Exam Pattern Recognition
Wednesday, February 1, 2023
17.00-19.30 hours

General Instructions

1. Write your name and student number on every sheet.
2. You are allowed to use a (graphical) calculator.
3. You are allowed to consult 1 A4 sheet of paper with notes (written or printed) on both sides.
4. Always show how you arrived at the result of your calculations. Otherwise you cannot get partial credit for incorrect final answers.
5. There are five questions for which you can earn 100 points.
6. Questions 1,2, and 3 are graded by Ad Feelders, and questions 4 and 5 by Albert Gatt. Since we would like to grade in parallel, please write your answers to Ad's and Albert's questions on separate sheets.

Question 1: Linear Regression (10 points)

According to Moore's law, the transistor count of integrated circuits doubles every two years. Table 1 displays a list of Intel CPUs, their transistor count (t), and year of introduction (x).

n	Processor	Transistors (t)	Year (x)
1	Intel 4004	2,300	1971
2	Intel 8008	3,500	1972
3	Intel 8080	4,500	1974
4	Intel 8085	6,500	1976
5	Intel 8086	29,000	1978
6	Intel 8088	29,000	1979
7	Intel 80186	55,000	1982
8	Intel 80286	134,000	1982
9	Intel 80386	275,000	1985
10	Intel i960	250,000	1988
11	Intel 80486	1,180,235	1989
12	Pentium	3,100,000	1993
13	Pentium Pro	5,500,000	1995

Table 1: Intel CPUs, transistor count (t), and year of introduction (x).

In order to test Moore's law, we define the transformed target variable $z_n = \log_2(t_n)$, where $\log_2(\cdot)$ denotes the logarithm with base 2. We specify the linear regression model

$$z_n = a + bx_n + \varepsilon_n,$$

where a and b are unknown coefficients to be estimated from the data. We estimate this model with the method of least squares using the first 12 observations from table 1. This gives the following result:

```
Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)           -925.49861   51.76781  -17.88 6.41e-09
x                        0.47515    0.02614   18.18 5.44e-09
---
Multiple R-squared:  0.9706
```

(a) (2 pts) What would be the value of b if Moore's law is correct?

(b) (4 pts) What transistor count (t) would the regression model predict for the processor that was introduced in 1995? Round the final answer to the nearest integer.

(c) (4 pts) What percentage of the total variation in log transistor count (z) is explained by the regression model?

Question 2: Logistic Regression (20 points)

To determine whether a text was generated by Chat GPT, we consider two features, called *complexity* and *variation*. Both features are computed from the raw text, and take values between 0 and 1. We expect that if the text has high complexity then it's more likely to be human-written rather than AI-generated. Likewise, we expect that humans tend to write with greater variation, whereas AI-generated texts tend to be more uniform. The precise definition of the features, or how they are computed from the text, is not relevant to this question, and will therefore not be discussed.

We train a logistic regression model on Chat GPT-generated texts ($t = 1$), and human-written texts ($t = 0$). The parameters of the model are estimated with the method of maximum likelihood, giving the following results:

```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)    4.363      1.393   3.131  0.00174
complexity    -5.310      1.686  -3.150  0.00163
variation     -3.056      1.292  -2.365  0.01801
```

(a) (5 pts) Are the coefficient estimates consistent with our expectations? Explain why or why not.

(b) (5 pts) Use the fitted model to estimate the probability that a text with *complexity = 0.5* and *variation = 0.5* was generated by Chat GPT.

(c) (5 pts) Use the fitted model to construct a linear classification rule, assuming we always predict the class with the highest probability according to the fitted model.

(d) (5 pts) Are the coefficients of *complexity* and *variation* significant at significance level $\alpha = 0.01$? Explain how you determined the answer.

Question 3: Support Vector Machines (20 points)

We receive the following output from the optimization software for fitting a support vector machine with linear kernel and perfect separation of the training data ("hard margin"):

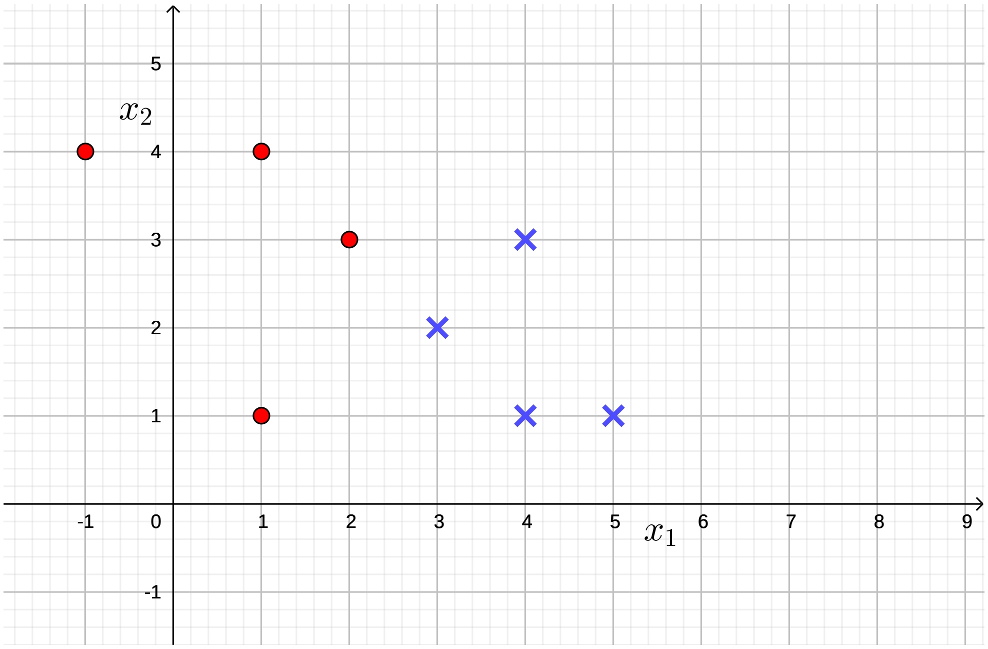
n	$x_{n,1}$	$x_{n,2}$	t_n	a_n
1	1	1	$+1$	$\frac{2}{9}$
2	1	4	$+1$	0
3	-1	4	$+1$	0
4	2	3	$+1$	$\frac{8}{9}$
5	3	2	-1	$\frac{10}{9}$
6	4	1	-1	0
7	4	3	-1	0
8	5	1	-1	0

Here $x_{n,1}$ denotes the value of x_1 for the n-th observation, t_n denotes the class label of the n-th observation, etc. You are given the following formulas:

$$b = t_s - \sum_{n=1}^{N} a_n t_n \mathbf{x}_s^\top \mathbf{x}_n \qquad \text{(for any support vector } \mathbf{x}_s \text{ with label } t_s\text{)}$$

$$\mathbf{w} = \sum_{n=1}^{N} a_n t_n \mathbf{x}_n$$

The dataset is plotted in the figure below. Red circles are examples of class $+1$, blue crosses are examples of class -1.



Answer the following questions:

(a) (4 pts) List the support vectors.

(b) (8 pts) Give the equation of the maximum margin linear decision boundary.

Next we apply a soft margin SVM with $C = 1$ and linear kernel to the same data set. We receive the following output from the optimization software: $a_1 = 0.2$, $a_4 = 0.8$, $a_5 = 1$, and the remaining Lagrange multipliers are all zero. The corresponding optimal linear decision boundary is given by:

$$1.6 - 1.2x_1 + 0.6x_2 = 0$$

(c) (8 pts) Give the value of the soft margin objective function $C \sum_{n=1}^{N} \xi_n + \frac{1}{2}\|\mathbf{w}\|^2$ for the solution given above.

PLEASE WRITE YOUR ANSWERS TO QUESTIONS 4 AND 5
ON A SEPARATE SHEET!

Question 4: Feedforward and Convolutional Networks (24 points)

(a) (6 pts) Consider the feedforward network in Figure 1. This network is designed for a classification task. What is the loss function we would typically use to train such a network, and why is it appropriate to use this function, rather than a squared error loss? [You may define the function mathematically, or explain it in plain English.]

(b) (6 pts) During backpropagation, we normally apply a learning rate to the gradients, before we update model parameters. Explain why this is necessary. What are the risks of setting the learning rate too high, or too low?

(c) (6 pts) Suppose you wanted to classify images. List at least two disadvantages of using a feedforward network such as the one in Figure 1. List at least two advantages of using convolutional layers instead.

(d) (6 pts) On the left side of the figure below is a hypothetical image, represented as a 4×4 matrix of pixel values (assume we have a single channel). Beside it is a 2×2 convolution filter. Write down the matrix that results from applying the convolution filter to the image, with a stride of 2.

$$\begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{bmatrix} \begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix}$$

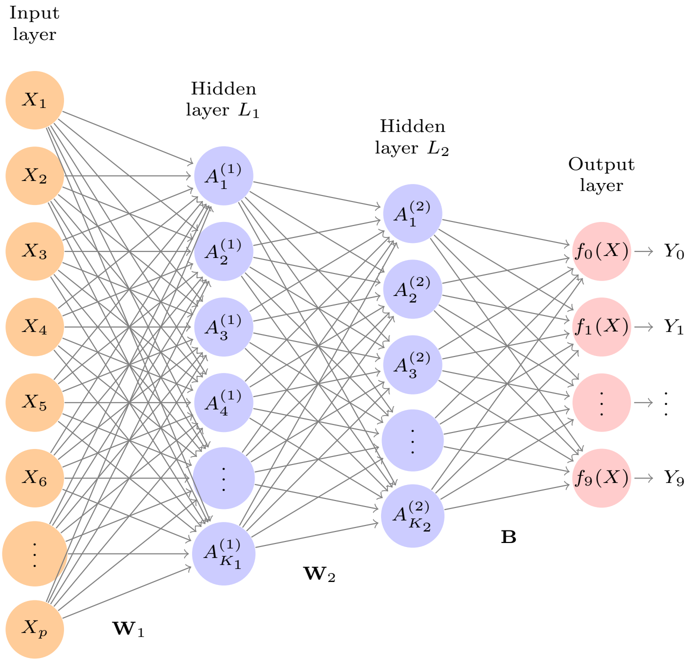


Figure 1: Feedforward network

Question 5: Recurrent neural networks (26 points)

(a) (7 pts) Why do standard (i.e. non-gated) RNNs suffer from the vanishing gradient problem?

(b) (12 pts) In a standard RNN-based encoder-decoder architecture, the hidden state of the decoder at timestep t is computed as:

$$h_t^d = g(\hat{y}_{t-1}, h_{t-1}^d, c),$$

where $g()$ is a nonlinear activation function, $\hat{y}_{t-1}$ is the output predicted at the previous timestep, h_{t-1}^d is the previous decoder hidden state, and c is the context.

Explain what the context c is. Explain further how the definition of the context changes when, instead of the standard formulation as given above, we use attention in the computation of the decoder hidden state. [You may give this explanation using formal notation, or in plain English.]

(c) (7 pts) In a Transformer architecture, a single transformer block usually consists of multiple self-attention heads, followed by normalization and feedforward layers. What is the role of a self-attention head, and why is it a good idea to have more than one in a given block?