Durham University

EXAMINATION PAPER

Examination Session: May/June	**Year:** 2025	**Exam Code:** MATH3431-WE01-SP

Title: Machine Learning & Neural Networks III (2023/24)

Time:	2 hours	
Additional Material provided:		
Materials Permitted:		
Calculators Permitted:	Yes	Models Permitted: Casio FX83 series or FX85 series.

Instructions to Candidates:	Answer all questions. Section A is worth 40% and Section B is worth 60%. Within each section, all questions carry equal marks. Write your answer in the white-covered answer booklet with barcodes. Begin your answer to each question on a new page.

Revision:	

ED01/2025

SECTION A

Q1 (a) A study was conducted in Nebraska from 1927 to 1928 to see if a simple method could be developed that farmers could use to estimate the volume of roughly hemispherical haystacks. The variables are the circumference (C) in feet of a haystack base, measured using a rope; and the haystack's volume (V) in cubic feet, measured using survey instruments that were not normally available to farmers in the 1920s. Overall 120 samples were taken but only the first four are displayed below:

C	V
69.0	2835
65.0	2702
73.0	3099
62.5	1306
⋮	⋮

R was used to fit a linear model

$$\hat{V} = \beta_1 + \beta_2 C^3$$

to estimate the volume of a haystack based on the cube of its base circumference. The extract of the **R output shown on the next page** will be useful in answering some of the following questions.

i) The matrix X in the matrix-multiplication representation of a regression model $y = X\beta + \epsilon$ is known as the design matrix. Write down the first two rows of the design matrix of this regression model.

ii) What are the estimated regression coefficients of the model?

iii) Calculate the predicted volume $\hat{V}$ and the residual for the second haystack shown above, according to the regression model.

iv) If the haystacks were perfect hemispheres, their circumference would have value $2\pi r$, and they would have volume $\frac{1}{2}\frac{4\pi}{3}r^3$, suggesting a theoretical value of $\beta_2 = \frac{1}{12\pi^2}$. On the other hand, a perfectly *cylindrical* haystack with height r would have volume πr^3, suggesting a theoretical value of $\beta_2 = \frac{1}{8\pi^2}$. Since a hypothetical hemispherical or cylindrical haystack of circumference zero would also have zero volume, the value of β_1 in either case would just be zero. Treat the set of four observations above as a test data set, and determine the best model of the three (the model fitted in R, hemispherical, or cylindrical) based on your calculation of the mean standard error.

[**Question 1 continues on the next page**]

```
> Hmodel <- lm(V ~ I(C^3), data = haystack)
> summary(Hmodel)

Call:
lm(formula = V ~ I(C^3), data = haystack)

Residuals:
     Min       1Q   Median       3Q      Max
-1508.87  -371.58   -13.57   350.28  2734.55

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 4.850e+001 2.588e+002   0.187    0.852
I(C^3)      8.857e-003 7.530e-004  11.762  <2e-016 ***
---
Signif. codes:  0  ‘***’  0.001  ‘**’  0.01  ‘*’  0.05  ‘.’  0.1  ‘ ’  1

Residual standard error: 623.2 on 118 degrees of freedom
Multiple R-Squared: 0.5397,     Adjusted R-squared: 0.5358
F-statistic: 138.3 on 1 and 118 degrees of freedom,     p-value: 0
```

(b) Recall that Least Squares (LS) produces estimates of coefficients $\beta_0, \beta_1, \beta_2, \ldots, \beta_p$ for a linear regression problem by minimising

$$RSS = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 .$$

An Elastic Net Regression minimises the cost function

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda_1 \sum_{j=1}^{p} \beta_j^2 + \lambda_2 \sum_{j=1}^{p} |\beta_j|,$$

i) Lasso and ridge regression are special cases of Elastic Net Regression. Identify the conditions under which Elastic Net Regression is equivalent to each of these other forms of regularised regression.

ii) Suppose you have a data set with a large number of feature variables, and you believe that only a few of them have an important effect on the output variable, but you are not sure which ones. Would it be better to use lasso or ridge regression? Support your answer with a schematic diagram showing a two-dimensional representation of lasso and ridge regression.

Q2 Consider the regression problem, with a predictive rule $h_w : \mathbb{R}^d \to (0,1)$ which receives inputs $x = (x_1, ..., x_d)^\top \in \mathbb{R}^d$ and returns values in $(0,1)$. Let $h_w(x)$ be modeled as a feedforward neural network (FNN) with equation

$$h_w(x) = \sigma_2 \left(\sum_{j=1}^{c} w_{2,1,j} \sigma_1 \left(\sum_{i=1}^{d} w_{1,j,i} x_i \right) \right)$$

We consider activation functions

$$\sigma_1(\xi) = \begin{cases} \exp(\xi) - 1 & , \xi \leq 0 \\ 1 & , \xi > 0 \end{cases}$$

and

$$\sigma_2(\xi) = \exp\left(-\xi^2\right)$$

for $\xi \in \mathbb{R}$. The parameters $c, d \in \mathbb{N}_+$ are known while the weights $\{w_{\cdot,\cdot,\cdot}\}$ of the NN are unknown. To learn the unknown weights $\{w_{\cdot,\cdot,\cdot}\}$, we specify the loss function

$$\ell(w, z = (x,y)) = -\log(1 + h_w(x) - y) + \log(1 - h_w(x) + y)$$

where $z = (x,y)$ denotes an example, $x \in \mathbb{R}^d$ is the input vector (features), and $y \in \mathbb{R}$ is the output value (target).

(a) Describe the algorithm necessary to perform the forward pass of the backpropagation procedure to compute the activations (which may be denoted as $\{\alpha_{t,i}\}$) and outputs (which may be denoted as $\{o_{t,i}\}$) at each layer t.

(b) Describe the algorithm necessary to perform the backward pass of the backpropagation procedure in order to compute the gradient

$$\nabla_w \ell(w, (x,y)) = \left(\left(\frac{\partial}{\partial w_{1,j,i}} \ell(w, (x,y)) \right)_{j=1,i=1}^{c,d}, \left(\frac{\partial}{\partial w_{2,1,j}} \ell(w, (x,y)) \right)_{j=1}^{c} \right)$$

of the loss function $\ell(w,z)$ with respect to w for any example $z = (x,y)$. Clearly state the steps of the procedure as well as state the quantities

$$\frac{\partial}{\partial w_{1,j,i}} \ell(w, (x,y)), \text{ and } \frac{\partial}{\partial w_{2,1,j}} \ell(w, (x,y))$$

for all $j = 1, ..., c$, and $i = 1, ..., d$.

SECTION B

Q3 (a) A principal component analysis is carried out using the correlation matrix Σ of a data set with $n = 25$ observations and $p = 4$ variables. The ordered eigenvalues of Σ are given by

$$\lambda_1 = 2.43, \lambda_2 = 0.92, \lambda_3 = 0.41, \text{and } \lambda_4 = ??$$

with the associated eigenvectors

$$\gamma_1 = \begin{pmatrix} 0.61 \\ 0.55 \\ 0.27 \\ 0.51 \end{pmatrix}, \gamma_2 = \begin{pmatrix} -0.01 \\ -0.01 \\ 0.90 \\ -0.44 \end{pmatrix}, \gamma_3 = \begin{pmatrix} 0.10 \\ ?? \\ 0.27 \\ 0.57 \end{pmatrix}, \gamma_4 = \begin{pmatrix} -0.79 \\ ?? \\ ?? \\ 0.46 \end{pmatrix}.$$

i) Compute the missing eigenvalue λ_4.

ii) Compute the three missing entries of the eigenvectors.

iii) How many components would you need in order to capture at least 90% of the total variance of the data cloud? Give an equivalent answer for 95% of the total variance.

iv) Explain how you would compute the matrix Σ from the given information. (Provide an explicit formula, but no numerical calculations.)

(b) i) In logistic regression, the expected value of the binary output variable Y can be written as $\mathbb{E}(Y|X = x) = p(Y = 1|X = x)$ for a given predictor state $X = x$. Given the logit transformation

$$\ln\left(\frac{p(X)}{1 - p(X)}\right) = \beta_0 + \beta_1 X$$

derive an expression for the logistic function that gives the probability $p(X) = p(Y = 1|X = x)$ of a "true" outcome state $(Y = 1)$.

ii) Write the generalised form of this logistic function for an n-dimensional set of feature variables $X = \boldsymbol{x}$.

iii) State the upper and lower limits of the output of the logistic function, and explain why we make use of this transformation in logistic regression.

Q4 (a) Consider a learning problem $(\mathcal{H}, \mathcal{Z}, \ell)$. Assume that the loss function $\ell(w, z)$ is convex, β-smooth, and nonnegative with $z \in \mathcal{Z}$ and $w \in \mathcal{H}$ where $\|w\| \leq B$. If we run the Stochastic Gradient Descent (SGD) algorithm with constant learning rate η, for total number of iterations T, and with purpose to minimize the risk function $R_g(w)$, then we have that for every $w^* \in \mathcal{H}$

$$\mathrm{E}\left(R_g\left(w_{\mathrm{SGD}}\right)\right) \leq \frac{1}{1-\eta\beta}\left(R_g\left(w^*\right)+\frac{\left\|w^*\right\|^2}{2\eta T}\right)$$

where w_{SGD} is the output of the SGD. In addition assume that $\ell(0, z) \leq 1$ for all z. Show that by running online SGD with $\eta = \frac{1}{\beta(1+3/\epsilon)}$ for $T \geq 12B^2\beta/\epsilon$ iterations, where $\epsilon > 0$, we obtain agnostic Probably Approximately Correct (PAC)-like guarantees.

(b) Consider the regression problem with inputs $x \in \mathcal{X} \equiv \mathbb{R}^2$, target $y \in \mathcal{Y} \equiv \mathbb{R}$, and prediction rule $h_w : \mathbb{R}^2 \to \mathbb{R}$ with $h_w(x) = w^\top x = w_1 x_1 + w_2 x_2$, $w = (w_1, w_2)^\top$. Consider a loss function $\ell : \mathbb{R}^2 \to \mathbb{R}_+$ with

$$\ell(w, z = (x, y)) = \left\|y - w^\top x\right\|_2^2 + \alpha \left\|w_2\right\|_1 + (1-\alpha)\left\|w_2\right\|_2^2 \qquad (1)$$

for some given value $\alpha \in (0, 1)$. Assume there is available a training data set $S_m = \{z_i = (x_i, y_i)\,;\, i = 1, ..., m\}$ of size m. Let g denote the real data generating process. Write down the algorithm of the Stochastic Gradient Descent (SGD) with constant learning rate $\eta > 0$, batch sample size equal to 1, and termination criterion $t > T$ for some $T > 0$, that aims to compute w^*, where

$$w^* = \arg\min_w \left(\mathrm{E}_{z\sim g}\left(\ell\left(w, z = (x, y)\right)\right)\right) \qquad (2)$$

The formulas in your algorithm should be specific to the loss function in (1).

(c) Let X be an instance set and let ψ be a feature mapping of $\mathcal{X}$ into some Hilbert feature space $\mathcal{V}$. Let $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a kernel function that implements inner products in the feature space $\mathcal{V}$.

Consider the binary classification algorithm that predicts the label of an unseen instance according to the class with the closest average. Formally, given a training sequence $\mathcal{S} = \{(x_1, y_1), ..., (x_m, y_m)\}$, for every $y \in \mathcal{Y} = \{-1, +1\}$ we define

$$c_y = \frac{1}{m_y} \sum_{i:y_i=y} \psi(x_i)$$

where $m_y = |\{i : y_i = y\}|$. We assume that m_{+1} and m_{-1} are nonzero. Then, the algorithm outputs the value of the following decision rule:

$$h(x) = \begin{cases} 1 & , \|\psi(x) - c_{+1}\|_2 \leq \|\psi(x) - c_{-1}\|_2 \\ -1 & , \text{otherwise.} \end{cases}$$

(i) Let $w = c_{+1} - c_{-1}$ and let $b = \frac{1}{2}\left(\|c_{-1}\|_2^2 - \|c_{+1}\|_2^2\right)$. Show that

$$h(x) = \mathrm{sign}\left(\langle w, \psi(x)\rangle + b\right)$$

(ii) Express $h(x)$ in terms of the kernel function, and without accessing individual entries of $\psi(x)$ or w.