**CHALMERS, GÖTEBORGS UNIVERSITET**

EXAM for
ARTIFICIAL NEURAL NETWORKS

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

| | |
|---|---|
| **Time:** | October 23, 2017, at $8^{30} - 12^{30}$ |
| **Place:** | Lindholmen-salar |
| **Teachers:** | Bernhard Mehlig, 073-420 0988 (mobile) |
| | Marina Rafajlović, 076-580 4288 (mobile), visits once at $9^{30}$ |
| **Allowed material:** | Mathematics Handbook for Science and Engineering |
| **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** $\geq$14 passed; $\geq$17.5 grade 4; $\geq$22 grade 5,
**GU** $\geq$14 grade G; $\geq$ 20 grade VG.

---

**1. Recognition of one pattern.** In the deterministic Hopfield model, the state $S_i$ of the $i$-th neuron is updated according to the Mc-Culloch Pitts rule

$$S_i \leftarrow \mathrm{sgn}(b_i), \text{ where } b_i = \sum_{j=1}^{N} w_{ij} S_j \,. \tag{1}$$

Here $N$ is the number of neurons, $w_{ij}$ are the weights. The weights depend on $p$ patterns $\boldsymbol{\zeta}^{(\mu)} = (\zeta_1^{(\mu)}, \ldots, \zeta_N^{(\mu)})^{\mathsf{T}}$ stored in the network according to Hebb's rule:

$$w_{ij} = \frac{1}{N} \sum_{\mu=1}^{p} \zeta_i^{(\mu)} \zeta_j^{(\mu)} \quad \text{for } i, j = 1, \ldots, N \,. \tag{2}$$

Here $\zeta_i^{(\mu)}$ takes values 1 or $-1$.

a) Store the two patterns shown in Figs. 1**A** and **B**. Compute the weights $w_{ij}$. (**0.5**p)

b) Apply the pattern shown in Fig. 1**A** to the network. Compute the network output after a single step of synchronous updating according to Eq. (1). Discuss your result. (**0.5**p)

c) Apply the pattern shown in Fig. 1**B** instead, and compute the network

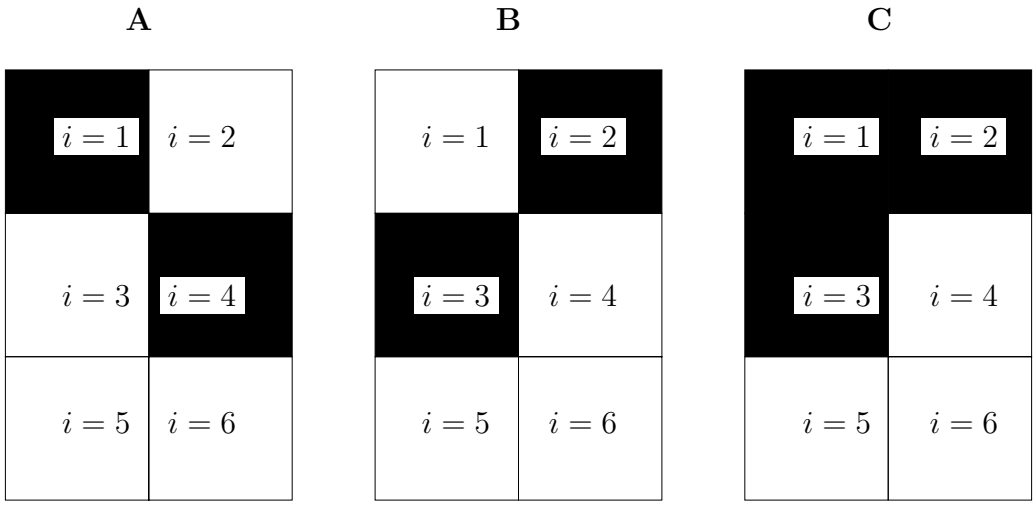


Figure 1: Question 1. Panel **A** shows pattern $\boldsymbol{\zeta}^{(1)}$. The pattern has $N = 6$ bits. Black corresponds to $\zeta_i^{(1)} = 1$, and white to $\zeta_i^{(1)} = -1$. Panel **B** shows pattern $\boldsymbol{\zeta}^{(2)}$, and panel **C** shows a perturbed version of input patterns.

output after a single step of synchronous updating according to Eq. (1). Discuss your result. (**0.5**p)

d) Now apply the pattern shown in Fig. 1**C** instead. Compute the network output after a single step of synchronous updating according to Eq. (1). Discuss your result. (**0.5**p)

**2. Gradient-descent algorithm in supervised learning.** To train a simple perceptron using a gradient-descent algorithm one needs update formulae for the weights and thresholds in the network.

a) Derive these update formulae for the network shown in Fig. 2 using a stochastic path through the weight space, and assuming constant and small learning rate $\eta$, no momentum, and no weight decay. The weights are denoted by $w_i$ where $i = 1, 2, 3$. The threshold corresponding to the output unit is denoted by $\theta$, and the activation function by $g(\cdots)$. The target value for input pattern $\boldsymbol{\xi}^{(\mu)} = (\xi_1^{(\mu)}, \xi_2^{(\mu)}, \xi_3^{(\mu)})^{\mathsf{T}}$ is $\zeta^{(\mu)}$, and the network output is $O^{(\mu)} = g(\sum_{i=1}^{3} w_i \xi_i^{(\mu)} - \theta)$. The energy function is $H = \frac{1}{2} \sum_{\mu=1}^{p} \Big(\zeta^{(\mu)} - O^{(\mu)}\Big)^2$, where $p$ denotes the total number of input patterns. (**1**p)

b) Now repeat the task **2a)** but implementing weight decay according to the gradient descent of the following energy function:

$$H = \frac{1}{2} \sum_{\mu=1}^{p} \Big(\zeta^{(\mu)} - O^{(\mu)}\Big)^2 + \frac{\gamma}{2} \sum_{i=1}^{3} w_i^2 \ . \tag{3}$$

Here $\gamma$ is a positive constant. Discuss differences to the update rule you derived in **2a)**. What is an advantage of this update rule over the update

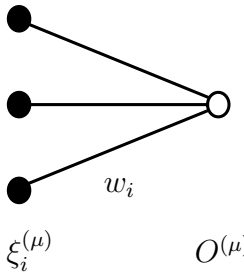

Figure 2: Question 2. Simple perceptron with three input units and one output unit.

rule in **2a)**? (**0.5**p)

c) Now repeat the task **2b)** but using the following energy function instead:

$$H = \frac{1}{2}\sum_{\mu=1}^{p}\Big(\zeta^{(\mu)} - O^{(\mu)}\Big)^2 + \frac{\gamma}{2}\sum_{i=1}^{3}\frac{w_i^2}{1+w_i^2}\ . \tag{4}$$

Here $\gamma$ is a positive constant. Discuss differences to the update rules you derived in **2a)** and **2b)**. What is an advantage of this update rule over the update rule in **2b)**? (**0.5**p)

**3. Multilayer perceptron.** Consider the problem in Table 1. Here, $\mu = 1, 2, \ldots, 11$ is the index of input pattern $\boldsymbol{\xi}^{(\mu)} = (\xi_1^{(\mu)}, \xi_2^{(\mu)})^\mathsf{T}$, and $\zeta^{(\mu)}$ is the corresponding target output of this input pattern. The problem is illustrated geometrically in Fig. 3 where patterns with $\zeta^{(\mu)} = 0$ are denoted by white circles, and patterns with $\zeta^{(\mu)} = 1$ are denoted by black circles.

a) Can this problem be solved by a simple perceptron with 2 input units, and a single output unit. Why? (**0.5**p)

b) This problem can be solved by a multilayer perceptron with 2 input units $\xi_i^{(\mu)}$ $(i = 1, 2)$, 3 hidden units

$$V_k^{(\mu)} = \theta\Big(\sum_{i=1}^{2} w_{ki}\xi_i^{(\mu)} - \theta_k\Big),\ \text{where } k = 1, 2, 3\,, \tag{5}$$

and one output unit

$$O^{(\mu)} = \theta\Big(\sum_{k=1}^{3} W_k V_k^{(\mu)} - \Theta\Big)\ . \tag{6}$$

Here

$$\theta(x) = \begin{cases} 0, \text{ if } x < 0\ , \\ 1, \text{ if } x > 0 \end{cases} \tag{7}$$

is the Heaviside step function, and $w_{ki}$, and $W_k$ are weights for the hidden, and the output layer, respectively. Finally, $\theta_k$ and $\Theta$ are the thresholds

assigned to the hidden units, and to the output unit, respectively. One possibility to solve the problem is illustrated in Fig. 3 where the three lines (solid, dashed, and dash-dotted line) are determined by weights $w_{ki}$ and thresholds $\theta_k$ assigned to the three hidden units in the hidden layer. Compute $w_{ki}$ and $\theta_k$ corresponding to the lines shown in Fig. 3. Note that the point where the dashed and dash-dotted lines intersect has the following coordinates $(0.5, 0.8)^\mathsf{T}$. (**0.5**p)

c) For each pattern $\boldsymbol{\xi}^{(\mu)}$ write its coordinates $V_k^{(\mu)}$ in the transformed (hidden) space. (**0.5**p)

d) Now illustrate graphically the problem in this transformed space. Is the problem represented in this transformed space linearly separable? If yes, illustrate a possible solution to the problem in this space. (**0.5**p)

e) Compute the corresponding weights $W_k$ and the threshold $\Theta$ corresponding to the solution you illustrated graphically in **3d)**. (**0.5**p)

| $\mu$ | $\xi_1^{(\mu)}$ | $\xi_2^{(\mu)}$ | $\zeta^{(\mu)}$ |
|---|---|---|---|
| 1 | 0.1 | 0.95 | 0 |
| 2 | 0.2 | 0.85 | 0 |
| 3 | 0.2 | 0.9 | 0 |
| 4 | 0.3 | 0.75 | 1 |
| 5 | 0.4 | 0.65 | 1 |
| 6 | 0.4 | 0.75 | 1 |
| 7 | 0.6 | 0.45 | 0 |
| 8 | 0.8 | 0.25 | 0 |
| 9 | 0.1 | 0.65 | 1 |
| 10 | 0.2 | 0.75 | 1 |
| 11 | 0.7 | 0.2 | 1 |

Table 1: Question 3. Inputs and target values for the problem in question 3.

**4. Oja's learning rule.** The aim of unsupervised learning is to construct a network that learns the properties of a distribution of input patterns $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_N)^\mathsf{T}$. Consider a network with one linear output-unit $\zeta$:

$$\zeta = \sum_{i=1}^{N} w_i \xi_i \ . \tag{8}$$

Under Oja's learning rule

$$w_i \leftarrow w_i + \delta w_i \ , \text{ where } \delta w_i = \eta \zeta (\xi_i - \zeta w_i) \text{ and } \eta > 0 \ , \tag{9}$$

the weight vector $\boldsymbol{w}$ converges to a steady state $\boldsymbol{w}^* = (w_1^*, \ldots, w_N^*)^\mathsf{T}$.

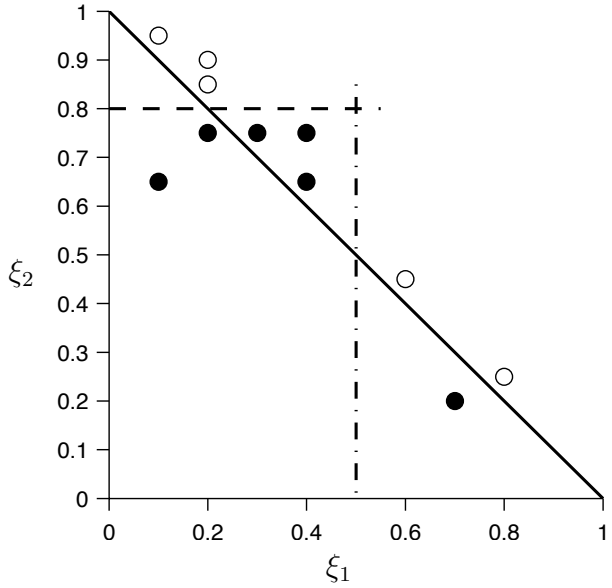


Figure 3: Question 3. Input patterns for the problem in question 3. The target output for each pattern $\mu$ is either $\zeta^{(\mu)} = 0$ (white circles) or $\zeta^{(\mu)} = 1$ (black circles). The three lines illustrate a solution to the problem by a multilayer perceptron (details in task **3b)**.

a) Compute the steady-state weight vector $\boldsymbol{w}^*$ for the input patterns shown in Fig. 4. Determine the maximal principal-component direction of the input data. Discuss: how is the steady-state weight vector $\boldsymbol{w}^*$ in this case related to the maximal principal-component direction of the input data? (**0.5**p)

b) Now compute the steady-state weight vector $\boldsymbol{w}^*$ for the input patterns shown in Fig. 5 instead. Determine the maximal principal-component direction of the input data. Discuss: how is the steady-state weight vector $\boldsymbol{w}^*$ in this case related to the maximal principal-component direction of the input data? Compare your findings to those obtained in task **4a)**. Discuss. (**1**p)

**5. Kohonen algorithm.** The update rule for a Kohonen network is:

$$w_{ij} \leftarrow w_{ij} + \delta w_{ij}, \text{ where } \delta w_{ij} = \eta \Lambda(i, i_0)(\xi_j - w_{ij}) \ . \tag{10}$$

Here $w_{ij}$ ($i = 1, \ldots, M$, and $j = 1, \ldots, N$) denotes the weight connecting the $i$-th output neuron to the $j$-th input neuron, $\eta > 0$ is the learning rate, $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_N)^\mathsf{T}$ is an $N$-dimensional input pattern chosen uniformly at random out of $p$ input patterns, and

$$\Lambda(i, i_0) = \exp(-\frac{|\boldsymbol{r}_i - \boldsymbol{r}_{i_0}|^2}{2\sigma^2}) \tag{11}$$

is a neighbouring function with width $\sigma > 0$. In Eq. (11), $\boldsymbol{r}_i$ denotes the position of the $i$-th output neuron in the output array. Finally, $i_0$ denotes

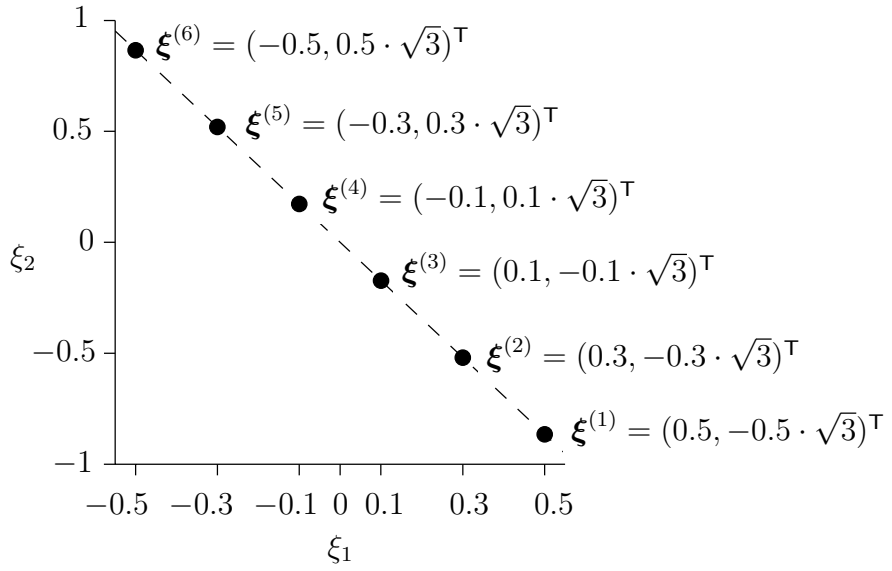


Figure 4: Question **4a)**. Input patterns $\boldsymbol{\xi}^{(1)}, \ldots, \boldsymbol{\xi}^{(6)}$ in input space $(\xi_1, \xi_2)^\mathsf{T}$.

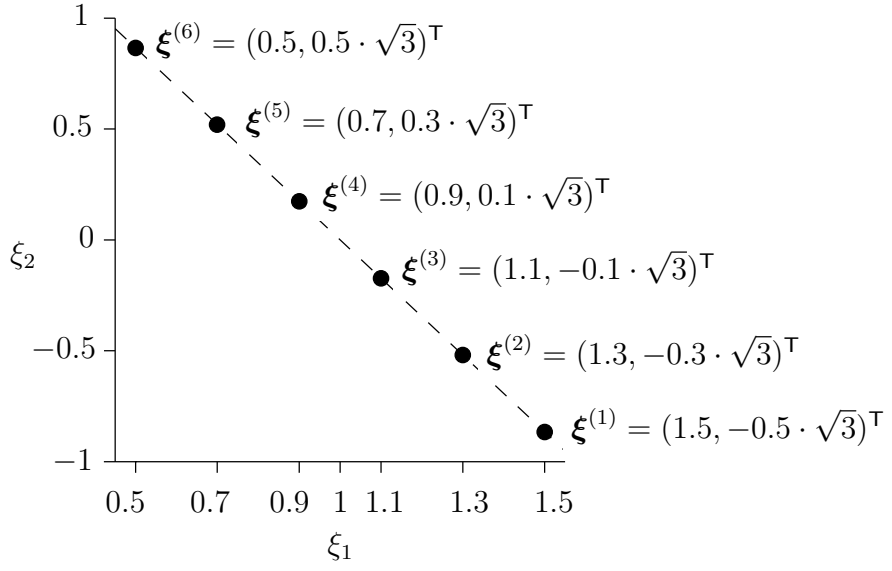


Figure 5: Question **4b)**. Input patterns $\boldsymbol{\xi}^{(1)}, \ldots, \boldsymbol{\xi}^{(6)}$ in input space $(\xi_1, \xi_2)^\mathsf{T}$.

the index of the winning output neuron for the input pattern $\boldsymbol{\xi}$, i.e the neuron satisfying

$$\sum_{j=1}^{N}(w_{i_0 j}-\xi_j)^2 \leq \sum_{j=1}^{N}(w_{ij}-\xi_j)^2, \text{ for all } i=1,\ldots,M \ . \tag{12}$$

a) Explain and discuss the implementation of Kohonen's algorithm in a computer program. In the discussion refer to and explain the terms listed next.

- (Un)supervised learning. Explain the difference between supervised and unsupervised learning, and state whether a Kohonen network is used for supervised or unsupervised learning.
- Initialisation of the algorithm.
- Output array.
- Neighbourhood function. Discuss the limit of $\sigma \to 0$.
- Ordering phase and convergence phase. Explain differences between these two phases: what is the aim of the former, and what is the aim of the latter phase? How are these aims achieved in practice, in terms of the parameters for the learning rate and the width of the neighbourhood function? (*You are not asked to suggest specific values for these parameters, but rather to explain whether or not these parameters are kept constant during learning in the different phases and, if not, how are they usually set to change during learning.*)

Your answer must not be longer than one A4 page. (**1**p)

b) Assume that the input data $\boldsymbol{\xi} = (\xi_1, \xi_2)^\mathsf{T}$ to a Kohonen network is uniformly distributed in two dimensions within the area shown by the black region in Fig. 6, and zero outside of this region. Furthermore, assume that the output array in this Kohonen network is one-dimensional, and that the number of output units ($M$) is large, yet much smaller than the total number of input patterns. Illustrate the algorithm described in **5a)** by schematically drawing the weight vectors $\boldsymbol{w}_i = (w_{i1}, w_{i2})^\mathsf{T}$ for $i = 1, \ldots, M$ in the input space at the start of learning, at the end of the ordering phase, and at the end of the convergence phase. (**1**p)

**6. Deep learning.** The parity function outputs 1 if and only if the input sequence of $n$ binary numbers has an odd number of ones, and zero otherwise. The parity function for $n = 2$ is also known as the Boolean XOR function.

a) The XOR function can be represented using a multilayer perceptron with an input layer of size 2, a fully connected hidden layer of size 2, an output layer of size 1, and by applying the Heaviside activation function

$$\theta(x) = \begin{cases} 1, \text{ for } x > 0 \ , \text{ and} \\ 0, \text{ for } x < 0 \end{cases} \tag{13}$$

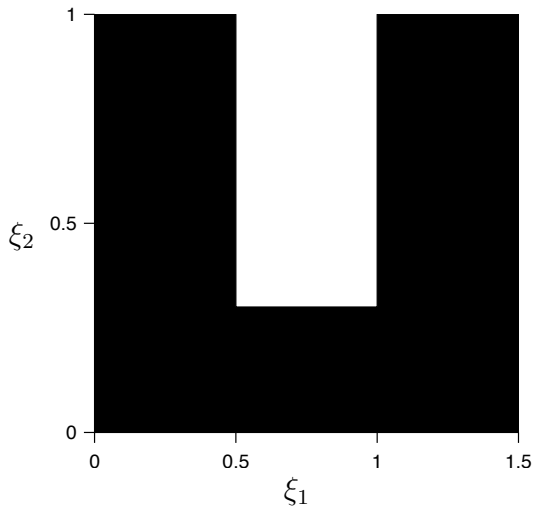


Figure 6: Question **5b)**. The pattern coloured black depicts the area inside which points are uniformly distributed. The properties of this distribution are to be learned by the Kohonen network, as explained in question **5b)**.

in all layers. Determine suitable weight vectors $\boldsymbol{W}_i$ and thresholds $\Theta_i$ for the two hidden nodes ($i = 1, 2$), as well as the weight vector $\boldsymbol{w}_1$ and the threshold $\theta_1$ for the output node that represent the XOR problem. (**0.5**p)

b) Illustrate the problem graphically in the input space, and indicate the planes determined by the weight vectors $\boldsymbol{W}_i$ and thresholds $\Theta_i$ that you determined in **6a)**. In a separate graph, illustrate the transformed input data in the hidden space and denote the plane determined by the weight vector $\boldsymbol{w}_1$ and the threshold $\theta_1$ that you determined in **6a)**. (**0.5**p)

c) Describe how you can combine several of the small XOR multilayer perceptrons analysed in **6a)-6b)** to create a deep network that computes the parity function for $n > 2$. Explain how the total number of nodes in the network grows with the size $n$ of the input. (**1**p)