CHALMERS, GÖTEBORGS UNIVERSITET

RE-EXAM for
ARTIFICIAL NEURAL NETWORKS

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

Time:	January 8, 2019, at $14^{00} - 18^{00}$
Place:	Johanneberg
Teachers:	Bernhard Mehlig, 073-420 0988 (mobile)
	Johan Fries, 070-370 1272 (mobile), visits once at 14^{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. 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 rule

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

There are N neurons. The weights w_{ij} are stored in the network according to Hebb's rule. There are two alternative ways of implementing Hebb's rule.

i) The first alternative is to assign

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

ii) The second alternative is

$$w_{ij} = \frac{1}{N}\sum_{\mu=1}^{p} x_i^{(\mu)} x_j^{(\mu)} \quad \text{for all } i \text{ and } j\,. \tag{3}$$

The pattern bits $x_i^{(\mu)}$ take the values 1 or -1, and the pattern index μ ranges from 1 to p. Assume random patterns ($x_i^{(\mu)} = 1$ or -1 with probability 0.5). Derive approximate expressions for the one-step error probability $P_{\text{error}}^{(t=1)}$ in the limit of large p and N, for two cases:

(a) Weights given by Equation (2). (**1**p).

(b) Weights given by Equation (3). (**1**p).

(c) For both cases, sketch the dependence of $P_{\text{error}}^{(t=1)}$ upon the storage capacity $\alpha = p/N$. Examine and explain the limiting behaviours as $\alpha \to \infty$. (**1**p).

2. Linear separability of Boolean functions. Consider Boolean functions with three inputs $x_i^{(\mu)}$ $(i = 1, 2, 3)$ and one output

$$O^{(\mu)} = \text{sgn}(\sum_{i=1}^{3} w_i x_i^{(\mu)} - \theta)\,. \tag{4}$$

Here w_i $(i = 1, 2, 3)$ are the weights, θ is a threshold assigned to the output, and $\mu = 1, \ldots, 2^3$. Assume that four targets equal 1, and 4 targets equal -1. An example of such a function is given in Table 1.

(a) Illustrate the function in Table 1 graphically. Colour inputs with targets $= 1$ black, and inputs with targets $= -1$ white. Using your illustration explain why this Boolean function can be solved by a simple perceptron with three inputs and one output. Draw a solution to the problem. Compute the weights w_i and the threshold θ corresponding to your solution. (**0.5**p)

(b) How many three-dimensional Boolean functions are there with 4 targets $= 1$, and 4 targets $= -1$? Describe how you arrive at the answer. (**0.5**p)

(c) How many of the Boolean functions you found in (b) can be solved by a simple perceptron with three input units and one output unit? Describe how you arrive at the answer. *Hint:* use symmetries to reduce the number of cases. (**1**p) .

3. Stochastic gradient descent. To train a multi-layer perceptron using stochastic gradient descent one needs update formulae for weights and thresholds. Derive these update formulae for *sequential training* using backpropagation for the network shown in Fig. 1. The weights for the first and second hidden layer, and for the output layer are denoted by $w_{jk}^{(1)}$, $w_{mj}^{(2)}$, and W_{1m}. The corresponding thresholds are denoted by $\theta_j^{(1)}$, $\theta_m^{(2)}$, and Θ_1, and the activation function by $g(\cdots)$. The target value for input pattern $\boldsymbol{x}^{(\mu)}$ is $t_1^{(\mu)}$, and the pattern index μ ranges from 1 to p. The energy function is $H = \frac{1}{2}\sum_{\mu=1}^{p}(t_1^{(\mu)} - O_1^{(\mu)})^2$. (**2**p).

$x_1^{(\mu)}$	$x_2^{(\mu)}$	$x_3^{(\mu)}$	$t^{(\mu)}$
-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

Table 1: Inputs $\boldsymbol{x}^{(\mu)} = [x_1^{(\mu)}, x_2^{(\mu)}, x_3^{(\mu)}]^\mathsf{T}$ and targets $t^{(\mu)}$ for a three-dimensional Boolean function. (Question **2**).

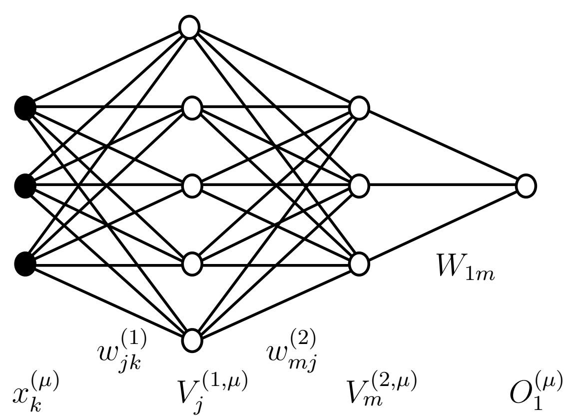

Figure 1: Multi-layer perceptron with three input terminals, two hidden layers, and one output. (Question **3**).

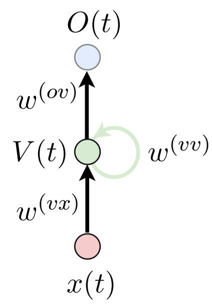

Figure 2: Recurrent network with one input unit $x(t)$ (red), one hidden neuron $V(t)$ (green) and one output neuron $O(t)$ (blue). (Question **4**).

4. Recurrent network. Figure 2 shows a simple recurrent network with one hidden neuron $V(t)$, one input $x(t)$ and one output $O(t)$. The network learns a time series of input-output pairs $[x(t), y(t)]$ for $t = 1, 2, 3, \ldots, T$. Here t is a discrete time index and $y(t)$ is the target value at time t (the targets are denoted by y to avoid confusion with the time index t). The hidden unit is initialised to a value $V(0)$ at $t = 0$. This network can be trained by backpropgation by *unfolding it in time*.

(a) Draw the unfolded network, label the connections using the labels shown in Figure 2, and discuss the layout (max half an A4 page). (**0.5**p).

(b) Write down the dynamical rules for this network, the rules that determine $V(t)$ in terms of $V(t-1)$ and $x(t)$, and $O(t)$ in terms of $V(t)$. Assume that both $V(t)$ and $O(t)$ have the same activation function $g(b)$. (**0.5**p).

(c) Derive the update rule for $w^{(ov)}$ for gradient descent on the energy function

$$H = \frac{1}{2}\sum_{t=1}^{T} E(t)^2 \quad \text{where } E(t) = y(t) - O(t)\,. \tag{5}$$

Denote the learning rate by η. *Hint:* the update rule for $w^{(ov)}$ is much simpler to derive than those for $w^{(vx)}$ and $w^{(vv)}$. (**1**p).

(d) Explain how recurrent networks are used for machine translation. Draw the layout, describe how the inputs are encoded. How is the *unstable-gradient problem* overcome? (Max 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 $P(\boldsymbol{x})$ of input patterns $\boldsymbol{x} = (x_1, \ldots, x_N)^{\mathsf{T}}$. 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 $\boldsymbol{w}$ converges to a steady state $\boldsymbol{w}^*$ with components w_j^*.

(a) Show that the steady state $\boldsymbol{w}^*$ is an eigenvector of the matrix $\mathbb{C}'$ with elements $C'_{ij} = \langle x_i x_j \rangle$. Here $\langle \cdots \rangle$ denotes the average over $P(\boldsymbol{x})$. (**1p**).

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