Exam in
Neural Networks and Learning Systems
TBMI26 / 732A55
Home exam - Part II

Date: 2021-06-09
Time: 14.00 - 16.30 (part 1) and **14.00 - 18.30** (part 2)
Teacher: Magnus Borga, Phone: 013-28 67 77

Read the instructions before answering the questions!

Part 2 Consists of four 5-point questions. These questions test deeper understanding and the ability to apply the knowledge to solve problems. All assumptions and calculations made should be presented. Reasonable simplifications may be done in the calculations. **This part needs to be submitted before 18:30**

Students with approved extended examination time (förlängd skrivtid) may divide this extended time between the two parts at their own discretion.

Write your answers by hand and then scan them using a scanner or mobile phone, or write the answers in a separate file using a word processor. The answers may be given in English or Swedish. **If you write by hand, please write clearly using block letters! (Do not use cursive writing.) Answers that are difficult to read, will be dismissed.** The exam should be submitted before the deadline in PDF format. Each part should be handed in as one single PDF file. **Do not use 'Okular' to produce PDFs as graphical elements might disappear!** The PDF files should be named with your LiU-ID followed by a the number of the part of the exam, e.g. 'abcde132-2'.

The maximum sum of points is 20 on each part. To pass the exam (grade 3/C) at least 13 points are required on part 1. For grade 4/B, an additional 10 points on part 2 are required and for grade 5/A, 15 points are required on part 2, in addition to pass part 1.

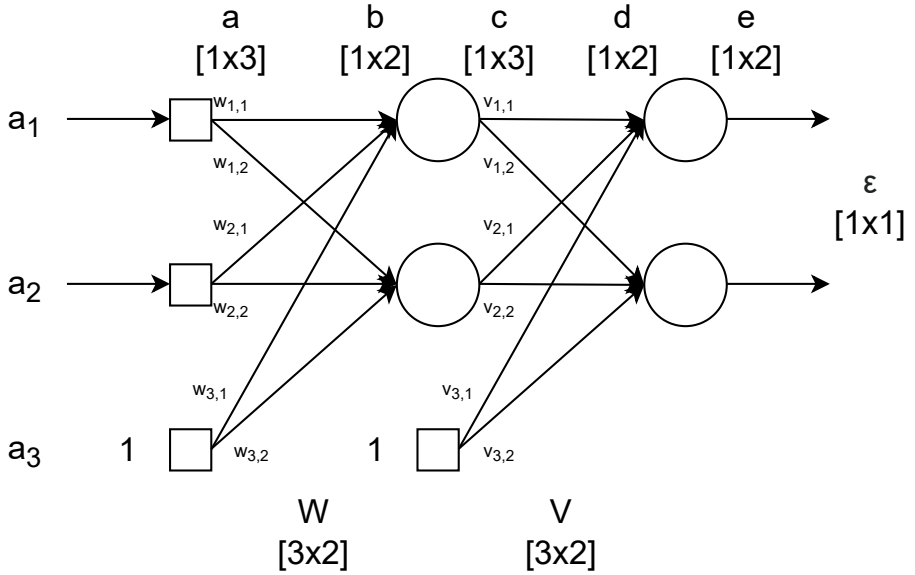
Note that all forms of collaboration or communication with any person except the course staff is strictly forbidden during the exam!

The result will be reported by 2021-06-30.

GOOD LUCK!

AID:	Exam Date: 2021-06-09
Course Code: TBMI26 / 732A55	Exam Code: TEN1

1. Consider the following neural network:



We want to use it to classify some data. However, we are not getting very good results, because we forgot to include any activation functions, that is, $\sigma(x) = x$.

a) Derive the *element-wise* weight update rules for online training that would minimize the square error cost function for this network. (2p)

We noticed our mistake, and decided to add tanh activation functions to both the hidden and the output layers.

b) Derive the *element-wise* weight update rules for the corrected network. (2p)

c) Describe the differences in the types of decision boundaries that can be learned by each network. (1p)

AID:	Exam Date: 2021-06-09
Course Code: TBMI26 / 732A55	Exam Code: TEN1

2. You have the following data (see Figure **??**):

$$\mathbf{X} = \begin{bmatrix} 1 & 1 & 1 & 2 & 2 & 2 & 3 & 3 \\ 1 & 2 & 3 & 1 & 2 & 3 & 2 & 3 \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 \end{bmatrix}$$

where $\mathbf{X}$ contains eight 2d-samples (one per column), and $\mathbf{y}$ contains classification labels for the corresponding samples. We have performed two iterations of AdaBoost using 'decision stumps' as weak classifiers on the data $\mathbf{X}$. We calculated the following weights $\mathbf{d}$, classification labels $\mathbf{c}$ and α for each weak classifier:

$$\mathbf{D} = \begin{bmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \\ \mathbf{d}_3 \end{bmatrix} = \begin{bmatrix} 1/8 & 1/8 & 1/8 & 1/8 & 1/8 & 1/8 & 1/8 & 1/8 \\ 1/14 & 1/2 & 1/14 & 1/14 & 1/14 & 1/14 & 1/14 & 1/14 \\ 1/24 & 7/24 & 1/24 & 1/24 & 1/4 & 1/24 & 1/4 & 1/24 \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 & -1 & -1 & -1 & -1 \end{bmatrix}$$

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} 0.97 \\ 0.90 \end{bmatrix}$$

a) Perform the third AdaBoost iteration on the data $\mathbf{X}$ using the labels $\mathbf{y}$ and 'decision stumps' as weak classifiers. Calculate $\mathbf{d}_4$, $\mathbf{c}_3$ and α_3 (3p)

b) Classify the data $\mathbf{X}$ using strong classifiers based on one, two, and three weak classifiers. Write the accuracy in each case. (2p)

Hint: The standard way of updating the weights in the standard AdaBoost method is $d_{t+1}(i) \propto d_t(i) e^{-\alpha_t y_i h_t(\mathbf{x})}$, where $\alpha_t = \frac{1}{2} \ln \frac{1-\epsilon_t}{\epsilon_t}$.

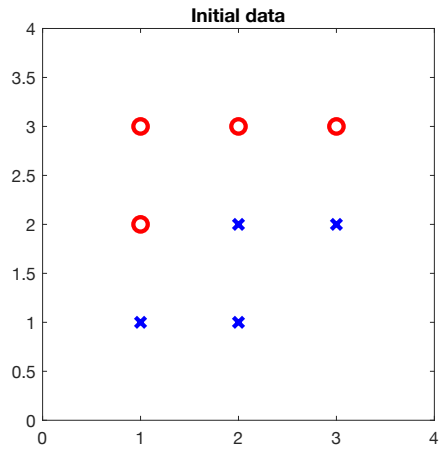


Figure 1: Blue crosses and red circles represent classes 1 and -1 respectively.

AID:	Exam Date: 2021-06-09
Course Code: TBMI26 / 732A55	Exam Code: TEN1

3. Assume we have a signal

$$\mathbf{x} = \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}$$

as given in the table below.

t	1	2	3	4	5
$x_1(t)$	-2	0	1	2	4
$x_2(t)$	-1	3	2	1	5

a) Show how the data volume can be reduced to half the size by using principal component analysis. (3p)

b) Reconstruct the original data as closely as possible from the reduced data. (1p)

c) Motivate whether or not PCA is be a suitable compression method for this dataset. (1p)

Hint: The eigenvalues of a 2×2-matrix $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ are:

$$\lambda_1 = \frac{T}{2} + \sqrt{\frac{T^2}{4} - D} \text{ and } \lambda_2 = \frac{T}{2} - \sqrt{\frac{T^2}{4} - D},$$

with trace $T = a + d$ and determinant $D = ad - bc$. The corresponding eigenvectors are

$$\mathbf{e}_1 = \begin{pmatrix} \lambda_1 - d \\ c \end{pmatrix} \text{ and } \mathbf{e}_2 = \begin{pmatrix} \lambda_2 - d \\ c \end{pmatrix}$$

AID:	Exam Date: 2021-06-09
Course Code: TBMI26 / 732A55	Exam Code: TEN1

4. The figure below shows a state model of a system were the task is to get from state **1** to the end state **5** with maximal reward over time. When moving from state **2** to state **5**, there is a probability p of obtaining a reward of 3, and a probability $1-p$ of obtaining a reward of 1 ($0 \leq p \leq 1$).

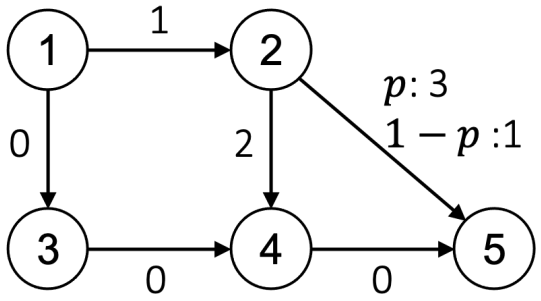


Figure 2: The state model. The numbers next to the arrows are the rewards for moving from one state to another.

a) Calculate the optimal Q- and V- functions for the system as a function of γ ($0 < \gamma \leq 1$) and p ($0 \leq p \leq 1$). (3p)

b) Calculate the Q-function after 5 steps of Q-learning. Start with the Q - function $Q(x, a) = 0$ for every state and action. Choose $\gamma = 1$ and $p = 0.1$, and apply q-learning using the following actions: 2→5, 1→2, 2→4, 3→4 and 2→4. Please observe that it is possible to perform Q-learning by making legal actions in a non-realistic sequence. (2p)