Module Code
COM00049H

BSc, BEng, MEng and MMath Degrees 2024–25

Open Assessment

Department Computer Science

Module Intelligent Systems: Probabilistic & Deep Learning

Title Open Individual Assessment

Issued 17 March 2025 Midday

Submission due 19 May 2025 Midday

Feeedback and Marks due 23 June 2025

Instructions:

- All data and any other additional files are available at the PADL VLE site in the **Assessment** section. Your submission should be **a single zip file**, which contains:

  1. a single Jupyter notebook `padl.ipynb` (combining code and explanations);
  2. a PDF `padl.pdf` of the state of the same notebook after all of its code has been executed (see below for advice on how to export your notebook to PDF);
  3. the file with your result for Question 3b;
  4. your trained network weights for Questions 4, 5 and 6;
  5. Python scripts that can be imported and used to run your trained networks for Questions 4, 5 and 6.

  In case of any discrepancies between the contents and output of the Jupyter notebook and the PDF, the former will be used for marking. Make sure you do not use archive formats other than zip. Your code should assume all data files are in the same folder as the notebook from which they are accessed.

- All Python code should be in Python3. Your Jupyter notebook must run correctly when run on Google Colab as practised in the practicals.

- Do not hide code blocks or entire tiles in your notebook, as it slows down the marking process.
- Unless otherwise indicated there are no word limits on your answers, but unnecessarily verbose or irrelevant comments will be marked down.
- Inline comments in the code are no substitute for text tiles with complete sentences addressing a point.
- To save your colab notebook in PDF format follow these instructions. First, download your notebook (File -> Download -> Download .ipynb) and save it as `padl.ipynb`. Second, create a new blank colab notebook and upload `padl.ipynb` to the session storage in this notebook. Third copy and paste the following commands into a code block in the new empty notebook and execute it:
  ```
  !sudo apt-get install texlive-xetex
  texlive-fonts-recommended texlive-plain-generic
  !jupyter nbconvert -to pdf /content/padl.ipynb
  ```
  Finally, refresh session storage and then download the PDF file.

All students should submit their answers through the appropriate VLE submission point in the Assessment area of the VLE site by 19 May 2025 Midday. An assessment submitted after this deadline will be marked initially as if it had been handed in on time, but the Board of Examiners will normally apply a lateness penalty.

Your attention is drawn to the section about Academic Misconduct in your Departmental Handbook.

Any queries you may have on this assessment should be posted on the Discussion Board on the Virtual Learning Environment (VLE) page for Probabilistic & Deep Learning (PADL) in the appropriate discussion area. **No questions will be answered after 5 May 2025**.

Note on Academic Integrity

We are treating this online examination as a time-limited open assessment, and you are therefore permitted to refer to written and online materials to aid you in your answers. However, you must ensure that the work you submit is entirely your own, and for the whole time the assessment is live you must not:

- communicate with other students on the topic of this assessment.
- seek advice or contribution from any other third party, including proofreaders, friends, or family members.

We expect, and trust, that all our students will seek to maintain the integrity of the assessment, and of their award, through ensuring that these instructions are strictly followed. Where evidence of academic misconduct is evident this will be addressed in line with the Academic Misconduct Policy and if proven be penalised in line with the appropriate penalty table. Given the nature of these assessments, any collusion identified will normally be treated as cheating/breach of assessment regulations and penalised using the appropriate penalty table (see AM3.3. of the Guide to Assessment).

1 (27 marks) Linear Regression Models

Download the files `PADL-Q11-train.csv`, `PADL-Q12-train.csv` and `PADL-Q13-train.csv` from the Assessment section of the PADL VLE site. Each comma-separated file contains data on a number of variables, $x_1, x_2, \ldots, x_n$, followed by one last variable, *out*. The label for each column is given in the first, header row of the file. Each of the remaining rows represents one data point:

x_1	x_2	$\ldots$	x_n	*out*
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$

In each case, your ultimate goal here is to train and submit a single regression model meeting the requirements, whose performance will be tested by the exam markers on unseen data not available to you.

Throughout this question you need to use scikit-learn – no marks will be given for the use of PyTorch! Regression should always be evaluated in terms of the R^2 value on out-of-sample data. For each model returned as part of your solution, you should also display its coefficients.

For each of the following three parts of the question, include a code tile in your Colab notebook that attempts to test your choice of model on a data set in the file `PADL-Q11-unseen.csv`, `PADL-Q12-unseen.csv` or `PADL-Q13-unseen.csv`, and compute the corresponding R^2. The unseen test files will have exactly the same format as the corresponding training file, including the header, but possibly a different overall number of rows. This means you can use a renamed copy of the training file to debug that part of your code.

(a) [9 marks] Using the data in file `PADL-Q11-train.csv`, submit a single regression model maximising the R^2 value on out-of-sample data. For that purpose, you can use any additional techniques covered in the material.

(b) [9 marks] Using the data in file `PADL-Q12-train.csv` and a regularisation technique of your choice, submit a single regression model with the aim of reducing the absolute values of the model's coefficients at the cost of at most 10% drop in R^2 value when compared to the best possible performance of a generic linear regression (LR) model. For both the generic and regularised LR model report their coefficients and matching R^2 value.

(c) [9 marks] Using the data in file `PADL-Q13-train.csv` and a data preprocessing technique of your choice, submit a single regression model (not using regularisation), and show the relative benefits of your preprocessing step as measured in terms of R^2.

Marking guidance Marks will be allocated for working code, and the quality of results.

2 (12 marks) Principal Component Analysis and Clustering

Download the `PADL-Q2.csv` file from the Assessment section of the PADL VLE site. The file contains rows of numerical data for five variables with real values, $x1$ to $x5$, as named in the first row of the file. The last column of the file, y, contains a class label expressed as an integer:

x_1	x_2	x_3	x_4	x_5	y
...	...	...	...	...	...
...	...	...	...	...	...
...	...	...	...	...	...

(a) [4 marks] Apply k-means clustering with the same number of clusters as those present in the data to columns $x_1, \ldots, x_5$ of the dataset, then apply principal component analysis (PCA) with a number of principal components $p = 2$ to the same data. Use two separate 2D plots with coordinates PC1 and PC2 to visualise all data points, once showing the original class labels, the second time showing the labels assigned by the clustering algorithm. Use either a different colour for each class label or vary the marker ('o', 'x', 'v' etc.) if you prefer. (Best practice advice: in Matplotlib, use a colour-blind friendly colour map, such as the default `viridis`).

(b) [4 marks] Use the result of applying principal component analysis (PCA) with $p = 2$ to columns $x_1, \ldots, x_5$ of the dataset as input to k-means clustering. To clarify, each data point presented to the clustering algorithm will only be represented by two attributes, PC1 and PC2. Again, set the number of clusters to be the same as those present in the data. Visualise the results of clustering as a 2D plot with with coordinates PC1 and PC2 displaying in a distinct way (through colour or marker shape) the members of each cluster produced by k-means.

(c) [4 marks] Compare the percentage of data points clustered correctly in each of the above two cases, i.e., when clustering is applied to all five original attributes vs. applying clustering to the first two principal components (PC1 and PC2). Compare the relative loss of accuracy to the percentage of variance in the data represented by the combination of PC1 and PC2.

Marking guidance Marks will be allocated for working code, and the quality of results.

3 (11 marks) Embeddings

Download the file `PADL-Q3.txt` from the Assessment section of the PADL VLE site. Each line of this text file consists of sequences of integers separated by space, and represents a random walk through a graph where each integer corresponds to a node in the graph, e.g.: "19 15 19 23 22 27".

(a) [5 marks] Treating each line in the data file as a sentence, train a Word2Vec Skip-gram model (as available from the `gensim` library). Demonstrate the results by printing the (cosine) similarities between node 5 and nodes 21 to 30.

(b) [6 marks] Using the same Skip-gram model, produce a distance matrix with as many rows as there are nodes in the graph. For a given row K, the row should contain a space-separated list of nodes, sorted from left to right from the most to the least similar to node K, as based on the model similarity measure. Save the resulting matrix in a text file and include in your submission as file `PADL-Q3-result.txt`.

Marking guidance Marks will be allocated for working code, and the quality of results.

4 (15 marks) Neural network regression

Your task in this question is to train a neural network to predict a body measurement given some other body measurements as input. Specifically, you will train a network that takes as input:

- Chest Circumference (mm)
- Hip Circumference (mm)
- Height (mm)
- Weight (kg)
- Gender (boolean)

and you must predict Waist Circumference (mm). On the VLE you can download a training dataset `body_measurements.csv` containing a header row with the above inputs and output name followed by one person per row comprising the above six attributes. Your task is to train a neural network using any architecture that takes as input the above five values and returns the estimated Waist Circumference value.

Your goal is for the mean absolute error to be as close to zero as possible on an unseen test set. You must include a python script `predict_waist.py` that can be imported and which contains a function `predict(measurements)` as explained in the assessment section of the VLE. The input `measurements` is a $B \times 5$ tensor containing the above measurements in millimetres, kilograms or boolean (0 or 1) as indicated above. Your function must load your pretrained network weights, pass the input values through your network and return a $B \times 1$ tensor containing the estimated Waist Circumference values in mm. Any preprocessing or normalisation that you need to apply to the images must be inside the `predict` function. You must submit your trained network weights as part of your submission and they must not exceed 10MiB.

(a) [5 marks] Describe and justify your chosen network architecture. Marks will be awarded for appropriateness of your design decisions.

(b) [10 marks] Up to 10 marks are available for the performance of your model on the unseen test set. A mean absolute error of < 15mm will score 10 marks. Partial marks will be awarded for larger errors.

5 (15 marks) Neural network image classification

Your task in this question is to train a neural network to classify the type of fashion garment in an image. Your network will take a colour image as input and must classify the image as either t-shirt (0), jumper or hoody (1) or jeans (2). Your task is to train a neural network using any architecture that takes as input an image and returns the predicted class.

On the VLE you can download a training dataset `garment_images.zip`. When unzipped, this contains images and their corresponding class label, numbered as above. Each image is of size $H = 256$, $W = 256$.

Your goal is for the classification accuracy to be as close to 100% as possible on an unseen test set. You do not have access to the test set, but you need to include a python script `predict_class.py` that I can import which contains a function `predict(images)` as explained in the assessment section of the VLE. This function should load your pretrained network from the weights you supply, pass the input images through the network and return `classes`. The input `images` is a $B \times 3 \times 256 \times 256$ PyTorch tensor containing a batch of B images, each with an RGB image of size 256×256 - i.e. the same format as the training data. Intensity values will be in the range $(0, 1)$. Any preprocessing or normalisation that you need to apply to the images must be inside the `predict` function. The output `classes` is a $B \times 1$ tensor containing the integer class, i.e. $\{0, 1, 2\}$, for each input image.

Your notebook must include your training and validation code along with discussion and justification for all design decisions. You are not allowed to use transfer learning on a pretrained network (i.e. you must train your network from scratch) and your saved network weights must not exceed 20MiB.

(a) [5 marks] Describe and justify your chosen network architecture. Marks will be awarded for appropriateness of your design decisions.

(b) [10 marks] Up to 10 marks are available for the performance of your model on the unseen test set. A classification accuracy of $\geq 99\%$ will score 10 marks. Partial marks will be awarded for lower accuracies.

6 (20 marks) Neural image compression

Your task in this question is to train neural networks to encode (compress) an image of a face to a 32D latent representation and to decode (decompress) the latent representation back to an image of the same size as the original. Both encoder and decoder networks may have any architecture of your choosing.

On the VLE you can download a training dataset `faces_images.zip`. When unzipped, this contains 2,095 grayscale face images of size $H = 192$, $W = 160$, aligned by their eye centres.

Your goal is for the reconstruction of an unseen test image from its latent code to be as accurate possible. This will be measured using the structural similarity index (ssim), as implemented in `skimage.metrics.structural_similarity`. You are aiming for the ssim between the original and reconstructed images to be as close to 1.0 as possible.

You do not have access to the test set, but you need to include a python script `compress_images.py` that I can import which contains functions `encode(images)` and `decode(latents)` as explained in the assessment section of the VLE. Each function should load the appropriate pretrained network from the weights you supply, pass the input through the network and return latents or images. Specifically, the input to encode is `images`, a $B \times 1 \times 192 \times 160$ PyTorch tensor containing a batch of B images, each with a grayscale image of size 192×160 - i.e. the same format as the training data. Intensity values will be in the range $(0, 1)$. Any preprocessing or normalisation that you need to apply to the images must be inside the `encode` function. The output `latents` is a $B \times 32$ PyTorch tensor containing the encoded latents. The input to decode is the latents just described. It should output images in the same format as input to your encode function.

Your notebook must include your training and validation code along with discussion and justification for all design decisions. You are not allowed to use transfer learning on a pretrained network (i.e. you must train your network from scratch) and your saved network weights must not exceed 20MiB.

(a) [5 marks] Describe and justify your chosen network architectures. Marks will be awarded for appropriateness of your design decisions.

(b) [5 marks] Plot training and validation losses and use this to justify hyperparameter choices.

(c) [10 marks] Up to 10 marks are available for the performance of your model on the unseen test set. An SSIM of ≥ 0.9 will score 10 marks. Partial marks will be awarded for lower similarities. If your encoder returns latent representations larger than 32D you will score zero marks for this part.

END OF PAPER