^{30} and 17^{30}
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. Recognising patterns with a Hopfield network. Five patterns are shown in Figure 1 with $N = 48$. Store the patterns $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ in a Hopfield network using Hebb's rule $w_{ij} = \frac{1}{N}\sum_{\mu=1}^{2} x_i^{(\mu)} x_j^{(\mu)}$ with $i, j = 1, \ldots, N$. Use the update rule

$$S_i \leftarrow \mathrm{sgn}\left(\sum_{j=1}^{N} w_{ij} S_j\right). \tag{1}$$

Follow the steps outlined below to determine which of the input patterns are attractors for synchronous updates.

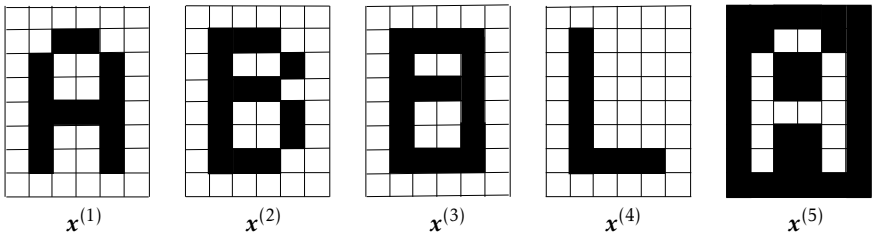

Figure 1: Patterns for question 1. The bits are encoded as follows: $x_i^{(\mu)} = -1$ (□), $x_i^{(\mu)} = 1$ (■).

(a) Compute $\sum_{j=1}^{N} x_j^{(\mu)} x_j^{(\nu)}$ for $\mu = 1$, $\nu = 1, \ldots, 5$ and for $\mu = 2$, $\nu = 1, \ldots, 5$. *Hint:* The result can be read off from the Hamming distances between the patterns shown in Figure 1. (**0.5**p).
(b) Consider the quantity $b_i^{(\nu)} = \sum_{j=1}^{N} w_{ij} x_j^{(\nu)}$, where w_{ij} are the weights obtained by storing patterns $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$. Compute $b_i^{(\nu)}$ for $\nu = 1, \ldots, 5$. Express your result as linear combinations of $x_i^{(1)}$ and $x_i^{(2)}$. *Hint:* use your answer from the first part of this question. (**1**p).
(c) Feed all patterns in Figure 1 to the network. Which of the patterns remain the same after one synchronous update according to Eq. (1)? (**0.5**p).

Solution: Define the quantity

$$Q_{\mu\nu} = \frac{1}{N} \sum_{j=1}^{N} x_j^{(\mu)} x_j^{(\nu)}. \tag{2}$$

The Hamming distance $d_{\mu\nu}$, which for $-1/+1$ bits can be read off as the number of bits that are different between patterns μ and ν, is related to $Q_{\mu\nu}$ as

$$Q_{\mu\nu} = \frac{N - 2d_{\mu\nu}}{N}. \tag{3}$$

Below is shown a table with the $NQ_{\mu\nu}$ and $d_{\mu\nu}$ for the different patterns

μ	ν	$d_{\mu\nu}$	$NQ_{\mu\nu}$
1	1	0	48
1	2	9	30
1	3	8	32
1	4	11	26
1	5	48	-48
2	2	0	48
2	3	3	42
2	4	8	32
2	5	39	-30

(a) Read off result from $NQ_{\mu\nu}$ column.
(b) Using Hebb's rule, the local field $b_i^{(\nu)} = \sum_{\mu=1}^{2} \frac{1}{N} \sum_{j=1}^{N} x_i^{(\mu)} x_j^{(\mu)} x_j^{(\nu)} = Q_{1\nu} x_i^{(1)} + Q_{2\nu} x_i^{(2)}$.
(c) Pattern 1 remains the same because $|Q_{11}| > |Q_{21}| = |Q_{12}|$. Pattern 2 remains the same because $|Q_{22}| > |Q_{12}|$. Pattern 3 changes and becomes pattern 2 after one synchronous update because $|Q_{23}| > |Q_{13}|$. Pattern 4 changes and becomes pattern pattern 2 after one synchronous update because $|Q_{24}| > |Q_{14}|$. Pattern 5 remains the same because $|Q_{15}| > |Q_{25}|$ and Q_{15} is negative.

2. Deterministic dynamics for restricted Boltzmann machine. Consider the energy function for the restricted Boltzmann machine

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

where M is the number of hidden neurons h_i, and N is the number of visible neurons v_i.
(a) Show that the energy cannot increase during asynchronous updates under the deterministic McCulloch-Pitts dynamics

$$h'_m = \mathrm{sgn}\left(b_m^{(\mathrm{h})}\right), \quad \text{and} \quad v'_n = \mathrm{sgn}\left(b_n^{(\mathrm{v})}\right) \tag{5}$$

with $b_i^{(\mathrm{h})} \sum_{j=1}^{N} w_{ij} v_j - \theta_i^{(\mathrm{h})}$ and $b_j^{(\mathrm{v})} = \sum_{i=1}^{M} h_i w_{ij} - \theta_j^{(\mathrm{v})}$. Note that it is not required that the weight matrix is symmetric, or that the diagonal elements are non-positive. (**1**p).
(b) Show that the energy function cannot increase during synchronous updates. The McCulloch-Pitts dynamics are also used in the Hopfield model. Why can the energy function increase during synchronous updates for the Hopfield model, but not for the restricted Boltzmann machine (no derivation required) (**1**p).
Solution: Consider first the changes in H when updating the hidden neurons, keeping the states of the visible neurons unchanged (constant). We write

$$H = -\sum_{i=1}^{M} h_i \Big(\sum_{j=1}^{N} w_{ij} v_j - \theta_i^{(\mathrm{h})} \Big) + \text{const}\,. \tag{6}$$

This allows us to express the change in H as

$$H' - H = -\sum_{i=1}^{M} (h'_i - h_i) \Big(\sum_{j=1}^{N} w_{ij} v_j - \theta_i^{(\mathrm{h})} \Big)\,. \tag{7}$$

Suppose that $h_i = 1$ and $h'_i = -1$, so that $h'_i - h_i < 0$. It follows from Equation (5) 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 (Exercise 2.9). In a similar fashion one shows that H cannot increase under synchronous updates of the visible neurons, if one keeps the states of the hidden neurons constant.

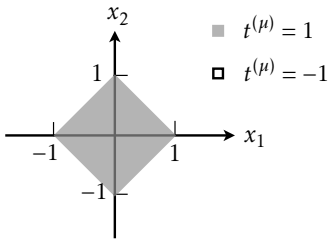

Figure 2: Classification problem for question 3.

3. Linearly inseparable problem. Figure 2 shows a classification problem where the input coordinates $\boldsymbol{x}^{(\mu)}$ inside the diamond have the target $t^{(\mu)} = 1$ and input coordinates outside of the diamond have target value $t^{(\mu)} = -1$. Design a fully-connected neural network with two neurons, a hidden layer with M neurons, and an output layer with one neuron that solves the classification problem. Use $g(b) = \text{sgn}(b)$ for all neurons. Denote the weights leading from the input to the hidden layer by w_{ij}, the thresholds of the hidden layer by θ_i, the weights leading from the hidden layer to the output layer by W_i and the threshold of the output neuron by Θ. Clearly write down all the parameter values chosen for the network. (**2**p).

Solution: Choose $M = 4$ to have one decision boundary for each boundary of the diamond. 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 pointing towards the origin:

$$\mathbb{W}^{(1)} = \begin{bmatrix} -1 & -1 \\ 1 & -1 \\ 1 & 1 \\ -1 & 1 \end{bmatrix}. \tag{8}$$

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 $\theta_1^{(1)} = \theta_2^{(1)} = \theta_3^{(1)} = \theta_4^{(1)} = -1$. 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 diamond. 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.

4. Convolutional network. Consider the images in Figures 3, where white bits have take the value 0, and black bits take the value 1.

(a) Design a convolutional neural network with a convolution layer with a 3×3 local receptive field using a stride $(1, 1)$ and padding $(0, 0, 0, 0)$, a max-pooling layer with a 3×3 local receptive field using a stride $(1, 1)$ and padding $(0, 0, 0, 0)$, and a single output neuron, each layer using the ReLU activation

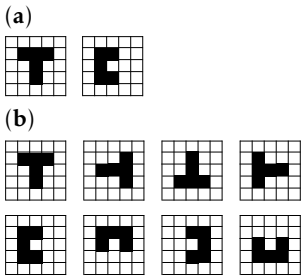

Figure 3: Patterns for question 4. The bits are encoded as follows: $x_i^{(\mu)} = 0$ (□), $x_i^{(\mu)} = 1$ (■).

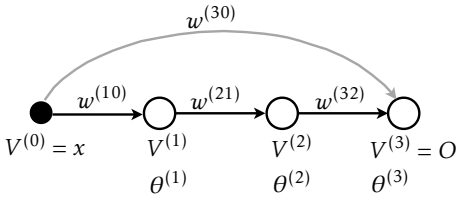

Figure 4: Network layout for question 5.

function, that can distinguish 'T' from 'C' in panel(a). (**1**p).
(b) Does your network classify all patterns in panel (**b**) correctly? If not, modify your network so that it does. (**1**p).
Solution: Set the kernel to be a plus sign made of 1s and the rest 0s (see Figure 5). Every T-pattern contains a plus sign with one side removed, meaning that convolving the kernel over such an image will always result in a convolution layer with a component equal to 4. In the C-patterns, the greatest overlap with the plus-shaped kernel will be a plus sign with two sides removed (or the middle and one side removed). Hence, the maximal value of the convolution layer when a C-pattern is fed will be 3. Setting the weight connecting the max-pooling layer to the output neuron to 1, we can thus use a threshold of 3.5 in the output layer to classify T-patterns from C-patterns.

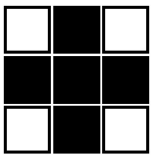

Figure 5: Kernel used for convolutional neural network.

5. Residual network. Consider the residual network shown in Figure 4, where $V^{(0)} = x$ is the input and $V^{(3)} = O$ is the output.
(a) Write down the dynamical rules for all hidden neurons and the output neuron. (**0.5**p).
(b) Derive the learning rules for the weights and thresholds of the network. (**1.5**p).
Solution: (a) The dynamical rules are

$$V^{(1)} = g(b^{(1)}) = g(w^{(10)}x - \theta^{(1)}), \tag{9}$$
$$V^{(2)} = g(b^{(2)}) = g(w^{(21)}V^{(1)} - \theta^{(2)}), \tag{10}$$
$$O = g(b^{(3)}) = g(w^{(32)}V^{(2)} + w^{(30)}x - \theta^{(3)}) \tag{11}$$

(b) Using the quadratic energy function $H = \frac{1}{2}(y-O)^2$, where we have used a batch size of 1 and omitted the dependency on input pattern μ, the update rules for the weights are

$$\delta w^{(32)} = -\eta \frac{\partial H}{\partial w^{(32)}} = \eta(y-O)g'(b^{(3)})V^{(2)}, \tag{12}$$
$$\delta w^{(21)} = -\eta \frac{\partial H}{\partial w^{(21)}} = \eta(y-O)g'(b^{(3)})w^{(32)}g'(b^{(2)})V^{(1)}, \tag{13}$$
$$\delta w^{(10)} = -\eta \frac{\partial H}{\partial w^{(10)}} = \eta(y-O)g'(b^{(3)})w^{(32)}g'(b^{(2)})w^{(21)}g'(b^{(1)})x, \tag{14}$$
$$\delta w^{(30)} = -\eta \frac{\partial H}{\partial w^{(30)}} = \eta(y-O)g'(b^{(3)})x. \tag{15}$$

The update rules for the thresholds are

$$\delta\theta^{(3)} = -\eta \frac{\partial H}{\partial \theta^{(3)}} = -\eta(y-O)g'(b^{(3)}), \tag{16}$$
$$\delta\theta^{(2)} = -\eta \frac{\partial H}{\partial \theta^{(2)}} = -\eta(y-O)g'(b^{(3)})w^{(32)}g'(b^{(2)}), \tag{17}$$
$$\delta\theta^{(1)} = -\eta \frac{\partial H}{\partial \theta^{(1)}} = -\eta(y-O)g'(b^{(3)})w^{(32)}g'(b^{(2)})w^{(21)}g'(b^{(1)}). \tag{18}$$

6. Ridge regression. Consider a reservoir computer with N input neurons, N_r reservoir neurons, and M output neurons. The activation function of the output layer is chosen to be the identity function $g(b) = b$. Training the network amounts to adjusting the output weights $\mathbb{W}^{(\text{out})}$ to minimise some energy function. Show that the output weights obtained by minimizing the quadratic energy function (here written in vector notation) with L_2-regularization

$$H = \frac{1}{2}\sum_{t}^{T}\Big(\boldsymbol{y}(t) - \boldsymbol{O}(t)\Big)^{\top}\Big(\boldsymbol{y}(t) - \boldsymbol{O}(t)\Big) + \frac{\gamma}{2}\text{tr}\left(\left[\mathbb{W}^{(\text{out})}\right]^{\top}\mathbb{W}^{(\text{out})}\right), \quad (19)$$

where $\text{tr}(\mathbb{A}) = \sum_i A_{ii}$ is the trace of some matrix $\mathbb{A}$, $\boldsymbol{y}(t)$ is an $M \times 1$ column vector representing the targets at time t, $\boldsymbol{O}(t) = \mathbb{W}^{(\text{out})}\boldsymbol{r}(t)$ is an $M \times 1$ column vector containing the outputs at time t, $\mathbb{W}^{(\text{out})}$ is an $M \times N_r$ weight matrix, and $\boldsymbol{r}(t)$ is an $N_r \times 1$ column vector containing the reservoir states at time t, is equivalent to the ridge regression estimation of the output weights with ridge parameter γ,

$$\mathbb{W}^{(\text{out})} = \left(\sum_{t=1}^{T}\boldsymbol{y}(t)\boldsymbol{r}^{\top}(t)\right)\left(\sum_{t=1}^{T}\boldsymbol{r}(t)\boldsymbol{r}^{\top}(t) + \gamma\mathbb{I}\right)^{-1} = \mathbb{Y}\mathbb{R}^{\top}\left(\mathbb{R}\mathbb{R}^{\top} + \gamma\mathbb{I}\right)^{-1}. \quad (20)$$

Here, $\mathbb{Y}$ is an $M \times T$ matrix with columns $\boldsymbol{y}(t)$ and $\mathbb{R}$ is an $N_r \times T$ matrix with columns $\boldsymbol{r}(t)$. You may show this using either vector or index notation. *Hint:* You may or may not use index notation to find the answer. If you prefer index notation, you can use that

$$\Big(\boldsymbol{y}(t) - \boldsymbol{O}(t)\Big)^{\top}\Big(\boldsymbol{y}(t) - \boldsymbol{O}(t)\Big) = \sum_i \left[y_i(t) - O_i(t)\right]^2 \quad (21)$$

and

$$\text{tr}\left(\left[\mathbb{W}^{(\text{out})}\right]^{\top}\mathbb{W}^{(\text{out})}\right) = \sum_{ij}\left(w_{ij}^{(\text{out})}\right)^2. \quad (22)$$

(**2**p).

Solution: Expanding Eq. 19 and taking the gradient with respect to $\mathbb{W}^{(\text{out})}$, one finds:

$$\nabla_{\mathbb{W}^{(\text{out})}}\left\{\frac{1}{2}\sum_{t=1}^{T}\left[\boldsymbol{y}^{\top}\boldsymbol{y} - 2\boldsymbol{y}^{\top}\mathbb{W}^{(\text{out})}\boldsymbol{r} + \left(\mathbb{W}^{(\text{out})}\boldsymbol{r}\right)^{\top}\mathbb{W}^{(\text{out})}\boldsymbol{r}\right] + \frac{\gamma}{2}\text{Tr}\left[\mathbb{W}^{(\text{out})\top}\mathbb{W}^{(\text{out})}\right]\right\}$$
$$= \sum_{t=1}^{T}\left(-\boldsymbol{y}\boldsymbol{r}^{\top} + \mathbb{W}^{(\text{out})}\boldsymbol{r}\boldsymbol{r}^{\top}\right) + \gamma\mathbb{W}^{(\text{out})}. \quad (23)$$

Here we omitted the dependence on t of $\boldsymbol{y}$ and $\boldsymbol{r}$ for brevity. At the global minimum the gradient must vanish. We set the gradient to zero and solve

for $\mathbb{W}^{(\text{out})}$. This gives

$$\mathbb{W}^{(\text{out})} = \left(\sum_{t=1}^{T} \boldsymbol{y}\boldsymbol{r}^{\top} \right) \left(\sum_{t=1}^{T} \boldsymbol{r}\boldsymbol{r}^{\top} + \gamma \mathbb{I} \right)^{-1} \tag{24}$$

which is equivalent to

$$\mathbb{W}^{(\text{out})} = \mathbb{Y}\mathbb{R}^{\top} \left(\mathbb{R}\mathbb{R}^{\top} + \gamma \mathbb{I} \right)^{-1} \tag{25}$$

where $\mathbb{Y}$ is a $M \times T$ matrix with columns $\boldsymbol{y}(t)$ and $\mathbb{R}$ is an $N_{\text{r}} \times T$ matrix with columns $\boldsymbol{r}(t)$, where N_{r} is the number of reservoir neurons. Equation (25) is the expression for ridge regression[1]. In summary, ridge regression is equivalent to minimising the quadratic energy function $H = \frac{1}{2} \sum_{t=1}^{T} \sum_{i=1}^{M} [E_i(t)]^2$ with linear activation functions with respect to the output weights $w_{mn}^{(\text{out})}$ using L_2-regularisation.

[1]See Section 1.5 of [W. N. van Wieringen, *Lecture notes on ridge regression*, arxiv:1509.09169].

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).