30.5/50

UNIVERSITY OF EDINBURGH

COLLEGE OF SCIENCE AND ENGINEERING

SCHOOL OF INFORMATICS

INFR10069 AND INFR11182 INTRODUCTORY APPLIED MACHINE LEARNING

Q1 — 7/12

December 2020

Q2 — 6/13

13:00 to 15:00

Q3 — 11.5/14

Q4 — 6/11

INSTRUCTIONS TO CANDIDATES

1. Note that **ALL QUESTIONS ARE COMPULSORY.**
2. **DIFFERENT QUESTIONS MAY HAVE DIFFERENT NUMBERS OF TOTAL MARKS. Take note of this in allocating time to questions.**
3. **This is an OPEN BOOK examination.**

Year 3 Courses

Convener: D.Armstrong
External Examiners: S. Rogers, H.Vandierendonck

MSc Courses

Convener: A.Pieris
External Examiners: W. Knottenbelt, M. Dunlop, E. Vasilaki

THIS EXAMINATION WILL BE MARKED ANONYMOUSLY

1. **Neural Networks and Decision Trees.** You have begun a new job as a machine learning engineer. Your colleague hands you a trained neural network that is being used in the company for a specific binary classification problem. The neural network takes a two dimensional attribute vector $\mathbf{x} \in \mathbb{R}^2$ as input and predicts a single continuous output $y \in [0, 1]$. This output can be interpreted as the probability that the input instance is the positive class i.e. $P(y = 1|\mathbf{x})$.

The network performs the following operations:

$$\begin{aligned} h_1 &= \tanh(x_1 w_{11} + x_2 w_{12} + b_1) \\ h_2 &= \tanh(x_1 w_{21} + x_2 w_{22} + b_2) \\ y &= \sigma(h_1 v_1 + h_2 v_2 + b_3), \end{aligned}$$

where $\sigma(a) = \frac{1}{1+e^{-a}}$ and $\tanh(a) = \frac{e^a - e^{-a}}{e^a + e^{-a}}$.

The network you receive has already been trained and has the following weights and biases:

$$\begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix} = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}, \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} -100 \\ -100 \end{bmatrix}, \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 100 \\ 100 \end{bmatrix}, b_3 = -100.$$

(a) Sketch the neural network depicted above. You can use circles for hidden units, and arrows for model parameters. Label each part of your drawing. [*2 marks*]

(b) This neural network uses two different types of non-linear activation functions. Why has the network designer not used a tanh function for the output of the network? [*1 mark*]

(c) Upon closer inspection you realise that the network classifies all inputs where both $x_1 > 1.0$ *and* $x_2 > 1.0$ as positive (i.e. $P(y = 1|\mathbf{x}) = 1$), and all other inputs as negative (i.e. $P(y = 1|\mathbf{x}) = 0$). Create a decision tree that produces the same predictions as the neural network for any input $\mathbf{x}$. Sketch the structure of this tree, taking care to report all the relevant parameters. [*3 marks*]

(d) Your next step is to verify that your decision tree produces the same predictions as the neural network. For each of the following input data points, report the predictions of the decision tree and the neural network: $\mathbf{x}_1 = (2, 2)^T$ and $\mathbf{x}_2 = (-2, 2)^T$. Here, each data point is of the form (x_1, x_2). You only need to return the predicted class label in each case (i.e. $y = 1$ if $P(y = 1|\mathbf{x}) > 0.5$), not the actual probabilities. Show your intermediate steps. [*3 marks*]

(e) You have been asked to change the neural network so that it now classifies all inputs where both $x_1 > 1.0$ *and* $x_2 > 2.0$ as positive (i.e. $y = 1$), and all other inputs as negative (i.e. $y = 0$). However, you must do this by changing only *one* weight or bias of the network. Which weight or bias should be changed and to what value? Justify your answer. [*3 marks*]

2. **Regression** . You have been tasked with modelling how a single viral news story spreads online. Instead of long time scales (i.e. days), your goal is to model how a story spreads in the first few minutes of being shared. Your model takes time in minutes as input and outputs an estimate of the number of people that get shown a particular story on their news feed.

You decide to use the following model: $f(x) = e^{wx}$, where e is the exponential function, and $x, f(x), w \in \mathbb{R}$ are all scalars representing the input time x, the predicted number of people who see the story $f(x)$, and the model weight w, respectively.

You have access to n input-output training instances i.e. $\{(x_1, y_1), ..., (x_n, y_n)\}$. The sum of squared errors for the model is thus $E(w) = \sum_{i=1}^{n}(y_i - f(x_i))^2$.

(a) Describe two advantages of this model over standard linear regression in the context of this particular application. *[2 marks]*

(b) Your next step is to estimate the weight w of the model using a numerical optimisation approach. Define the gradient descent update rule, taking care to explain the meaning of each term. Why is gradient descent a good approach for this model? Note, you do not need to derive an expression for the gradient of the error just yet, instead write the update rule for gradient descent in general terms. *[2 marks]*

(c) Derive an expression for the derivative of the sum of squared errors with respect to w for our model $f(x) = e^{wx}$. Show your working. You can make use of the fact that $\frac{de^u}{dv} = e^u \frac{du}{dv}$. *[3 marks]*

(d) You have collected the following three input-output training instances:

instance	x	y
1	0.5	2.00
2	1.0	3.25
3	2.0	11.00

Assuming an initial value of $w_{t=0} = 1.0$ and a step size of 0.001, manually compute one step of gradient descent using these three training instances and show that the new estimate of $w_{t=1} = 1.11$ at iteration $t = 1$. Show your intermediate steps. You should use the expression for the gradient of the error that you derived above. You may make use of the following values: *[3 marks]*

x	e^x	$e^{1.11x}$
0.5	1.65	1.74
1.0	2.72	3.03
2.0	7.39	9.21

QUESTION CONTINUES ON NEXT PAGE

QUESTION CONTINUED FROM PREVIOUS PAGE

(e) Report the sum of squared errors using the three training instances above for the initial estimate of $w_{t=0} = 1.0$ and for the updated estimate of $w_{t=1} = 1.11$. How does the error change? Is this to be expected? Explain why. *[2 marks]*

(f) Comment on one consideration that should be taken into account when applying gradient descent on very large regression problems with millions of observations and tens of thousands of attributes. *[1 mark]*

3. **Clustering.** Consider the following dataset, consisting of six 2-dimensional instances:

$$\text{a:} \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \text{b:} \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \text{c:} \begin{pmatrix} 4 \\ 0 \end{pmatrix}, \text{d:} \begin{pmatrix} 2 \\ 2.5 \end{pmatrix}, \text{e:} \begin{pmatrix} 4 \\ 2.5 \end{pmatrix}, \text{f:} \begin{pmatrix} 5 \\ 2.5 \end{pmatrix}.$$

The (symmetric) Euclidean distance matrix between all elements is given below:

	a	b	c	d	e	f	
a		1	3	2.7	3.9	4.7	a
b	1		2	2.5	3.2	3.9	b
c	3	2		3.2	2.5	2.7	c
d	2.7	2.5	3.2		2	3	d
e	3.9	3.2	2.5	2		1	e
f	4.7	3.9	2.7	3	1		f
	a	b	c	d	e	f	

? (a) Calculate the mean of the dataset. [3 marks]
Make a sketch plot of the data, showing the 6 datapoints and the mean. On your sketch plot, show the approximate direction of the principal component of the data. Note: you are **not** required to compute the principal component, you should just sketch it.

✓ (b) Run the k-means clustering algorithm to *convergence* on the above dataset. Set the initial centroids to be $\mathbf{m}_1 = [4, 5]^T$ and $\mathbf{m}_2 = [2, -2]^T$. Report the final centroid values and list the instances in each cluster when the algorithm terminates. Show your working. [4 marks]

✓ (c) Run the **single-link** agglomerative clustering algorithm on the dataset above. Show your working, including the distances between clusters when they merge. [3 marks]

✓ (d) Run the **complete-link** agglomerative clustering algorithm on the dataset above. Draw a *dendrogram*. Threshold the dendrogram at Euclidean distance of 2.01 and show the instances in each remaining cluster. Show your working. [4 marks]

4. **Classification.** You are working as a data analyst with a clinician to make a predictor for adult patients developing a certain disease by age 70. For each patient you have 500 binary (0/1) features which indicate the presence/absence of specific genetic variants, plus three demographic variables *weight* (in kilograms), *height* (in centimetres) and age. The age is coded in ranges 10-19, 20-29, ..., 60-69 years. You also have a target label $y \in \{0, 1\}$ for each patient, indicating if they developed the disease or not by age 70. You have data for $n = 1000$ patients in total.

You decide to try logistic regression (with a weight penalty of $\lambda ||\mathbf{w}||^2$) and k-nearest-neighbours as classifiers.

(a) Describe any pre-processing that you would carry out on the dataset, and explain why you would do this. **Do not** apply any dimensionality reduction. *[3 marks]*

(b) Both logistic regression and k-nearest-neighbours have free parameters (λ and k respectively). Describe how you would design your experiements so as to give the clinician an *unbiased* estimate of the performance of each method. *[2 marks]*

(c) A friend suggests that you should try using a deep neural network with two hidden layers, each containing 200 hidden units. Do you think that this is a good idea? Explain your answer. *[2 marks]*

(d) You believe that only a few of the genetic features are relevant to the prediction problem, but you don't know which ones. State whether k-nearest-neighbours or logistic regression would handle this better, and justify your answer. Specify one other classifier studied in IAML which would handle this aspect well, and explain your reasoning. *[2 marks]*

(e) The clinician asks you to investigate the *fairness* of the chosen classifier with respect to a protected attribute A. Explain the concept of *disparate impact*, and describe how you would investigate the performance of the classifier with respect to different values of the attribute A. *[2 marks]*