COMP 549 – Brain-inspired AI

Programming Assignment 2

Winter 2025

Due: April 4th, 2025 - before 11:59 PM.

In this assignment, you will complete the code for a Helmholtz network and train it using the MNIST dataset. Helmholtz networks 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 7 of the class. As well, there is a required reading for this assignment, Hinton et al. (1995), and you will have to answer some questions based on this paper. You will also use it to help you design your code.

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.

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

3) To have you engage in some more in depth thought about unsupervised learning and challenges of measuring/observing it.

To be clear, Helmholtz networks can sometimes be wonky to train. Remember that Arna 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 the 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 5 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 data (`mnist.pkl.gz`) and skeleton code in the Jupyter notebook (`helmholtz_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 helmholtz_template.ipynb`
- Make sure you understand how to access and plot MNIST images. I have already given you examples in the notebook. Try plotting some other images, make sure you understand what you are looking at.
- Make sure you have also downloaded the Hinton et al. (1995) paper and read it before you start coding. You should also have already read the Dayan et al. (1995) paper from the regular course readings.
- 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:
    - `helmholtz_[firstname]_[lastname]_[mcgillid].ipynb`
    - Where the items in brackets must be replaced with your specific info.

Part 1 – Understanding Helmholtz networks (32 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 (4 marks):** In your own words (don't just quote the paper), what is the goal of a Helmholtz machine? In other words, what are we trying to learn with a Helmholtz machine?
- **Question 1.2 (8 marks):** In your own words, what is the description length cost of a representation in a Helmholtz machine? What does a low description length imply for the recognition and generative pathways in a Helmholtz machine?
    - *Hint*: the answer to this question lies in equations (2) and (3) of the Hinton et al. (1995) paper.
- **Question 1.3 (8 marks):** What is the loss function that Helmholtz networks reduce? Explain in your own words what the different variables and terms are and what they mean for learning.
    - *Hint*: the answer to this question lies in equations (5), (6), and (8) of the Hinton et al. (1995) paper and equations (2.2)-(2.6), of the Dayan et al. (1995) paper. Compare the two different papers' equations to help yourself understand what they mean.
- **Question 1.4 (4 marks):** There are two different phases of training for a Helmholtz machine with different weight updates, a wake phase, and a sleep phase. What are the weight updates for each phase? Define all the variables clearly and describe in your own words how to calculate them.
    - *Hint*: see equations (4) and (7) of the Hinton et al. (1995) paper.
- **Question 1.5 (8 marks):** What components of the loss function do each training phase (wake and sleep) help to reduce? In your own words, how do they do this and why are they complementary?
    - *Hint*: Consider figure 2 from the Dayan et al. (1995) paper.

Part 2 – Filling in the code for the Helmholtz network (18 marks in total)

Here, your job is to fill in the missing code from the `wake` and `sleep` functions in the Helmholtz network class.

- **Question 2.1 (4 marks):** What variables do you need to calculate here to make your weight update? Why are these variables necessary for reducing the various components of the loss function?
  - *Hint*: Consider your answer to Questions 1.4 and 1.5 here.
- **Question 2.2 (4 marks):** Is your code well commented, is it logical? This question is rhetorical, don't answer it. The TAs will mark you based on your code style here. You get all 4 marks only if they can understand your code in 5 minutes flat.
- **Question 2.3 (10 marks):** Does your code run and does the network train? To determine this, examine the plot of the description length cost. If it goes down at first and then comes up, training is working appropriately. If necessary, play with the learning rate. Try generating some dreams and examine the weights. You should be able to see something semi-intuitive. You will get full marks here if training is clearly working well, and partial marks if the code runs but training doesn't really work. You will get zero if the code doesn't run at all (e.g. if the TAs get error messages).

Part 3 – Examining your Helmholtz network (20 marks in total + 5 bonus marks)

Here you will examine the impact of different training regimes for your Helmholtz network. If your code doesn't work, it will be very hard to get full marks here, unfortunately. But, you can still get partial marks by answering the questions below with an educated *guess* about what you might expect to see.

- **Question 3.1 (5 marks):** What happens when you train the network on a single class? What sorts of "dreams" does the network have and what do the weights show? What about if you train it on two classes? And what if you train it on all the classes? Given what you see, do you believe that the network is achieving its ultimate goals in training or not (consider your answer to Question 1.1)? What problems do you see?
- **Question 3.2 (5 marks):** What happens when you increase the number of hidden units by a lot (e.g. to 100)? Does this change what you said in Question 3.1 at all? Note that you may have to adjust the learning rate when you adjust the number of units.
- **Question 3.3 (10 marks + 5 bonus marks):** What happens when you turn off either the wake or sleep phase (one at a time, not both at once, which obliviously just eliminates any training)? How does it affect the dreams and the weights? Explain these results in your own words.
  - **For 5 bonus marks:** How does turning off either wake or sleep affect the description length cost? Can you explain why using the loss function and your answer to Question 1.4, and can you use this to explain why the description length starts to increase a bit during training?