# Advanced Programming and Deep Learning for AI

## Prof. Andrea Pozzi; Prof. Jianyi Lin

# *COURSE AIMS AND INTENDED LEARNING OUTCOMES*

In recent years, the advances in artificial neural networks, supported by plentiful availability of data and computational resources, have led to the design of deep learning models that are widely used in data analytics frameworks. First, this course provides students with further programming techniques in Python language, selected topics on data structures and algorithmics for sequential and graph-based data. The second part of the course aims to introduce the modern deep neural network models and tools, chiefly convolutional and recurrent architectures, which are widely applied in various AI domains, such as artificial vision and natural language processing.

At the end of the course, students are expected to achieve the following course learning outcomes:

* understand the concepts of computational problems and algorithms, classical linear and graph-based data structures, as well as the notion of abstract data type; demonstrate knowledge of the characteristics of modern deep learning architectures and related algorithms (*Knowledge and understanding*)
* elaborate and implement algorithmic solutions for a given computational problem using an appropriate data structure; implement specific neural network models for data analysis applications (*Applying knowledge and understanding*)
* evaluate the computational complexity of an algorithm; choose suitable and efficient data structures; assess the applicability and performance of a deep learning technique for a given task (*Making judgements*)
* describe and justify a solution using proper computer science terminology, such as an algorithm for solving a computational problem or a deep learning architecture for some prediction task (*Communication skills*)
* autonomously refer to authoritative documentation of the language and framework adopted; recognise data structures and neural network architectures at abstract level (*Lifelong learning skills*)

***COURSE CONTENT***

Module I

1. Computational problems and algorithms: asymptotic notation, analysis of time and space complexity;
2. Searching and sorting algorithms;
3. Programming techniques: recursion, divide-et-impera, dynamic programming, backtracking;
4. Data structures: lists, queues, stacks, hash tables, trees, graphs;
5. Selected algorithms on trees and graphs: balanced search trees, graph traversals, minimum spanning tree, shortest path, network flows and cuts;
6. Advanced topics in Python: exceptions, access to file system, comprehensions, generators, lambda functions, decorators

Module II

1. Introduction to neural networks: artificial neuron model, linear networks, activation functions, multilayer perceptron, model selection and assessment, regularization
2. Deep Learning computing: layers, parameters, GPU computing, backpropagation and automatic differentiation, optimization methods, data batches
3. Convolutional Neural Networks: filter kernels, feature maps, convolutional layers, pooling operations, data augmentation, Deep CNN architectures
4. Recurrent Neural Networks: sequence and text representation, hidden states, recurrence, GRU, LSTM, Modern RNNs
5. Applications in computer vision and natural language processing
6. Elements of generative models and reinforcement learning

***READING LIST[[1]](#footnote-1)***

M. T. Goodrich, R. Tamassia, M. H. Goldwasser. *Data Structures and Algorithms in Python*, Wiley, 2013

B. N. Miller, D. L. Ranum. *Problem Solving with Algorithms and Data Structures*, 2nd Edition. Franklin, Beedle & Assoc., 2011. Interactive version freely available at

 <https://runestone.academy/runestone/books/published/pythonds/index.html>

T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*, MIT Press

A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola. *Dive into Deep Learning*, 2020. Freely available at <http://d2l.ai>

I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*, MIT Press, 2016.

Further instructional material, e.g. class notes, source code, handouts, will be posted on Blackboard.

***TEACHING METHOD***

A blend of classical lectures and practice sessions with computer-based activities (60 hours).

***ASSESSMENT METHOD AND CRITERIA***

Written exam part (50% weight) with open-ended questions, closed-ended questions and problems on programming, algorithms and data structures. A practical assessment part consisting of a project for testing student skills in developing and applying deep neural networks for data analytics tasks (50% weight).

***NOTES AND PREREQUISITES***

Students are supposed to be proficient in using a computer and to have mastered the basic notions and skills in Python programming, i.e., the topics covered in the course of “Database systems and computer programming”. Moreover, knowledge of topics in linear algebra, calculus and statistics is required, i.e., the topics covered in the courses “Mathematical methods and probability” and “Statistical inference” taught within this MSc.

1. The texts listed in the bibliography can be purchased from the University bookstores; they can also be purchased from other retailers. [↑](#footnote-ref-1)