Deep Learning Assignment 1 2024

- In this assignment, you will implement a deep neural network from first principles
  - This will greatly deepen your understanding of neural nets!
  - Multi-part assignment, drawing on material covered each week
  - You are expected to start work on it immediately; it is a bad idea to wait until near the deadline
- You must use a Jupyter Notebook:
  - Python recommended but R is OK too
  - You can use low-level functionality of numpy and scikit-learn, but **not** their implementations of anything core (for example, you cannot use their optimisers, gradient descent implementations, etc.)

1-Person or 2-Person Assignment: You Choose

- You can do the assignment alone or you can join with 1 other person
  - Groups larger than 2 are not allowed
  - Final part will involve separate contributions by both people
  - I recommend doing it with another person rather than alone: two heads are helpful when trying to figure out algorithms and debug code
- Policy on two-person submissions:
  - You are encouraged to work in pairs for Parts 1-4, but you cannot collaborate with any other people other than that
  - Part 5 must be done twice for a 2-person assignment
  - Your documentation and code must be clear about what contributions each person made; it is **not** sufficient to say something like “we both worked on everything together” – put a comment in each code block saying who wrote it, or I will not award any marks for that section
  - Across all parts of the assignment, I may award different marks to each person doing a 2-person assignment if I feel that their contributions are not equal.

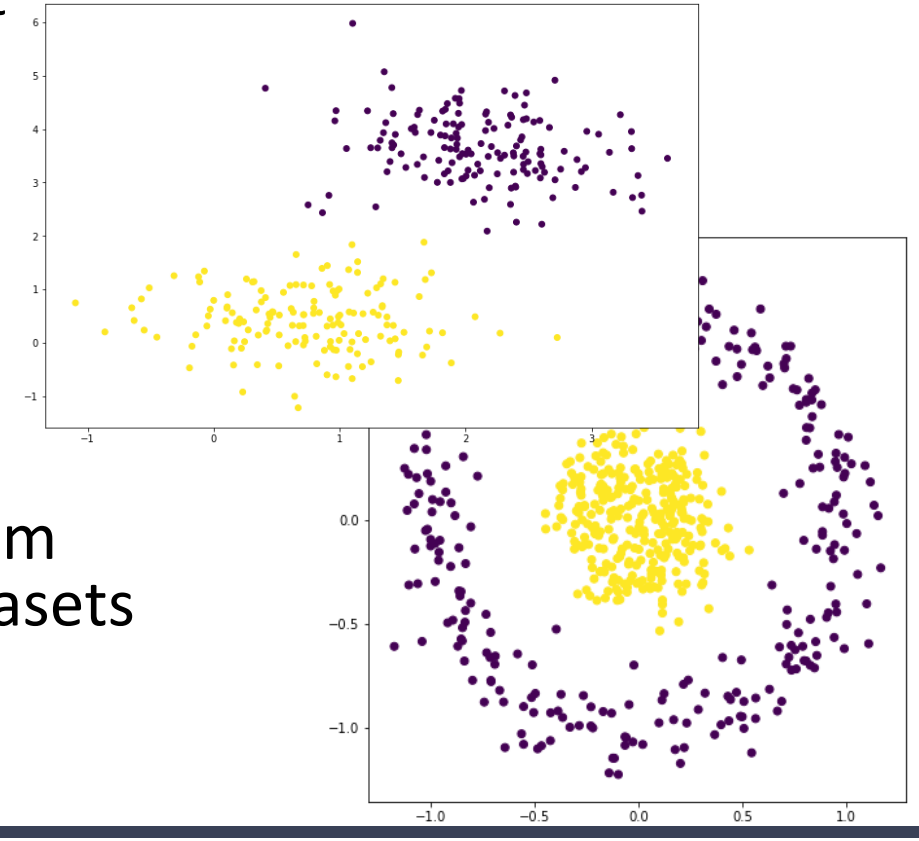
Part 1: Implement Logistic Regression

- Implement Logistic Regression (Topic 2):
  - Use Jupyter Notebook (Python or R) to implement a neural network approach to logistic regression (no hidden layers, one output node)
  - Your code should follow my notes to implement the algorithm from scratch
  - Your notebook should include a brief description of the algorithm, with all references
  - Your code must handle different numbers of inputs and different numbers of training cases, but you don't have to support more than one binary output node

- Please note:
  - Post questions on discussion forum if there are gaps/inconsistencies, or if you are unclear about anything
  - **Do not submit code taken from the web (plagiarism)**
  - **You are not permitted to use generative AI in this assignment**
  - ANY academic misconduct will result in 0 for full assignment
  - Reference ALL sources you used (not doing so is academic misconduct)

Part 2: Easy Tasks

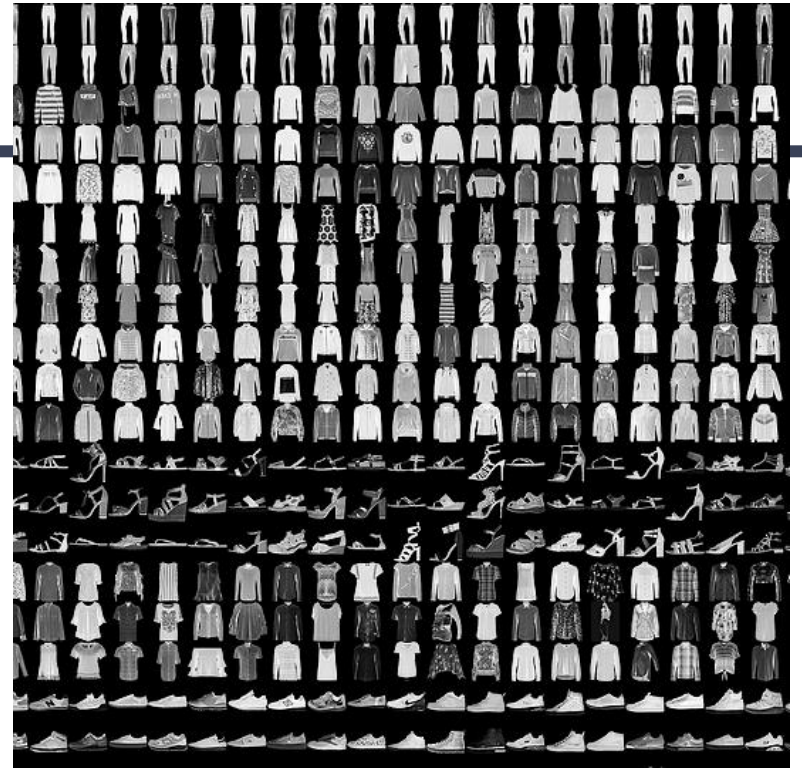
- I will supply two fairly small datasets:
  - One will be linearly separable (almost or fully), the other will not
  - I will provide sample Python code to load and plot the datasets; you **are allowed** to use this code in your own assignment
- Divide each dataset randomly into:
  - Training set (70%): use for main training
  - Validation set (15%): use for tuning, e.g. selecting learning rates
  - Test set (15%): held out set for final performance evaluation
- Train a logistic regressor using your code from Part 1, and see how it performs on both datasets
- In your notebook, summarise results and provide observations and conclusions



Part 3: Implement and Test a Shallow Neural Network

- Implement a shallow neural network (Topic 3)
  - Start with your logistic regression code
  - Again, follow my notes as closely as possible
  - Include a brief description of the algorithm, with all references
  - Your code just needs to support 1 hidden layer, no more
  - Your code just needs to support 1 output node, no more
  - Your code **must** support different numbers of input nodes and different numbers of nodes in the hidden layer
- Test your implementation on the two small datasets:
  - Review the results and provide observations and conclusions in the notebook
  - Is it able to handle the linearly separable data? (I hope so)
  - Does it perform better than logistic regression second dataset?
  - Does changing the number of nodes in the hidden layer help?

Part 4: Challenging Task



- We will tackle an image recognition task
  - We will use the Fashion-MNIST dataset https://github.com/zalandoresearch/fashion-mnist
  - 10 classes: different types of clothing
  - Each image is 28x28 pixels, 8-bit grayscale
  - 60k training images, 10k testing images: you can select random subsets
  - Small but interesting & challenging dataset
  - I'll provide sample code to read it
- You must try to distinguish between **2 classes only:**
  - **You must send me an email message to request your pair of classes;** don't just pick a pair of classes yourself
  - If doing it in a group, send one email and CC your collaborator
  - Use a network with 1 hidden layer (larger than first case), 1 output
- *PhD students: If you want to use a different dataset related to your research, contact me in advance for agreement*

Part 5: Deep Learning Enhancements

- Each student: **support an arbitrary number of hidden layers** and pick one other enhancement that is characteristic of deep learning (Topics 4 & 5)
  - Update your code to support an arbitrary number of layers, and an arbitrary different number of nodes in each layer layer
  - In your notebook, have a section that briefly describes how you extend the code to support more layers, and describing your enhancement, with references
  - Implement your enhancement
  - Test how well this works relative to the shallow NN on the image dataset

- If in a group, this part must be done individually:
  - Do **not** do it with the other member of your group
  - The two group members must pick different enhancements from each other
  - Include 2 separate sections in your notebook, each written separately by each of you, about what you alone did in Part 5

Assignment Submission & Deadline

- Make a submission at or before the deadline, covering all parts of the assignment, via Canvas
- You must submit it in TWO formats:
  1. The Notebook itself, including all output from running it (do not expect us to run it)
  2. A PDF print of the Notebook (easier for us to grade)

  Note: avoid unnecessary outputs in your notebook; for example, don't print out all rows of the training data file, or every single training iteration
- For a 2-person group:
  - Only 1 person must make the submission, with both group members' names and IDs on it.
  - The submission must have two separate sections for Part 5, one for each group member.
- **DEADLINE: Sunday 18 February 2024 at 18:00**

Assignment Marking Scheme

- I have posted a detailed marking scheme on Canvas, showing exactly where all marks are earned.
- As noted earlier, this is a multi-part assignment, and later parts build on earlier parts
  - However, all parts will earn you marks
  - Therefore, it is important that you don't delete or over-write your code from Part 1 or Part 2 when you move on to Parts 3, 4, 5
  - If necessary, copy and modify the code so that we can see all parts and grade them all
- Even if you don't get the assignment completely finished by the deadline, submit what you have done and you should earn most of the marks (unless any part has academic integrity problems)

Academic Integrity

- *Academic misconduct includes: submitting work as your own that was done in part by somebody else, or created using AI; using contract cheating services; submitting work that came from a friend or from a previous assignment; including unreferenced material (even if paraphrased); providing your solutions to other people.*
- It is **your** responsibility to ensure that you understand the university's rules: please see QA220 Academic Integrity Policy and other resources provided.
- You are required to act with academic integrity in carrying out your assignments.
- If you have notified me that you are doing your assignment in a 2-person group, you should collaborate with that person, but you must not work with anybody else.
- You are **NOT** permitted to use generative AI in this assignment for any purpose, including creating any parts of your code or helping you write part of the text.
- Academic misconduct is particularly serious at postgraduate level.
- If there is academic misconduct in **any part** of your assignment, you will get 0 for your full assignment, with no option to resubmit, and your case will be entered on the university's Academic Misconduct Register.