CHALMERS, GÖTEBORGS UNIVERSITET

RE-EXAM for
ARTIFICIAL NEURAL NETWORKS

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

Time:	January 8, 2020, at 14^{00}
Place:	Maskin-salar
Teachers:	Bernhard Mehlig, 073-420 0988 (mobile)
	Marina Rafajlovic, 076-580 4288 (mobile)
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. Higher-order Hopfield nets. Consider a Hopfield network with the energy function

$$H = -\frac{1}{2}\sum_{i,j} w_{ij}^{(2)} s_i s_j - \frac{1}{6}\sum_{i,j,k} w_{ijk}^{(3)} s_i s_j s_k \ . \tag{1}$$

Here, s_i $(i = 1, \ldots, N)$ is the state of neuron i, $w_{ij}^{(2)}$ and $w_{ijk}^{(3)}$ are weights. The state of each neuron is either $+1$ or -1. The update rule for the state of neuron m is given by

$$s_m' = \mathrm{sgn}(b_m) \text{ with } b_m = -\frac{\partial H}{\partial s_m} \ . \tag{2}$$

Determine under which conditions on $w_{ij}^{(2)}$ and $w_{ijk}^{(3)}$ the energy function (1) cannot increase in a single step of the asynchronous update (2). *Hint:* the sought conditions must be independent of the states of the neurons. (**2**p).

2. Radial basis functions. Radial basis-function nets make use of the fact that it is easier to separate patterns when they are embedded in a higher-dimensional space.

(a) Consider classification problems with p pattern vectors $\boldsymbol{u}^{(\mu)}$ embedded in m-dimensional space and with random targets, $t^{(\mu)} = \pm 1$ with equal probability. A problem is homogeneously linearly separable if there is a m-dimensional weight vector $\boldsymbol{W}$ so that $\boldsymbol{W} \cdot \boldsymbol{u} = 0$ is a valid decision boundary that goes through the origin:

$$\boldsymbol{W} \cdot \boldsymbol{u}^{(\mu)} > 0 \quad \text{if} \quad t^{(\mu)} = 1 \quad \text{and} \quad \boldsymbol{W} \cdot \boldsymbol{u}^{(\mu)} < 0 \quad \text{if} \quad t^{(\mu)} = -1\,. \tag{3}$$

Show that the probability for $p = 3$ patterns to be separable in $m = 2$ dimensions equals $\frac{3}{4}$. *Hint:* assume that the patterns together with the origin are in general position. (**1**p).

(b) Radial basis functions

$$u_\nu(\boldsymbol{x}) = \exp\big(-\tfrac{1}{2}\,|\boldsymbol{x} - \boldsymbol{w}_\nu|^2\big) \tag{4}$$

produce localised outputs, and this makes it possible to map the input patterns into localised regions that can be classified with a single linear unit with weight vector $\boldsymbol{W}$. Imagine for a moment that we have as many radial basis functions as input patterns. Then one can simply take $\boldsymbol{w}_\nu = \boldsymbol{x}^{(\nu)}$ in Eq. (4), for $\nu = 1, \ldots, p$. Show that this gives the following solution of the classification problem:

$$W_\mu = \sum_\nu [\mathbb{U}^{-1}]_{\mu\nu} t^{(\nu)}\,, \tag{5}$$

if all patterns are pairwise different. Here $\mathbb{U}$ is the matrix with entries $U_{\nu\mu} = u_\nu(\boldsymbol{x}^{(\mu)})$. Explain qualitatively why one can usually get away with fewer radial basis functions. (**1**p).

3. Backpropagation. To train a multi-layer perceptron using stochastic gradient descent one requires update formulae for the weights and thresholds in the network. Derive these update formulae for the network shown in Figure 1 using the stochastic gradient-descent algorithm with constant learning rate η, mini-batch size $m_B = 1$, no momentum, and no regularisation. The weights for the hidden layer and for the output layer are denoted by w_{mn} and W_{1m}, and the corresponding thresholds are denoted by θ_m, and Θ_1. The activation function $g(\cdots)$ is used in both the hidden layer and in the output layer. The target value for input pattern $\boldsymbol{x}^{(\mu)}$ is $t_1^{(\mu)}$, and the network output is $O_1^{(\mu)}$. The energy function is $H = \frac{1}{2}\sum_{\mu=1}^{p}\big(t_1^{(\mu)} - O_1^{(\mu)}\big)^2$, where p denotes the number of training patterns. (**2**p).

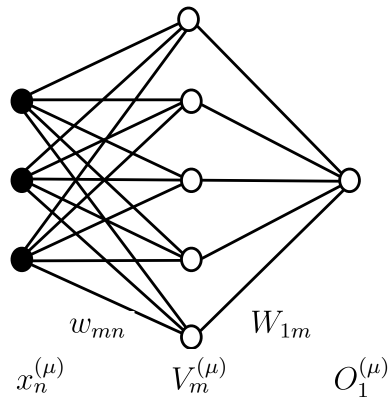

Figure 1: Multi-layer perceptron (question **3**). The perceptron has three input units, one hidden layer, and one output unit.

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

$$fH = \frac{1}{2}\sum_{t=1}^{T} E(t)^2 \quad \text{where } E(t) = y(t) - O(t)\,. \tag{6}$$

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).

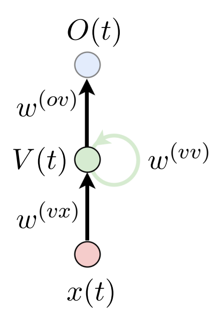

Figure 2: Recurrent network with one input unit $x(t)$ (red), one hidden neuron $V(t)$ (green) and one output neuron $O(t)$ (blue). (Question **4**).

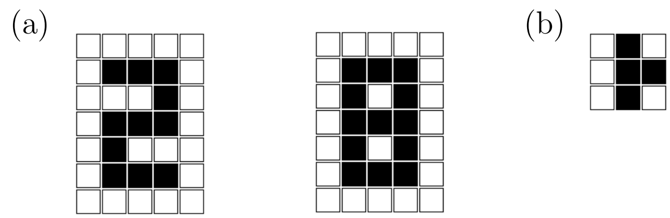

Figure 3: (a) Input patterns with 0/1 bits (□ corresponds to $x_i = 0$ and ■ to $x_i = 1$). (b) 3×3 kernel of a feature map. ReLU units, zero threshold, weights 0 or 1 (□ corresponds to $w = 0$ and ■ to $w = 1$). (Question **6**).

5. Parity function. The parity function equals 1 if the input sequence of N binary numbers has an odd number of ones, and 0 otherwise. The parity function for $N = 2$ is also known as the Boolean XOR function.

(a) Show how the XOR function can be represented by a neural net with one hidden layer with two neurons. Determine all weights and thresholds and draw the input plane with input patterns and decision boundary. (1**p**).

(b) Show how a parity function with $N = 2^k$ inputs ($k = 1, 2, \ldots$) can be represented by a combination of XOR nets of sub-problem (a). Draw the resulting network for $k = 2$. Show that the total number of *neurons* in the entire net equals $3(2^k - 1)$. (1**p**).

6. Convolutional net. The two patterns shown in Figure 3(a) are processed by a very simple convolutional network that has one convolution layer with one single 3×3 kernel with ReLU units, zero threshold, and weights as given in Figure 3(b). Stride (1,1). The resulting feature map is fed into a 3×3 max-pooling layer with stride (1,1). Finally there is a fully connected classification layer with two output units with Heaviside activation functions.

(a) For both patterns determine the resulting feature map and the output of the max-pooling layer. (1**p**).

(b) Determine weights and thresholds of the classification layer that allow to classify the two patterns into different classes. (1**p**).