Durham University

# EXAMINATION PAPER

| | | |
|---|---|---|
| **Examination Session:**<br>May/June | **Year:**<br>2024 | **Exam Code:**<br>MATH31520-WE01 |

| |
|---|
| **Title:**<br>Machine Learning and Neural Networks V |

| | | |
|---|---|---|
| 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.<br>Section A is worth 40% and Section B is worth 60%. Within each section, all questions carry equal marks.<br>Students must use the mathematics specific answer book. |

| | |
|---|---|
| **Revision:** | |

## SECTION A

**Q1** (a) Explain the difference between supervised and unsupervised machine learning. Give two examples of supervised machine learning techniques and two examples of unsupervised machine learning techniques.

(b) Ridge and lasso regression each correspond to a constrained optimisation of least squares, with a cost function that includes a tuning parameter $\lambda \geq 0$:

$$\sum_{i=1}^{n}\big(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\big)^2 + \lambda \sum_{j=1}^{p} f(\beta_j)$$

(i) Specify the form of $f(\beta_j)$ for ridge **and** for lasso regression. You may refer to Figure 1 to ensure you have assigned the correct formula to each regression type.

(ii) Explain the behaviour of each form of constrained optimisation when

A. $\lambda \to \infty$

B. $\lambda \to 0$

(c) Why is it important to normalise your feature variables $X_j$ when using this type of constrained optimisation?

(d) You are attempting to fit a model to a large data set with many feature variables, where you believe the vast majority of the feature variables $X_j$ each have an independent effect on the output variable $Y$. Should you use ridge or lasso regression? Explain your answer.

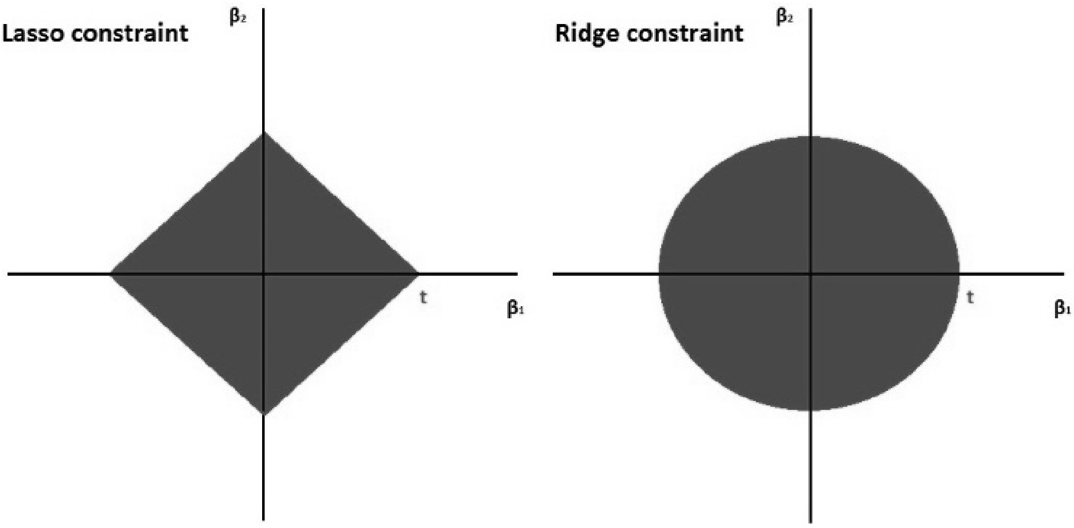


**Figure 1:** Two-dimensional schematic representation of the constraints applied during ridge and lasso regression.

**Q2** Consider the regression problem, with a predictive rule $h : \mathbb{R}^d \rightarrow \mathbb{R}^q$ which receives inputs $x = (x_1, ..., x_d)^\top \in \mathbb{R}^d$ and returns values in $\mathbb{R}^q$. Let $h(x)$ be modeled as a feedforward neural network (FNN) with equation $h(x) = (h_1(x), ..., h_q(x))^\top$ and

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

for $k = 1, ..., q$. We consider activation functions $\sigma_1(\xi) = \frac{\xi}{1+\exp(-3\xi)}$ and $\sigma_2(\xi) = \frac{\xi+\sqrt{\xi^2+4}}{2}$ for $\xi \in \mathbb{R}$. The parameters $c, d, q \in \mathbb{N}_+$ are known while the weights $\{w_{\cdot,\cdot,\cdot}\}$ of the FNN are unknown. To learn the unknown weights $\{w_{\cdot,\cdot,\cdot}\}$, we specify the loss function

$$\ell(w, z = (x, y)) = \sum_{k=1}^{q} (h_k(x) - y_k - 1) - \sum_{k=1}^{q} \exp(y_k - h_k(x) + 1)$$

where $z = (x, y)$ denotes an example, $x \in \mathbb{R}^d$ is the input vector (features), and $y = (y_1, ..., y_q)^\top \in \mathbb{R}^q$ is the output vector (targets).

(a) Describe the algorithm necessary to perform the forward pass of the back-propagation 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 back-propagation procedure in order to compute the gradient

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

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

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

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

## SECTION B

**Q3** **3.1** Piecewise polynomial regression defines $K$ least-squares fits between pairs of interior knots $[\xi_{k-1}, \xi_k]$ for $k = 1, \ldots, K$ (with implicit knots at $\xi_0 = -\infty$ and $\xi_{K+1} = +\infty$). For a polynomial of degree $d$, the equation of a piecewise polynomial regression model is

$$y = \sum_{j=0}^{d} \beta_{jk} x_k^j, \quad \text{if } \xi_{k-1} < x_k < \xi_k$$

To define *spline regression*, constraints are applied to the piecewise polynomial model at each knot to ensure the spline model is well-behaved.

(a) State the three constraints that are applied to create cubic splines (using either words or equations).

(b) Would applying a fourth constraint based on $y_k'''$ improve a cubic spline model? Why/why not?

(c) A degree $d$ spline with knots at $\xi_k$ for $k = 1, \ldots, K$ can be represented by *truncated power functions*, denoted by $b_i$ for $i = 1, \ldots, K + d$, so that

$$y = \beta_0 + \beta_1 b_1(x) + \ldots + \beta_{K+d} b_{K+d}(x) + \epsilon$$

where $\epsilon$ is the residual, and the functions $b_i(x)$ are defined as:

$$b_1(x) = x^1$$

$$\vdots$$

$$b_d(x) = x^d$$

$$b_{k+d}(x) = (x - \xi_k)_+^d, \quad k = 1, \ldots, K$$

Define $(x - \xi_k)_+^d$. Include a schematic diagram for $d = 1$ in your answer.

**3.2** A spline model is being used to predict height (in centimetres) as a function of age (in **months**, between 2 and 20 years) for a sample of 5,000 observations.

(a) Explain why splines are a better technique to use than simple linear regression when modelling this data set.

(b) The locations of knots are important hyperparameters that can have a major impact on the the quality of a spline-based model. Two models are being compared: one with knots at the 25th, 50th, and 75th percentiles of age, and one with a single interior knot at age 14.

(i) Which model do you expect to have lower training error? Which model do you expect to have lower test error? Why?

(ii) What is the risk associated with the model with lower training error?

(c) R uses *B-Splines* in its calculations. In this representation, as well as the interior knots $\xi_k$, the endpoints of the feature data are viewed as exterior knots $\xi_{min}$ and $\xi_{max}$. For a linear spline with a single interior knot at $x = \xi$, this representation reduces to:

$$y = \beta_0 + \beta_1 b_1 + \beta_2 b_2$$

$$b_1 = \begin{cases} \frac{\xi_{max} - x}{\xi_{max} - \xi} & \text{if } x > \xi \\ \frac{x - \xi_{min}}{\xi - \xi_{min}} & \text{otherwise} \end{cases}$$

$$b_2 = \begin{cases} \frac{x - \xi}{\xi_{max} - \xi} & \text{if } x > \xi \\ 0 & \text{otherwise} \end{cases}$$

Figure 2 shows part of the summaries from RStudio for two different spline models.

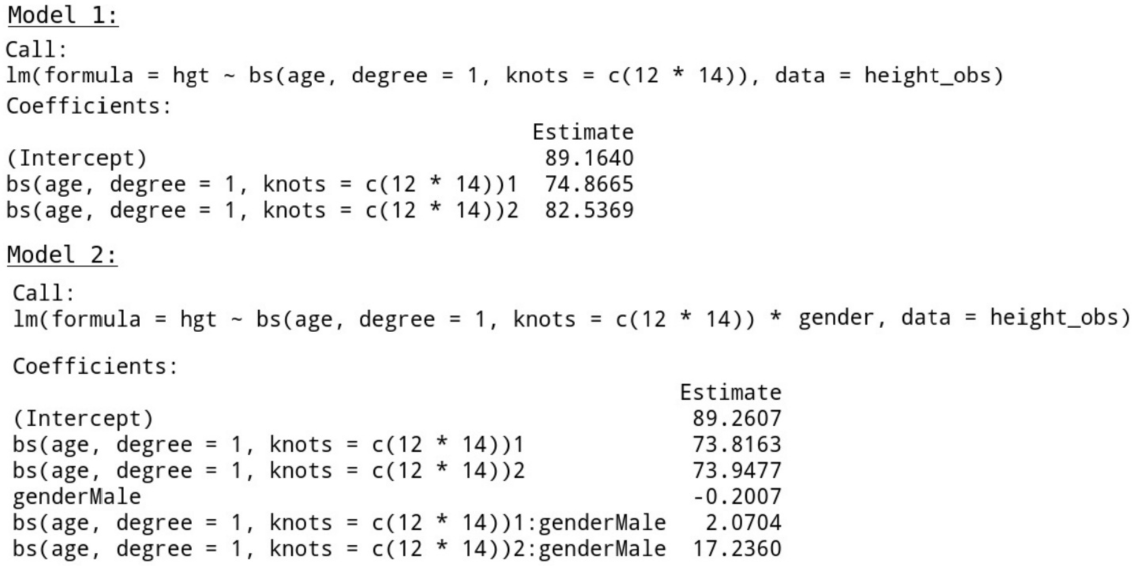
(i) State the estimated function of Model 1, including coefficient values accurate to two decimal places. In this data set, $\xi_{min} = 24.5$ and $\xi_{max} = 239.5$.
You may use the model output in Table 1 to **sanity check** your function. Note: you should not expect to replicate that level of precision when using the approximate form of the function asked for in this question.

(ii) Which model would be most effective at:

A. explaining the connection between age and height?

B. extrapolating the height of a 25-year-old woman?

(iii) Estimate the height of a 25-year-old woman based on your choice of most effective model accurate to the nearest centimetre. Do you have any concerns about the reliability of this prediction?



```
Model 1:
Call:
lm(formula = hgt ~ bs(age, degree = 1, knots = c(12 * 14)), data = height_obs)
Coefficients:
                                       Estimate
(Intercept)                             89.1640
bs(age, degree = 1, knots = c(12 * 14))1  74.8665
bs(age, degree = 1, knots = c(12 * 14))2  82.5369

Model 2:
Call:
lm(formula = hgt ~ bs(age, degree = 1, knots = c(12 * 14)) * gender, data = height_obs)

Coefficients:
                                                  Estimate
(Intercept)                                        89.2607
bs(age, degree = 1, knots = c(12 * 14))1           73.8163
bs(age, degree = 1, knots = c(12 * 14))2           73.9477
genderMale                                         -0.2007
bs(age, degree = 1, knots = c(12 * 14))1:genderMale   2.0704
bs(age, degree = 1, knots = c(12 * 14))2:genderMale  17.2360
```

**Figure 2:** List of coefficients for two different spline-based models representing the data set described in Question **3.2**.

| Model | Age: 8 years | Age: 14 years | Age: 17 years |
|---|---|---|---|
| Model 1 | 126.47 | 164.03 | 167.89 |
| Model 2 (Male) | 126.87 | 164.95 | 172.65 |
| Model 2 (Female) | 126.04 | 163.08 | 163.14 |

**Table 1:** Predicted height in centimetres (to two decimal places) for three ages, based on the two spline models described in Question **3.2**.

**Q4** Consider a learning problem $(\mathcal{H}, \mathcal{Z}, \ell)$ with $\mathcal{H} \subset \mathbb{R}^d$, $d > 0$, and loss function $\ell : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}_+$ which is convex, $\beta$-smooth and non-negative. Let $\mathfrak{A}$ be a learning algorithm with output $\mathfrak{A}(\mathcal{S})$ trained against training dataset $\mathcal{S} = \{z_1, ..., z_m\}$ of IID samples $z_1, ..., z_m \sim g$ where $g$ is a data generating distribution. In particular, consider that $\mathfrak{A}(\mathcal{S})$ is the Regularized Loss Minimization learning rule that outputs a hypothesis in

$$\min_{w} \left\{ \hat{R}_{\mathcal{S}}(w) + \lambda \|w\|_2^2 \right\}$$

for $\lambda \geq \frac{2\beta}{m}$ where $\hat{R}_{\mathcal{S}}(w) = \frac{1}{m} \sum_{i=1}^{m} \ell(w, z_i)$ for all $w \in \mathcal{H}$.

(a) Prove that

$$\mathrm{E}_{\mathcal{S}\sim g}\left(\hat{R}_{\mathcal{S}}(\mathfrak{A}(\mathcal{S}))\right) \leq R_g(w) + \lambda \|w\|_2^2$$

for all $w \in \mathcal{H}$. $R_g(\cdot)$ denotes the risk function under the real data generating distribution $g$.

(b) Prove that

$$\mathrm{E}_{\mathcal{S}\sim g}\left(R_g(\mathfrak{A}(\mathcal{S})) - \hat{R}_{\mathcal{S}}(\mathfrak{A}(\mathcal{S}))\right) \leq \frac{48\beta}{\lambda m} \mathrm{E}_{\mathcal{S}\sim g}\left(\hat{R}_{\mathcal{S}}(\mathfrak{A}(\mathcal{S}))\right).$$

**Hint:** If needed you can use the following:

Let $\mathcal{S}^{(i)} = \{z_1, ..., z_{i-1}, z', z_{i+1}, ..., z_m\}$ be a set resulting from $\mathcal{S}$ by replacing its $i$-th element $z_i$ with an independently drawn $z' \sim g$. Then

$$24\beta\ell(\mathfrak{A}(\mathcal{S}), z_i) + \lambda m \ell(\mathfrak{A}(\mathcal{S}), z_i) + 24\beta\ell\left(\mathfrak{A}\left(\mathcal{S}^{(i)}\right), z'\right) - \lambda m \ell\left(\mathfrak{A}\left(\mathcal{S}^{(i)}\right), z_i\right) \geq 0$$

(c) Show that the learning algorithm $\mathfrak{A}$ is on-average-replace-one-stable with rate $\varepsilon$. Specify $\varepsilon$ as a function of $\beta$, $\lambda$, $m$ and possibly other user-specified constants if needed. Explain how the shrinkage parameter $\lambda$, the training dataset size $m$, and the smoothness parameter $\beta$ affect the stability of the learning algorithm $\mathfrak{A}$.

(d) Show that the expected risk is bounded as follows:

$$\mathrm{E}_{\mathcal{S}\sim g}\left(R_g(\mathfrak{A}(\mathcal{S}))\right) \leq \left(1 + \frac{48\beta}{\lambda m}\right)\left(R_g(w) + \lambda \|w\|_2^2\right)$$

for all $w \in \mathcal{H}$.