TEST EXAM for
ARTIFICIAL NEURAL NETWORKS

COURSE CODES: **FFR 135, FIM720GU, PhD**

Time:	October 22, 2016, at $14^{00} - 18^{00}$
Place:	Lindholmen-salar
Teachers:	Bernhard Mehlig, 073-420 0988 (mobile)
	Marina Rafajlović, 076-580 4288 (mobile), visits once at 14^{30}
Allowed material:	Mathematics Handbook for Science and Engineering
Not allowed:	any other written or printed material, calculator

Maximum score on this exam: 12 points.
Maximum score for homework problems: 12 points.
CTH ≥14 passed; ≥17.5 grade 4; ≥22 grade 5,
GU ≥14 grade G; ≥ 20 grade VG.

1. One-step error probability in deterministic Hopfield model. In the deterministic Hopfield model, the state S_i of the i-th neuron is updated according to the Mc-Culloch Pitts rule

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

Here N is the number of neurons in the model, w_{ij} are the weights, and p patterns $\boldsymbol{\zeta}^{(\mu)} = (\zeta_1^{(\mu)}, \ldots, \zeta_N^{(\mu)})^\mathsf{T}$ are stored in the network by assigning

$$w_{ij} = \frac{1}{N} \sum_{\mu=1}^{p} \zeta_i^{(\mu)} \zeta_j^{(\mu)} \ \ \text{for } i \neq j , \text{ and } w_{ii} = 0 . \tag{2}$$

a) Apply pattern $\boldsymbol{\zeta}^{(\nu)}$ to the network. Derive the condition for bit $\zeta_i^{(\nu)}$ of this pattern to be stable after a single asynchronous update according to Eq. (1). Rewrite this stability condition using the "cross-talk term". (**0.5**p)

b) Take random patterns, $\zeta_j^{(\mu)} = 1$ or -1 with probability $\frac{1}{2}$. Derive an approximate expression for the probability that bit $\zeta_i^{(\nu)}$ is stable after a single asynchronous update according to Eq. (1), valid for large p and N. (**1**p)

2. Hopfield model: recognition of one pattern. The pattern shown in Fig. 1 is stored in a Hopfield model using Hebb's rule:

$$w_{ij} = \frac{1}{N} \zeta_i^{(1)} \zeta_j^{(1)} . \tag{3}$$

Figure 1: Question 2. Stored pattern $\boldsymbol{\zeta}^{(1)}$ with $N = 4$ bits, $\zeta_1^{(1)} = 1$, and $\zeta_i^{(1)} = -1$ for $i = 2, 3, 4$.
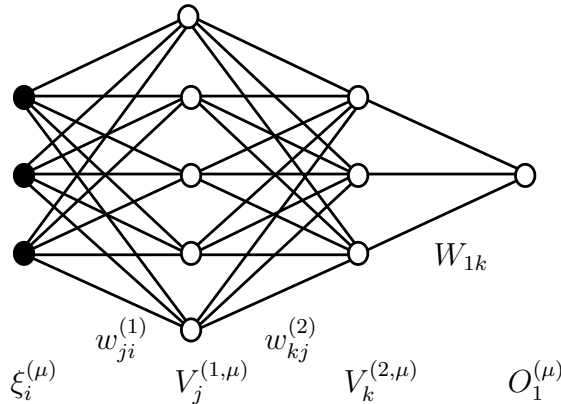

Figure 2: Question 3. Multi-layer perceptron with three input units, two hidden layers, and one output unit.

There are 2^4 four-bit patterns. Apply each of these to the Hopfield model and apply one synchronous update according to Eq. (1). List the patterns you obtain and discuss your results. (**1**p)

3. Back-propagation I. To train a multi-layer perceptron using back-propagation one needs update formulae for the weights and thresholds in the network. Derive these update formulae for the network shown in Fig. 2. The weights for the first and second hidden layer, and for the output layer are denoted by $w_{ji}^{(1)}$, $w_{kj}^{(2)}$, and W_{1k}. The corresponding thresholds are denoted by $\theta_j^{(1)}$, $\theta_k^{(2)}$, and Θ_1, and the activation function by $g(\cdots)$. The target value for input pattern $\boldsymbol{\xi}^{(\mu)}$ is $\zeta_1^{(\mu)}$. (**2**p)

4. Back-propagation II. Explain how to train a multi-layer perceptron by back-propagation. Draw a flow-chart of the algorithm. In your discussion, refer to and explain the following terms: "forward propagation", "backward propagation", "hidden layer", "energy function", "gradient descent", "local energy minima", "batch mode", "training set", "validation set", "classification error", "overfitting". Your answer must not be longer than one A4 page. (**1**p)

5. Oja's rule. The aim of unsupervised learning is to construct a network that learns the properties of a distribution $P(\boldsymbol{\xi})$ of input patterns

$\boldsymbol{\xi} = (\xi_1, \ldots, \xi_N)^\mathsf{T}$. Consider a network with one linear output-unit ζ:

$$\zeta = \sum_{j=1}^{N} w_j \xi_j \ . \tag{4}$$

Under Oja's learning rule

$$\delta w_j = \eta \zeta (\xi_j - \zeta w_j) \tag{5}$$

the weight vector $\boldsymbol{w}$ converges to a steady state $\boldsymbol{w}^*$ with components $w_j^\star$. The steady state has the following properties:

i) $|\boldsymbol{w}^\star|^2 \equiv \sum_{j=1}^{N} (w_j^*)^2 = 1$.

ii) $\boldsymbol{w}^\star$ is the leading eigenvector of the matrix $\mathbb{C}$ with elements $C_{ij} = \langle \xi_i \xi_j \rangle$. Here $\langle \cdots \rangle$ denotes the average over $P(\boldsymbol{\xi})$.

iii) $\boldsymbol{w}^\star$ maximises $\langle \zeta^2 \rangle$.

a) Show that iii) follows from i) and ii). (**1**p)

b) Prove i) and ii), assuming that $\boldsymbol{w}^\star$ is a steady state. (**1.5**p)

c) Write down a generalisation of Oja's rule that learns the first M principal components for zero-mean data, $\langle \boldsymbol{\xi} \rangle = 0$. Discuss: how does the rule ensure that the weight vectors remain normalised? (**0.5**p)

6. Kohonen's algorithm. The update rule for a Kohonen network reads:

$$\delta w_{ij} = \eta \Lambda(i, i_0)(\xi_j - w_{ij}) \ . \tag{6}$$

Here i_0 labels the winning unit for pattern $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_N)^\mathsf{T}$. The neighbourhood function $\Lambda(i, i_0)$ is a Gaussian

$$\Lambda(i, i_0) = \exp\left(-\frac{|\boldsymbol{r}_i - \boldsymbol{r}_{i_0}|^2}{2\sigma^2}\right) \tag{7}$$

with width σ, and $\boldsymbol{r}_i$ denotes the position of the i-th output neuron in the output array.

a) Explain the meaning of the parameter σ in Kohonen's algorithm. Discuss the nature of the update rule in the limit of $\sigma \to 0$. (**0.5**p)

b) Discuss and explain the implementation of Kohonen's algorithm in a computer program. In the discussion, refer to and explain the following terms: "output array", "neighbourhood function", "ordering phase", "convergence phase", "kinks". Your answer must not be longer than one A4 page. (**1**p)

c) Assume that the input data are two-dimensional, and uniformly distributed within the unit disk. Illustrate the algorithm described in b) by schematically drawing the input-space positions of the weight vectors $\boldsymbol{w}_i$ at the start of learning, at the end of the ordering phase, and at the end of the convergence phase. Assume that the output array is one dimensional, and that the number of output units M is large, yet much smaller than the

μ	$\xi_1^{(\mu)}$	$\xi_2^{(\mu)}$	$\zeta_1^{(\mu)}$
1	1	1	0
2	1	0	1
3	0	1	1
4	0	0	0

Table 1: Question 7. Inputs and target values for the XOR problem.

number of input patterns. (**0.5**p)

7. Radial basis functions. Consider the Boolean XOR problem, Table 1.
a) Show that this problem cannot be solved by a simple perceptron. (**0.5**p)
b) The problem can be solved upon transforming input space using radial-basis functions, and applying a simple perceptron to the transformed input data. Show that the two-dimensional Boolean XOR problem can be solved using the following two radial basis functions:

$$g_1(\boldsymbol{\xi}^{(\mu)}) = \exp(-|\boldsymbol{\xi}^{(\mu)} - \boldsymbol{w}_1|^2) \quad \text{with } \boldsymbol{w}_1 = (1, 1)^\mathsf{T},$$
$$g_2(\boldsymbol{\xi}^{(\mu)}) = \exp(-|\boldsymbol{\xi}^{(\mu)} - \boldsymbol{w}_2|^2) \quad \text{with } \boldsymbol{w}_2 = (0, 0)^\mathsf{T}. \tag{8}$$

Draw the positions of the four input patterns in the transformed space $(g_1, g_2)^\mathsf{T}$, encoding the different target values. [Hint: to compute the states of input patterns in the transformed space $(g_1, g_2)^\mathsf{T}$, use the following approximations: $\exp(-1) \approx 0.37$, $\exp(-2) \approx 0.14$.] Explain the term "decision boundary", draw a decision boundary for the XOR problem, and give the corresponding weights and thresholds for the simple perceptron. (**1**p)