In this assignment, you will build and train a Boltzmann machine on the MNIST dataset. Boltzmann machines are a specific type of PDP model designed for unsupervised learning. You can find out more about them in the lectures and readings for week 4 of the class.

The purpose of this assignment is three-fold:

1) To get you familiar with a canonical, yet slightly esoteric, neural network model, so that you don't fall prey to the idea that neural networks are all deep convolutional networks.

2) To force you to think about the actual learning algorithm for a neural network and how to implement that in code in an elegant manner.

3) To have you engage in some more in depth thought about unsupervised learning and the relationship between unsupervised learning and brains.

To be clear, Boltzmann machines are not always easy to get working. It is possible that your network will not train well. Don't stress, that's not the entire point of this assignment. Most of your marks will come from your answers to some questions, the general structure of your code, and evidence that you at least understood how it *should* work. Even if your Boltzmann machine doesn't train well, you can still do okay in this assignment. If it does train well, and you spend the time to answer the questions below, you will likely smash this assignment. Also, remember that Raymond and I are here to help you. Use office hours if you're getting stuck!

Submit your assignment on MyCourses. You must submit both your code (i.e. the modified Jupyter notebook) and also a **pdf** document with your answers to the questions below. You will be marked on both your code and the document, don't skip either!!!!

The assignment is out of 70 marks, and there are 10 bonus marks available (see the marking scheme below).

Part 0 – Getting started (no marks, though if you don't do this you won't get any marks!)

- To do this assignment you need to have Python installed, as well as numpy, a plotting library (e.g. matplotlib), and Jupyter notebook. One way to kill multiple birds with one stone here is to install Anaconda or Miniconda: https://docs.conda.io/en/latest/
- Make sure you can open the Jupyter notebook provided (boltzmann_template.ipynb), which gives you the skeleton for your code. To open it you use Jupyter, e.g., on a Linux system if you have Jupyter installed you can type this into the command prompt:
    - $ jupyter notebook boltzmann_template.ipynb
- Make sure you understand how to access and plot MNIST images. Raymond has already given you an example in the notebook. Try plotting some other images, make sure you understand what you are looking at.

- You must put your code into this notebook, and submit it with a new name. The new name of the notebook must adhere to this format:
    - boltzmann_[firstname]_[lastname]_[mcgillid].ipynb
    - Where the items in brackets must be replaced with your specific info.

Part 1 – Understanding Boltmzann machines (30 marks total)

Answer the questions below, and put your answers in the pdf document that you will submit along with your code.

- **Question 1.1 (8 marks):** In writing, define a Boltzmann machine using the eight elements of the PDP framework described in week 3 of the class (see the lecture slides and the readings).
- **Question 1.2 (2 marks):** What is the loss function that a Boltzmann machine minimizes?
- **Question 1.3 (2 marks):** What is the partial derivative of the loss function with respect to a synaptic weight?
- **Question 1.4 (4 marks):** Explain in words what the various terms of the equation from Q1.3 represent. What do you have to do to calculate them?
- **Question 1.5 (4 marks):** What is the "temperature" of a Boltzmann machine, i.e. what does it control and what impact does it have on the network's behaviour?
- **Question 1.6 (4 marks):** In your own words, what is the "annealing schedule" of a Boltzmann machine and why is it important?
- **Question 1.7 (6 marks):** What does the equation in Q1.3 have to do with dreaming? What is the implied theory of what dreams are and why we forget them?

Part 2 – Designing your Boltzmann machine (20 marks in total)

Answer the questions below, and put your answers in the .pdf document that you will submit along with your code.

- **Question 2.1 (4 marks):** What variables did you declare in your Boltzmann machine class? Why did you include these in the class, i.e. justify your design decisions?
- **Question 2.2 (4 marks):** How will you calculate the weight updates for the network?
- **Question 2.3 (4 marks):** How will you define the annealing and training schedule?
- **Question 2.4 (8 marks):** Is your code well commented, is it logical? This question is rhetorical, don't answer it. Raymond will mark you based on your code style here. You get all 8 marks only if he can understand your code in 5 minutes flat.

Part 3 – Running your Boltzmann machine (20 marks in total + 10 available bonus marks)

Answer the questions below as indicated.

- **Question 3.1 (10 marks + 5 bonus marks available):** You don't actually need to answer the questions below in your pdf document. Instead, here, Raymond will try to run the code in the Jupyter notebook you submit. Your marks will depend on what happens for him. Does your code run for Raymond and print out training and testing image loss? If so, you get 10 marks. Partial marks if it does something but doesn't actually run multiple training epochs. Does the training and testing loss go down? If so, you get 5 bonus marks.
- **Question 3.2 (10 marks + 5 bonus marks):** If you run your Boltzmann machine freely (i.e. without clamping the visual units), what happens? What images are generated on the visible units? In your own words, describe what you think your Boltzmann machine has or hasn't learned, and how this might relate to a healthy or unhealthy brain. You will get 5 marks for your answers to these questions and the ideas behind them (write it in your pdf document). You will get another 5 marks if Raymond can run your image generation code and see some plotted images. You will get 5 bonus marks if the images your Boltzmann machine generates look anything like MNIST digits.