# COMP596 – Brain-inspired AI
# Programming Assignment 1
## Winter 2021
## Due: February 22nd before 11:59 PM.

In this assignment, you will build and train a Hopfield network to store images from the MNIST dataset. Hopfield networks are a specific type of PDP model designed for auto-associative learning. You can find out more about them in the lectures and readings for week 5 of the class.

The purpose of this assignment is three-fold:

1) To get you familiar with a classic neural network model that is very different from most of the models we get exposed to in machine learning nowadays, 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 auto-associative learning and the relationship between auto-associative learning and brains.

To be clear, Hopfield networks can sometimes be wonky to train. Remember that Raymond and I are here to help you. Use office hours if you're getting stuck! But, also note that your marks not only depend on the code, but also on the style of your code and your answer to some questions (given below). So, even if your network doesn't work perfectly you can still get a decent mark.

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 download the code and data provided and open the Jupyter notebook (`hopfield_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 hopfield_template.ipynb`
- Make sure you understand how to access and plot MNIST images. I have 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:
  - ○ `hopfield_[firstname]_[lastname]_[mcgillid].ipynb`
  - ○ Where the items in brackets must be replaced with your specific info.

## Part 1 – Understanding Hopfield networks (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 Hopfield network 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 energy function for a traditional Hopfield network, and why do we call it an "energy" function?
- **Question 1.3 (2 marks):** What is the partial derivative of the energy function with respect to a synaptic weight?
- **Question 1.4 (4 marks):** If we used the answer you gave to Question 1.3 to do gradient descent for training one pattern at a time, what type of synaptic plasticity rule is being implemented (i.e. what type of neuroscience based rule)? Explain your answer—what makes this that type of rule?
- **Question 1.5 (4 marks):** If you wanted to train a Hopfield network on multiple patterns in one go, what equation could you use to determine the values of the synaptic weights?
- **Question 1.6 (4 marks):** In your own words, what do these learning rules do to the energy landscape of a Hopfield network?
- **Question 1.7 (6 marks):** What do you think would happen in a Hopfield network if you presented the complement (i.e. the opposite) of a stored pattern to the network? Consider the equation you gave in to Questions 1.2 and 1.5, and think about what it implies for memory storage in Hopfield networks.

## Part 2 – Designing your Hopfield network (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 Hopfield network class? Why did you include these in the class, i.e. justify your design decisions?
- **Question 2.2 (4 marks):** First, design a one-pattern-at-a-time learning rule function. How will you calculate the weight and threshold updates for the network?
- **Question 2.3 (4 marks):** Now, define a mass pattern storage learning function. How will you set the synaptic weights and thresholds?

- **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 Hopfield network (20 marks in total + 10 available bonus marks)

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.

- **Question 3.1 (10 marks + 5 bonus marks available):** First, Raymond will set the `one_pattern` flag to `True` (see the code to understand what this does). Does your code run for Raymond and print out a sequence of images showing the evolution of the activity pattern in the Hopfield network starting at a noisy version of one of the images and converging to a stored image? If so, you get 10 marks. Partial marks if it does something but doesn't actually converge to a stored image. What if he gives it a cross between two images—does it converge to one of them? If so, you get 5 bonus marks.
- **Question 3.2 (10 marks + 5 bonus marks):** Now, Raymond will set the `one_pattern` flag to `False`. Again, does your code run for Raymond and print out a sequence of images showing the evolution of the activity pattern in the Hopfield network starting at a noisy version of one of the images and converging to a stored image? If so, you get 10 marks. Next, what happens if he gives the network a cross between two images? Again, you will get another 5 marks if it converges to one of the stored images.