COMP9444: Neural Networks and Deep Learning

Week 1a. Overview

Sonit Singh
School of Computer Science and Engineering

May 28, 2024

Course Web Pages

- **WebCMS**: https://webcms3.cse.unsw.edu.au/COMP9444/24T2/
- **Ed**: https://edstem.org/au/courses/16392/discussion/
- **Moodle**: https://moodle.telt.unsw.edu.au/course/view.php?id=84359

Lecturers, Course Admins

- **Lecturer**: Dr Alan Blair
- **Lecturer-in-Charge**: Dr Sonit Singh
- **Course Admin**: Zhongsui Guo
- **Email**: cs9444@cse.unsw.edu.au

Course Schedule

Lectures (Week 1 to Week 10)

- **Lecture A**: Tuesday; 2-4 pm; Science Theatre (K-F13-G09)
- **Lecture B**: Wednesday; 11-1 pm; Science Theatre (K-F13-G09)

Tutorials (Week 1 to Week 5)

- Check your respective tutorial on https://timetable.unsw.edu.au/2024/COMP9444.html#S2-4290

Project Mentoring Sessions (Week 6 to Week 10)

- Will be released by Week 4

CSE Help Sessions (Optional, tentative)

Teaching Strategies

- Course materials will be delivered through the course Ed page — including text, images, online discussion forums, quizzes, and coding exercises.
- You are encouraged to read through the materials on Ed before each Lecture.
- Lecture time will be used to summarise the material, discuss recent developments, and answer questions.
- Tutorials in Week 1 to Week 5, to discuss worked examples and develop a deeper understanding of fundamental topics.
- Mentoring sessions (Week 6 to 10) to assist with Group Project.
- Help sessions will assist with any queries related to course content and to assist to assessments.

Teaching Strategies

You must keep up with lectures, either by attending in person or watching the recordings. Students enrolled in the Web stream are welcome to attend in person if space is available.

You are expected to:

- review course materials before and after each lecture.
- attempt tutorial questions beforehand and be ready to ask questions.
- complete quizzes, coding exercises, and relevant questions.
- discuss the material with your fellow students if possible.
- consider further exploring topics of particular interest.
- ask questions and contribute to discussion in online Ed forums.

Textbook(s)

The textbook for this course is:

Deep Learning
by Ian Goodfellow, Yoshua Bengio and Aaron Courville
MIT Press, 2014
https://www.deeplearningbook.org/
https://mitpress.mit.edu/9780262035613/deep-learning/

Another good reference book is:

Understanding Deep Learning
by Simon J.D. Prince
MIT Press, 2023
https://udlbook.github.io/udlbook/

Assumed Knowledge

The course will assume knowledge of the following mathematical topics:

- Linear Algebra (2.1-2.8)
- Probability (3.1-3.14)
- Calculus and Chain Rule (6.5.2)

You should study the relevant sections of the textbook (shown in brackets) and, if necessary, try to revise these topics on your own during the first two weeks of the course.

Planned Topics (Weeks 1-5)

- **Week 0**: Introduction to course, Python refresher, Numpy refresher, Matplotlib refresher, Google Colab refresher
- **Week 1**: Neuroanatomy and Perceptrons, Multi-layer perceptrons and Backpropagation
- **Week 2**: Probability, Generalisation & Overfitting, PyTorch
- **Week 3**: Cross Entropy, SoftMax, Weight decay, Momentum, Hidden Unit Dynamics
- **Week 4**: Convolutional Neural Networks, Image Processing
- **Week 5**: Recurrent Neural Networks (RNN), Long Short-Term Memory Network (LSTM), Gated Recurrent Unit (GRU)

Planned Topics (Weeks 6-10)

- **Week 6**: Flexibility Week
- **Week 7**: Word Vectors, Language Processing, Large Language Models (LLMs)
- **Week 8**: Reinforcement Learning, TD-learning and Q-learning, Policy Learning, Deep Reinforcement Learning
- **Week 9**: Autoencoders, Adversarial Training, Multimodal learning
- **Week 10**: Generative Artificial Intelligence (GenAI), Review

Assessments

Assessments will consist of:

- Assignment (individual): 20%
- Group Project (group-based): 35%
- Final Exam (in-person, invigilated): 45%

Due dates:

- Assignment 1: Due Week 5
- Group Project: Due Week 10
- Final Exam: UNSW Exam Period

Note: In order to pass the course, you must achieve a total mark of at least 50.
Note: Students are expected to form themselves into groups of 5 for the group project by the end of Week 3. Each group will be assigned a Mentor.

Assignment (individual)

The assignment may involve, for example:

- using code written in PyTorch
- writing your own code
- running experiments and analysing the results

Further details will be provided on the course website.

Group Project

The group project involve the following:

- forming team (5 members) by the end of Week 3
- team members can be a mix of undergraduates/postgraduates or from different tutorials
- choose group based on the provided list
- a mentor will be assigned to each group based on their expertise
- a new timetable will be released for group mentoring sessions
- Discuss your project progress with your mentor and seek help throughout the term
- Project evaluations in Week 10
- Deliverables: Source code (Jupyter Notebook), Presentation, Report

Group Project

List of projects will be provided and students have the flexibility to choose one.

Project Title: Insect pest specifies identification.

Area of Research: Computer Vision

Problem Statement: Insect pest classification plays a crucial role in various domains, including agriculture, pest control, and ecological research. Rapid and accurate identification of insect pests is essential for effective pest management strategies, early detection of invasive species, and preservation of crop yield and quality. However, manual classification of insects based on visual inspection can be time-consuming, error-prone, and challenging, particularly when dealing with conditions in the wild. The goal of this project is to correctly identify the species of insects in an automated manner using advanced artificial intelligence algorithms which has high accuracy, robust to varying environmental conditions, appearance, and deploying these algorithms for real-time monitoring.

Dataset:

Figure 1. Example images of the IP102 dataset. Each image belongs to a different species of insect pests.

The IP102 dataset [1] is a benchmark dataset for insect pests. The details about the dataset can be found in [1] and be downloaded from the URL given below:

Dataset URL: https://github.com/xpwu95/IP102/tree/master

Task: To develop an automatic insect recognition system using neural networks and deep learning which provides high accuracy, robust to varying appearance and similarity between different insect species, and faster so that it can be deployed in the real world.

Relevant Papers

[1]. X. Wu, C. Zhan, Y.-K. Lai, M. -M. Cheng and J. Yang, "IP102: A Large-Scale Benchmark Dataset for Insect Pest Recognition," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 2019, pp. 8779-8788, doi: 10.1109/CVPR.2019.00899.
https://ieeexplore.ieee.org/document/8954351
https://openaccess.thecvf.com/content_CVPR_2019/papers/Wu_IP102_A_Large-Scale_Benchmark_Dataset_for_Insect_Pest_Recognition_CVPR_2019_paper.pdf

[2]. A. Setiawan, N Yudistira, and R.C. Wihandika, "Large scale pest classification using efficient Convolutional Neural Network with augmentation and regularizers", Computers and Electronics in Agriculture, Vol. 200, Sept 2022.
https://www.sciencedirect.com/science/article/pii/S0168169922005191

[3]. W. Linfeng, L. Yong, L. Jiayao, W. Yunsheng, and X. Shipu, "Based on the multi-scale information sharing network of fine-grained attention for agricultural pest detection", PLOS ONE 18(10):e0286732.
https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0286732

[4]. An J, Du Y, Hong P, Zhang L, Weng X., "Insect recognition based on complementary features from multiple views", Scientific Reports. 2023 Feb;13(1):2966. DOI: 10.1038/s41598-023-29600-1.
https://europepmc.org/article/pmc/pmc9940688

[5]. S. Kar, J. Nagasubramanian, D. Elango, M. E. Carroll, C. A. Abel, A. Nair, D.S. Mueller, M. E. O'Neal, A. K. Singh,

PyTorch

We will be using PyTorch for implementing neural networks in this course. Please try to install equal or later versions on your own machine.

python3 3.11.2 torch 1.13.0
numpy 1.24.2 sklearn 1.2.1

Anaconda is the recommended package manager since it installs all dependencies. Please install anaconda depending on your operating system.
Anaconda
Getting started with PyTorch

We are going to make heavy use of Jupyter Notebooks for demos and showcasing code examples. Jupyter Notebook holds text, code, and output, all in the same file, making it really a great document. You can read more about Jupyter Notebooks on the project website Jupyter.

Hardware acceleration for deep learning

- This course will require coding and running tasks that involve heavy numerical computing, such as multiplication of large number of matrices. Although part of the content can be follow along using a standard personal computer or laptop having PyTorch installed on it. However, we anticipate that training deep learning algorithms from scratch or running a pretrained network, and doing assessments will require a CUDA-enabled GPU machine.
- Note that currently only NVIDIA's graphics cards support CUDA and cuDNN libraries, and hence can be used accelerated training of deep learning algorithms.
- Cloud based options: Google Colab

Plagiarism

- Plagiarism is taken seriously by UNSW/CSE and treated as Academic Misconduct. ALL work submitted for assessment must be your own work.
- For an individual assignment, collaborative work in the form of “think tanking” is encouraged, but students are not allowed to derive code together as a group during such discussions. In the case of a group assignment, code must not be obtained from outside the group.
- Plagiarism detection software may be used on submitted work.
- Check Academic Integrity and Plagiarism

Related Courses

- COMP3411/9414 Artificial Intelligence
- COMP9417 Machine Learning and Data Mining
- COMP9418 Advanced Topics in Statistical Machine Learning
- COMP4418 Knowledge Representation and Reasoning
- COMP9491 Applied Artificial Intelligence
- COMP9517 Machine Vision
- COMP3431 Robotic Software Architecture
- COMP9727 Recommender Systems
- COMP6713 Natural Language Processing
- 4th Year Thesis topics

Tasks to do in O-Week

- Refresh course pre-requisites
  - Linear Algebra, Probability, Calculus and Chain Rule
  - Python refresher
  - Numpy refresher
  - Matplotlib refresher
  - Google Colab
  - Introduction to PyTorch