# 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, led to the conception 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 at introducing the modern deep neural network models and tools, chiefly convolutional and recurrent architectures, which are widely applied in different AI domains, such as artificial vision and natural language processing.

At the end of the course, the student is 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 a suitable 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 on some task (*Making judgements*)
* describe and justify the 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. Selected topics in Python programming: exceptions, access to file system, comprehensions, generators, lambda functions, decorators
2. Computational problems and algorithms; asymptotic notation, analysis of time and space complexity
3. Data structures: lists, queues, stacks, hash tables, trees, graphs
4. Searching and sorting
5. Programming techniques: recursion, divide-et-impera, dynamic programming, backtracking
6. Selected algorithms on trees and graphs: balanced search trees, graph traversals, minimum spanning tree, shortest path, network flows and cuts

Module II

1. Introduction to neural networks: artificial neuron model, activation functions, linear networks, multilayer perceptron, model selection and assessment, regularization
2. Deep Learning computation: layers, parameters, GPU computing, backpropagation and automatic differentiation, optimization methods
3. Convolutional Neural Networks: filter kernels, feature maps, convolutional layers, pooling operations, Deep CNNs
4. Recurrent Neural Networks: sequence and text models, hidden states, GRU, LSTM, Modern RNNs
5. Applications in computer vision and natural language processing
6. Elements of 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. I testi indicati nella bibliografia sono acquistabili presso le librerie di Ateneo; è possibile acquistarli anche presso altri rivenditori. [↑](#footnote-ref-1)