CHALMERS, GÖTEBORGS UNIVERSITET

EXAM for
ARTIFICIAL NEURAL NETWORKS

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

Time:	
Place:	
Teachers:	
	August 19, 2021, at $14-18$
Allowed material:	Mathematics Handbook for Science and Engineering
Not allowed:	Any other written material, calculator

Maximum score on this exam: 12 points. Each question gives at most 2 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.

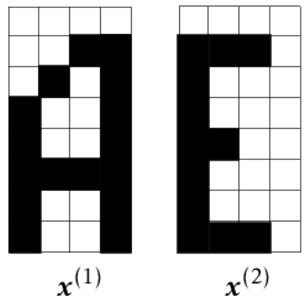

Figure 1: Input patterns $\boldsymbol{x}^{(1)}$ and $\boldsymbol{x}^{(2)}$ with 0/1 bits (□ corresponds to x_i=0 and ■ to x_i=1). Question 1.

1. Convolutional net. Construct a simple convolutional network to classify the two patterns shown in Fig. 1. Assume that the convolution layer has one single 3×3 kernel with ReLU units with weights w_{ij} that can take the values 0 or 1, and with threshold θ. The resulting feature map connects to a 2×2 max-pooling layer. Finally there is a fully connected output layer with one output $O^{(\mu)}$ with Heaviside activation function, with weights W_k and threshold Θ. Determine the parameters of the network (weights, thresholds, strides, padding if needed) so that network outputs $O^{(1)} = 0$ for input pattern $\boldsymbol{x}^{(1)}$, and $O^{(2)} = 1$ for input pattern $\boldsymbol{x}^{(2)}$.

2. Kullback-Leibler divergence. Show that the Kullback-Leibler divergence

$$D_{\mathrm{KL}} = \sum_{\mu=1}^{p} P_{\mathrm{data}}(\boldsymbol{x}^{(\mu)}) \log[P_{\mathrm{data}}(\boldsymbol{x}^{(\mu)})/P_{\mathrm{B}}(\boldsymbol{s} = \boldsymbol{x}^{(\mu)})] \tag{1}$$

is non-negative, and that it assumes its global minimum $D_{\mathrm{KL}} = 0$ when the Boltzmann distribution $P_{\mathrm{B}}(\boldsymbol{s} = \boldsymbol{x}^{(\mu)})$ equals the data distribution $P_{\mathrm{data}}(\boldsymbol{x}^{(\mu)})$. Show that minimising D_{KL} is equivalent to maximising the log-likelihood function

$$\log \mathscr{L} = \log \prod_{\mu=1}^{p} P_{\mathrm{B}}(\boldsymbol{s} = \boldsymbol{x}^{(\mu)}) = \sum_{\mu=1}^{p} \log P_{\mathrm{B}}(\boldsymbol{s} = \boldsymbol{x}^{(\mu)}) \,. \tag{2}$$

Note: these properties are used to derive training algorithms for Boltzmann machines.

x_1	x_2	x_3	t
0	0	0	0
0	0	1	1
0	1	0	1
1	0	0	1
0	1	1	0
1	0	1	0
1	1	0	0
1	1	1	1

Table 1: Value table for a three-dimensional Boolean function. Question 3.

3. Boolean function I. Table 1 shows the value table for a three-dimensional Boolean function. Demonstrate that the function is not linearly separable by drawing it in three-dimensional input space. Construct a network with hidden layers that represents this function. *Hint:* one possibility is to wire together several two-dimensional XOR networks.

4. Boolean function II. The parity function can be viewed as a generalisation of the XOR function to $N > 2$ input dimensions, because it becomes the XOR function for $N = 2$. Another way to generalise the XOR function to $N > 2$-dimensional inputs is to define a Boolean function that gives unity if exactly one of its inputs equals unity. Otherwise the function evaluates to zero. Construct networks that represent this function, for $N = 3$ and $N = 4$.

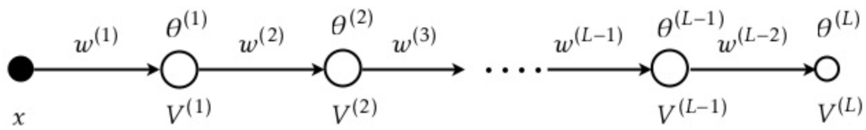

Figure 2: Chain of neurons to be trained by backpropagation. Question 5.

5. Backpropagation
Figure 2 shows a chain of neurons $V^{(\ell)} = g(w^{(\ell)}V^{(\ell-1)} - \theta^{(\ell)}) \equiv g(b^{(\ell)})$ with energy function $H = \frac{1}{2}(t - V^{(L)})^2$. Derive the backpropagation algorithm for this chain. (a) Show that the learning rule reads

$$\delta w^{(\ell)} = \eta \delta^{(\ell)} V^{(\ell-1)} \quad \text{with} \quad \delta^{(\ell-1)} = \big(t - V^{(L)}\big)\frac{\partial V^{(L)}}{\partial V^{(\ell-1)}} g'(b^{(\ell-1)})\,. \qquad (3)$$

(b) Evaluate the partial derivative $\partial V^{(L)}/\partial V^{(\ell-1)}$.

6. Reinforcement learning. Train a binary stochastic neuron to maximise its average immediate reward. The neuron computes

$$y = \begin{cases} +1 & \text{with probability } p(b)\,, \\ -1 & \text{with probability } 1 - p(b)\,, \end{cases} \tag{4}$$

where $b = \boldsymbol{w} \cdot \boldsymbol{x}$ is the local field (no thresholds), and $p(b) = (1 + e^{-2b})^{-1}$. Given inputs $\boldsymbol{x}$ and output y, the environment provides a stochastic reward $r(\boldsymbol{x}, y) = \pm 1$ drawn from a *reward distribution* $p_{\text{reward}}(\boldsymbol{x}, y)$:

$$r(\boldsymbol{x}, y) = \begin{cases} +1 & \text{with probability } p_{\text{reward}}(\boldsymbol{x}, y), \\ -1 & \text{with probability } 1 - p_{\text{reward}}(\boldsymbol{x}, y). \end{cases} \tag{5}$$

The average immediate reward for a given input $\boldsymbol{x}$ is defined as

$$\langle r \rangle = \sum_{y=\pm 1} \big\langle r(\boldsymbol{x}, y) P(y|\boldsymbol{x}) \big\rangle_{\text{reward}}\,. \tag{6}$$

The average $\langle \cdots \rangle_{\text{reward}}$ is over the response of the environment determined by the stationary reward distribution $p_{\text{reward}}(\boldsymbol{x}, y)$. Further, $P(y|\boldsymbol{x})$ is the probability that the stochastic neuron outputs y given $\boldsymbol{x}$. Derive a learning rule by gradient ascent. (a) Show that

$$\frac{\partial \langle r \rangle}{\partial w_n} = \big\langle r(\boldsymbol{x}, y) \big[y - \tanh(b) \big] \big\rangle x_n\,. \tag{7}$$

The average is over the possible outputs, and over the response of the environment. (b) From this expression derive a weight increment δw_n that is an unbiased estimator of the gradient of the average immediate reward (6). Explain the term *unbiased estimator.*