CHALMERS, GÖTEBORGS UNIVERSITET

RE-EXAM for
ARTIFICIAL NEURAL NETWORKS

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

Time:	January 4, 2022, at $14^{00} - 18^{00}$
Place:	Johanneberg
Teachers:	Bernhard Mehlig, 073-420 0988 (mobile)
	Ludvig Storm, visits at 14^{30} and 17^{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 >13.5 passed; >17 grade 4; >21.5 grade 5,
GU >13.5 grade G; > 19.5 grade VG.

---

1. Hopfield model with time-continuous dynamics. Consider a Hopfield net with continuous-time dynamics:

$$\tau \tfrac{\mathrm{d}}{\mathrm{d}t} n_i = -n_i + g(\sum_j w_{ij} n_j - \theta_i)$$

with $g(b) = (1 + \mathrm{e}^{-b})^{-1}$ and time scale τ. Show that the energy function

$$E = -\tfrac{1}{2} \sum_{ij} w_{ij} n_i n_j + \sum_i \theta_i n_i + \sum_i \int_0^{n_i} \mathrm{d}n \, g^{-1}(n)$$

cannot increase under the network dynamics if the weights are symmetric. Here g^{-1} is the inverse function of g, so that $g^{-1}\big(g(b)\big) = b$. *Hint:* use the fact that $g(b)$ is a monotonically increasing function of b. (**2**p).

2. Three-point probabilities in 3×3 bars-and-stripes data set. Demonstrate that a Boltzmann machine requires hidden units to learn the 3×3 data set shown in Figure 1(**a**). To this end, evaluate all eight three-point probabilities $P(x_1 = \pm 1, x_2 = \pm 1, x_3 = \pm 1)$ for x_1, x_2, and x_3 as shown in panel (**b**). Here $x_j = +1$ represents ■, and $x_j = -1$ stands for □. Check whether these three-point probabilities factorise. For example, does

$$P(x_1 = 1, x_2 = 1, x_3 = -1) = P(x_1 = 1, x_2 = 1) P(x_3 = -1)$$

hold, or not? Use your results to explain why a Boltzmann machine needs hidden units to learn the data set (**a**). Now consider the data set in Figure 1(**c**), only stripes. Explain why no hidden units are needed for (**c**). (2**p**).

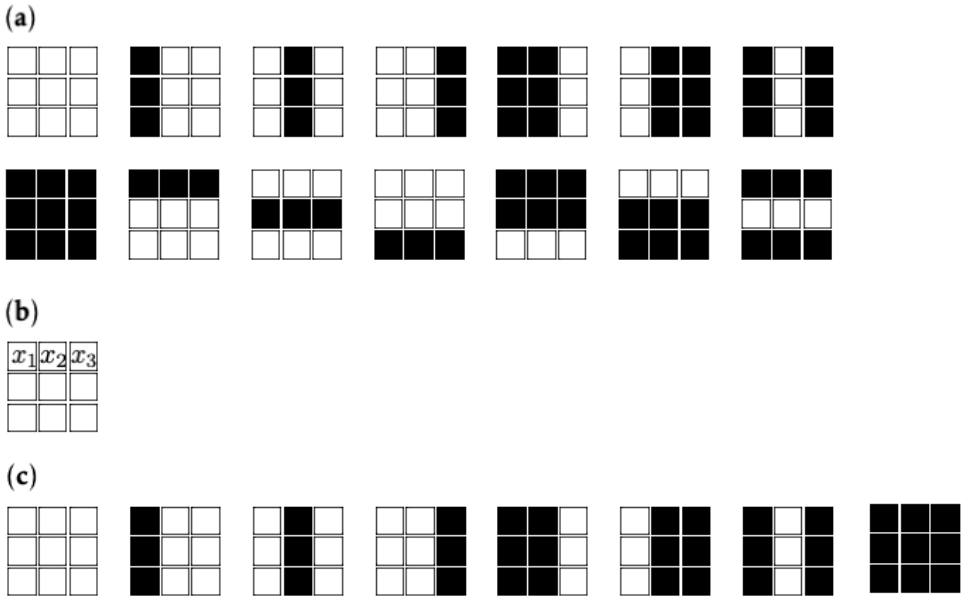

Figure 1: (**a**) 3×3 bars-and-stripes data set. The shown patterns occur with probability $P_{\text{data}} = \frac{1}{14}$, all other patterns have $P_{\text{data}} = 0$. (**b**) Data set with stripes only. The shown patterns occur with probability $P_{\text{data}} = \frac{1}{8}$, all other patterns have $P_{\text{data}} = 0$.

3. Linearly inseparable problem. A classification problem is given in Figure 2. Inputs $\mathbf{x}^{(\mu)}$ inside the gray triangle have targets $t^{(\mu)} = 1$, inputs outside the triangle have targets is $t^{(\mu)} = 0$. The problem can be solved by a perceptron with one hidden layer with three neurons $V_j^{(\mu)} = \theta_{\rm H}\left(-\theta_j + \sum_{k=1}^{2} w_{jk} x_k^{(\mu)}\right)$, for $j = 1, 2, 3$. The network output is computed as $O^{(\mu)} = \theta_{\rm H}\left(-\Theta + \sum_{j=1}^{3} W_j V_j^{(\mu)}\right)$. Here $\theta_{\rm H}(b)$ is the Heaviside function:

$$\theta_{\rm H}(b) = \begin{cases} 1 \text{ if } b > 0 \\ 0 \text{ otherwise} \end{cases}$$

Find weights w_{jk}, W_j and thresholds θ_j, Θ that solve the classification problem (2**p**).

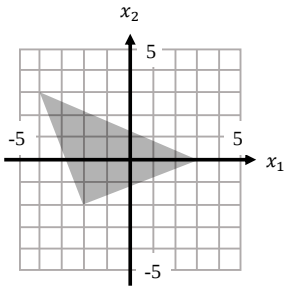

Figure 2: Linearly inseparable classification problem, Question 3.

4. Convolutional neural network. The two patterns shown in Figure 3 are processed by a very simple convolutional network that has one convolution layer with one single 2×2 kernel with ReLU neurons, zero threshold, and stride (1,1). The resulting feature map is fed into a 2×3 max-pooling layer with stride (1,1). Finally there is a fully connected classification layer with one output neuron with the Heaviside activation function. Determine weights of the kernel and weights and thresholds of the classification layer that allow to classify the two patterns into different classes (2**p**).

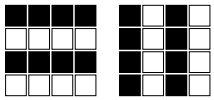

Figure 3: Patterns to be classified by convolutional network. Black squares = 1, white squares = 0. Question 4.

5. Lyapunov exponent in deep neural networks. The error in a multilayer perceptron propagates backwards according to the rule:

$$\delta_i^{(l-1)} = \sum_j^N \delta_j^{(l)} w_{ji}^{(l)} g'(b_i^{(l-1)}) \,.$$

Here $g' = \mathrm{d}g/\mathrm{d}b$ is the derivative of the activation function. Assume that the network is trained on random input patterns with independent bits. Further assume that the weights are random, Gaussian distributed with mean zero and variance $\langle w_{ij} w_{kl} \rangle = \sigma_w^2 \delta_{ik} \delta_{jl}$, and that the thresholds are set to zero. Here δ_{ik} is the Kronecker delta, don't confuse it with the error!

(a) Compute the mean of the error $\delta_i^{(l-1)}$ in the limit $N \to \infty$ neglecting any correlations between local fields, weights, or errors (0.5**p**).

(b) Show that the variance of the error in the limit $N \to \infty$ obeys the recursion

$$\langle (\delta_i^{(l-1)})^2 \rangle = N \sigma_w^2 \langle (\delta_j^{(l)})^2 \rangle \langle [g'(b_i^{(l-1)})]^2 \rangle \,,$$

under the same assumptions as in task (a) (0.5**p**).

(c) It can be shown that the distribution of the local fields $b_j^{(l)}$ converges to a Gaussian with zero mean and a fixed variance σ_f^2, for large N and many layers. Assuming that the distribution has this form, derive an approximation for the maximal Lyapunov exponent. It is defined as

$$\lambda_1 = \log \left| \delta^{(l-1)} / \delta^{(l)} \right|.$$

Explain why σ_w^2 should be chosen to be on the order N^{-1} for the network to learn well. *Hint*: write $\langle [g'(b_i^{(l-1)})]^2 \rangle$ as an integral expression, you do not need to evaluate the integral. (1**p**).

6. Backpropagation. To train a multi-layer perceptron by gradient descent one needs update formulae for weights and thresholds. Derive these update formulae for sequential training using backpropagation for the netweork shown in Fig. 4. The weights for the hidden layer are denoted by w_{jk}, and those for the output layer by W_{1j}. The corresponding thresholds are denoted by θ_j and Θ_1, and the activation function by $g(\dots)$. The target values for input patterns $\mathbf{x}^{(\mu)}$ 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**).

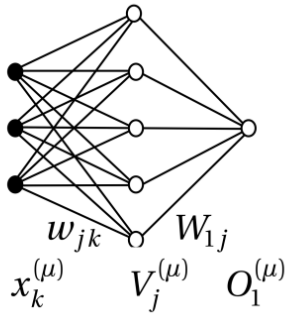

Figure 4: Multi-layer perceptron with three input terminals, one hidden layer, and one output. Question 6.