Introduction to neural networks and deep learning

PSYC GR

Fall 2021 (3 points)
Mondays, 4 pm - 6 pm, Zoom

Instructor: Nikolaus Kriegeskorte (n.kriegeskorte@columbia.edu)

Prerequisites: This seminar requires the ability to program in Python and an understanding of linear algebra. If you are planning to take this course, but are not confident in either of these two areas, consider taking relevant courses in preparation (for example online courses). You should have Python and PyTorch installed on your laptop and have some experience programming in Python. You should also have some experience with Jupyter notebooks. Experience using PyTorch is not required. A final essential prerequisite is that you have looked at the textbook *Dive into Deep Learning* (https://d2l.ai/) to get a sense of the volume and nature of the required reading and that you have read Chapter 1 (Introduction).

Course description: This seminar will introduce both the concepts and practical implementation in PyTorch of neural networks and deep learning, with a focus on general principles and examples from vision.

Detailed description: Inspired by brains, neural network models provide an abstract framework for implementing parallel distributed information processing systems. This framework has earned a central role in both the engineering of artificial intelligence and the modeling of the information processing in the brains of humans and other animals. In this graduate-level seminar, we will learn about neural network models and deep learning. We will aim to develop: (1) an understanding of how neural network models compute and learn and (2) the practical ability to implement our own models using Python and PyTorch. The course will cover architectures including multilayer perceptrons, convolutional neural networks, and recurrent neural networks, along with key concepts including supervised and unsupervised learning, overfitting and generalization, and the backpropagation algorithm. We will equally emphasize an intuitive and a mathematical understanding of these concepts. In addition, we will learn how to implement neural network models of our own design using Python and PyTorch. We will not discuss any aspect of how to program in Python, but will discuss the conventions of the PyTorch library, which powerfully automatizes many of the required computations, leveraging the specialized hardware of graphics processing units (GPU).

Learning objectives: This seminar will develop a conceptual understanding of and the practical skills needed to implement neural network models.

- Understand the basic components (units, activation functions, architectures) of deep neural networks.
- Understand how multilayer perceptrons, convolutional neural networks, and recurrent neural networks work.
- Understand stochastic gradient descent and the backpropagation algorithm.
- Understand alternative optimization algorithms.
- Understand the concepts of overfitting, generalization, and out-of-distribution generalization.
- Understand the implementation of neural networks using PyTorch and GPUs.
- Be able to implement and train novel neural network architectures in Python using PyTorch.

Structure of the weekly work and seminar sessions: Every week we will learn about a different piece of the puzzle, usually focused on a set of chapters of the online textbook *Dive into Deep Learning* (`https://d2l.ai/`). The reading constitutes a major commitment, sometimes exceeding 100 pages per week. The second major commitment is the coding of solutions to exercises and of creative extensions thereof. In class, the first hour is for slide presentations. The slide presentation, by the instructor and or a participant, will complement the reading, recapitulating essential insights with the aim to deepen our conceptual, intuitive, and mathematical understanding of the week's topic. The second hour is for code discussion. Participants will present and discuss Python notebooks they have prepared, reviewing the PyTorch implementation of the week's topic, their solutions to programming exercises, and their own creative extensions related to the topic. Every participant will prepare and present at least three Python notebooks. All notebooks should contain text explanations that enable the reader to understand the code, in the spirit of literate programming. The notebooks will be shared among all participants as a record for future reference and as a springboard for development of their own neural network models.

Grading: Each participants are expected to do the reading, reflect on the ideas, play with the code and do the exercises. In addition, each participant will rank the weekly topics by interest and will be assigned by an optimization algorithm to do an interactive slide presentation on one of the topics and to prepare Python notebooks for three of the topics. The Python notebooks should be self-contained explorations of topic of the week or one of two flavors: (1) tackling an exercise or (2) creatively expanding on the topic. In either case the notebook should contain a full and self-contained explanation of the problem as well as of the solution, along with the code and high-quality figure output. Grading will be based on the quality of the notebooks (50%), the quality of the interactive slide presentation (25%), and on attendance and class participation (25%).

Weekly topics and readings

Date	**Topic**	**Material**
09/13	Introduction and preparation	Preface, Installation, Notation, Chapter 2
09/20	Mathematics for deep learning	Chapter 18
09/27	Linear neural networks	Chapters 3
10/04	Multilayer perceptrons	Chapters 4
10/18	Deep learning computation	Chapter 5
10/25	Convolutional neural networks	Chapter 6
11/01	Modern convolutional neural networks	Chapter 7
11/08	Recurrent neural networks	Chapter 8
11/15	Modern recurrent neural networks	Chapter 9
11/22	Attention mechanisms	Chapter 10
11/29	Optimization algorithms	Chapter 11
12/06	Computational performance	Chapter 12
12/13	Computer vision	Chapter 13
12/20	Generative adversarial networks	Chapter 17