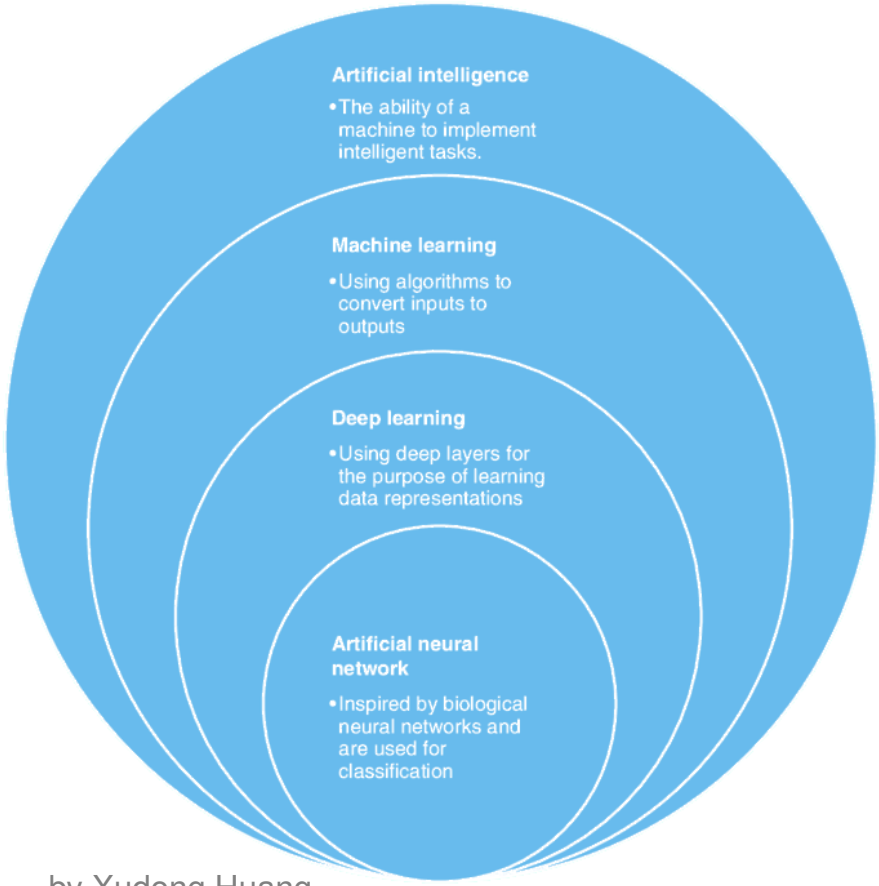
A^3 - Aprendizagem Automática Avançada

Fundamentals of Machine Learning

$\mathbb{G}$. Marques

What is Machine Learning:

Machine Learning is a sub-field of artificial intelligence and is the science that enables computers to learn from the data without being explicitly programmed.



by Xudong Huang

What is Machine Learning:

Machine Learning is a sub-field of artificial intelligence and is the science that enables computers to learn from the data without being explicitly programmed.

In order to build a machine learning algorithm, one needs to define a mathematical model and a performance measure, and use the available data (training data) to adjust the models parameters in order to increase the performance.

The objective is to have a system capable of generalizing on new data: *i.e.* make sufficiently accurate predictions on unseen data.

Deep Learning

Deep learning is a sub-field of machine learning concerning a family of methods based on artificial neural networks. Artificial neural networks have been around since the 80s, but since the last decade, these methods have seen a renewed interest because of their ability to achieve recognition accuracies at higher levels than ever before. This is due to model complexity: traditional neural networks typically have 2 or 3 layers while deep neural networks can have several hundreds.

Machine Learning Approaches

Typically, machine learning approaches are divided into three broad categories, depending on the type of data and labels available.

1. SUPERVISED LEARNING: The training data (input) comes with a desired response (output). The system is "thought" to map inputs to outputs.
2. UNSUPERVISED LEARNING: The system is only provided with the training data without any other information. The goal is to discover hidden patterns in the data or informative, lower-dimensional data representations.
3. REINFORCEMENT LEARNING: The system (agent) takes *actions* within an *environment* based on its *observations*, and in return receives *rewards*. The goal is to maximize the expected rewards over time.

Supervised Learning

Supervised learning means learning from examples. More specifically, a model is trained with labeled data in order to produced a desired response. A supervised learning algorithm analyses all the training data and adapts its model parameters in order to minimize a *loss* measure (or maximize a gain measure). The objective is to construct a function that is able to predict correctly the labels of new input data.

- Bias-Variance trade-off
- Data representation and data dimensionality
- Model parameters (flexibility)
- Evaluation methodologies

Supervised Learning

Two main tasks in supervised learning:

- CLASSIFICATION:
  The task of identifying to which of a set of pre-defined classes an observation belongs to.
- REGRESSION:
  The process of predicting the value of a variable based on a set of observations (independent variables).

In supervised learning the regression and classification models are trained with a set of observations whose class membership or desired values are know. Typically, the construction of supervised algorithms undergoes the following phases.

1. Data collection and preparation.
2. Model selection and implementation.
3. Model training.
4. Model evaluation.

Unsupervised Learning

The goal of unsupervised learning is to detect hidden patterns or manifolds in the data without any human supervision (i.e. no labels are available). Two main areas of unsupervised learning are:

- CLUSTERING:
  Clustering techniques divide the data into groups (clusters) such that data points that belong to the same cluster are more similar to each other (in some sense) than those in other clusters. There are many approaches to clustering such as hard or soft clustering , and non-hierarchical/hierarchical clustering.
- DIMENSIONALITY REDUCTION:
  Dimensionality reduction refers to the mapping of data from a high-dimensional space into a low-dimensional one, ideally without loosing the meaningful properties of the original data. Working with high-dimensional data is often undesirable because analyzing the data can be computationally intractable and because of **the curse of dimensionality**.

Reinforcement Learning

In reinforcement learning, an software agent takes a series of decisions to interact with its environment. The agent either receives a reward or penalty for the actions it performs and the goal is to maximize the expected reward over time.

Reinforcement learning has many areas of application such as game theory, control theory, operations research, information theory, simulation-based optimization, multi-agent systems, swarm intelligence, statistics, and many others.

For more information, check out Richard Sutton and Andrew Barto's book Reinforcement Learning: An Introduction.

Among the many amazing feats achieved with reinforcement learning models, the AlphaGo stands out: this system was able to defeat legendary professional Go players. Check out its story .

Designing ML Systems

In a machine learning project there are several steps that often need to be taken in order to successfully conclude it. Here are the main steps needed to to tackle a machine learning problem:

Project Checklist

1. Understand/frame the problem
2. Get the data and visualize it.
3. Clean the data, and prepare it for ML
4. Pick (several) model(s)
5. Divide dataset into train and test sub-sets (or other divisions)
6. Use the training set to train models
7. Evaluate the models with the test set
8. Fine tune the best models
9. Present results

These steps depend on the problem at hand, and while a few may be skipped, like data cleaning or preparation, others are essential parts of a machine learning project, such as model selection, training and evaluation.

Designing ML Systems

The process of going through the steps of a machine learning project carries some implications.

Implications

1 Preparing the Data

- Most machine learning models work with vectors.
  One needs to convert each observation into a ***d*-dimensional vector**.
- Not a straightforward for many types of data.
  - Text: some text to vector algorithms are available in Python such as tf-idf or text2vect.
  - Audio: work in the time-frequency domain (via spectrograms or other frequency representations), since this representation is more interpretable and less burdensome than its time-domain counterpart.
- Noisy and incorrect data.
  The data can have outliers, missing values and erroneous labels. These"bad points" hinder the performance of machine learning models and it is desirable to clean them, but this process is often time consuming and difficult.
- Data pre-processing
  Normalize the data vectors, or transform the data into lower dimensional spaces via linear or non-linear transformations.

Designing ML Systems

The process of going through the steps of a machine learning project carries some implications.

Implications

2. **Choosing a Model**
   Ideally one should choose a large number candidates in a reasonable amount of time. Here are some guidelines:
   - If the dataset is too large, you may want to sample a smaller one to speed up model training.
   - Choose also simple models. Generally, these are easy to train and can serve as a baseline performance measure.
   - Be aware that some models like deep neural networks are very demanding in terms of computational and memory requirements and penalize the use of advanced testing methodologies such as cross-validation.

Designing ML Systems

The process of going through the steps of a machine learning project carries some implications.

Implications

3. Training and Testing Methodologies
   In supervised learning on has to work with finite data. The goal is to train the model with the available observations so that it can make correct predictions on new ones. One has to make sure that the model is not **over-fitting** the training data. For this purpose, one can split the data into training and test sets, or some other data partition(s) scheme(s).
   - Basic split: train/test sub-sets, or train/validation/test sub-sets.
   - K-fold cross-validation.
   - Shuffle-split strategies.

Designing ML Systems

The process of going through the steps of a machine learning project carries some implications.

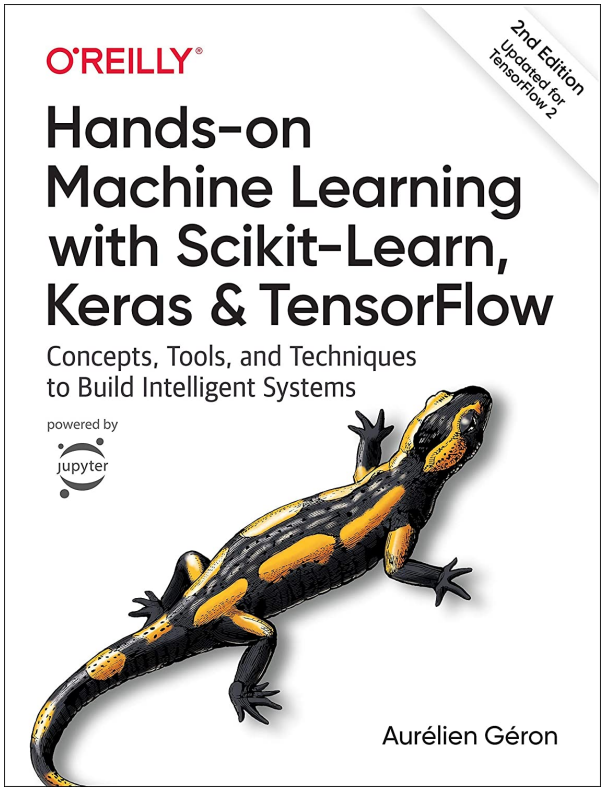
Implications

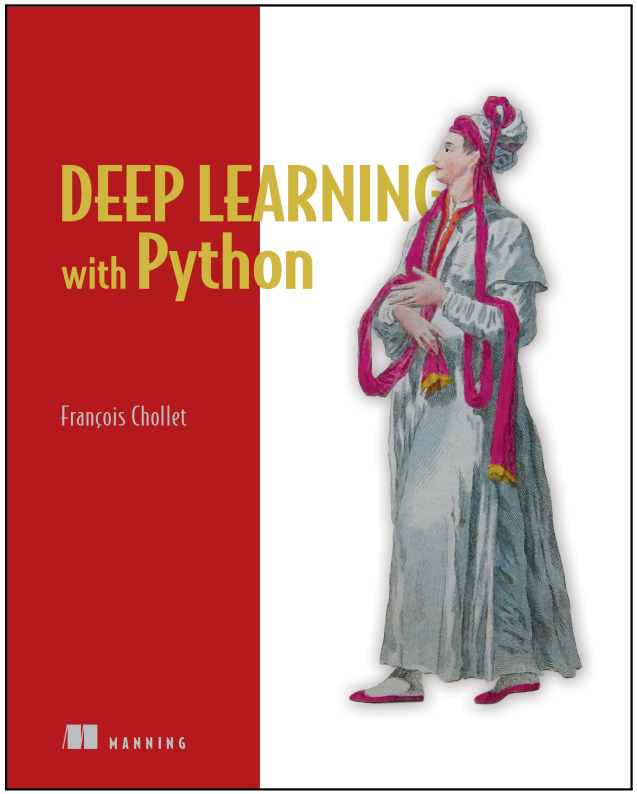
4. **Model Training**
   Model training implies choosing a cost/gain function and minimize/maximize it. The model parameters are optimized in iterative fashion through **gradient descent** methods. In this process, one can include a **regularization** term to avoid over-fitting.
5. **Model Evaluation**
   Performance measures depend on the type of machine learning problem.
   - Multi-Class Classification: confusion matrix, accuracy, ranking methods, ...
   - Binary Classification: precision, recall, f-score, ROC and precision/recall curves, mean average precision...
   - Regression: coefficient of determination R^2, maximum squared error, maximum absolute error, ...
   - Clustering: compactness, distance from other clusters, intra cluster overlap ...

Course Syllabus

1. Classification
   - Classification systems
   - Multi-class evaluation metrics
   - Binary evaluation metrics
2. Classification Project
   - Data pre-processing
   - Cross-validation
   - Model fine-tuning
   - Model calibration
   - Error analysis
3. Log-Linear Classifiers
   - Binary classification
   - Gradient descent
   - Multi-class classification
   - Scikit-Learn implementation
4. Neural Networks
   - Multi-layer perceptron
   - Back-propagation
   - TensorFlow implementation
5. Introduction to CNNs
   - The convolution operation
   - Feature maps
   - Pooling layers
   - TensorFlow implementation
6. Advanced CNN Methods
   - Popular CNN models
   - Transfer learning
   - Data augmentation
   - Model fine-tuning
7. Visualizing CNNs
   - Layers
   - Filters
   - Class activations
8. Object Detection
   - Definition
   - Existing Methods

Main Bibliography

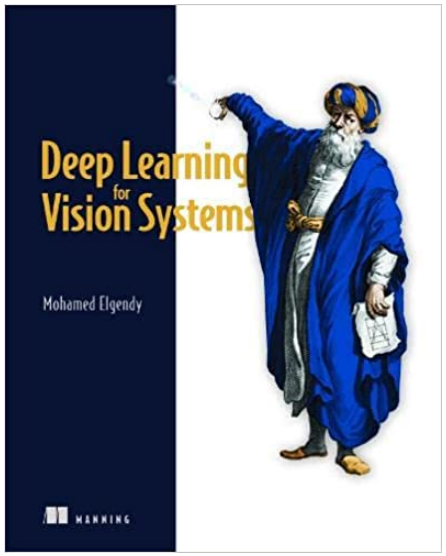


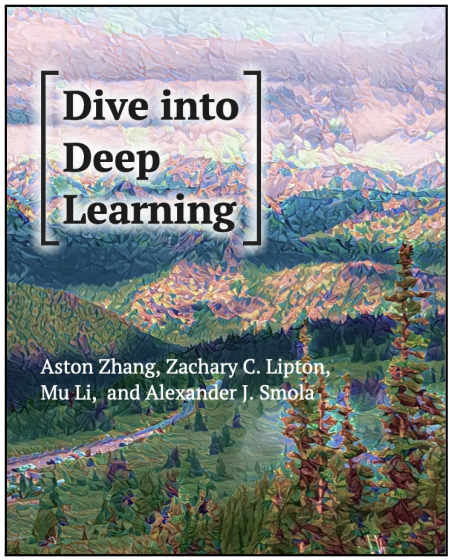


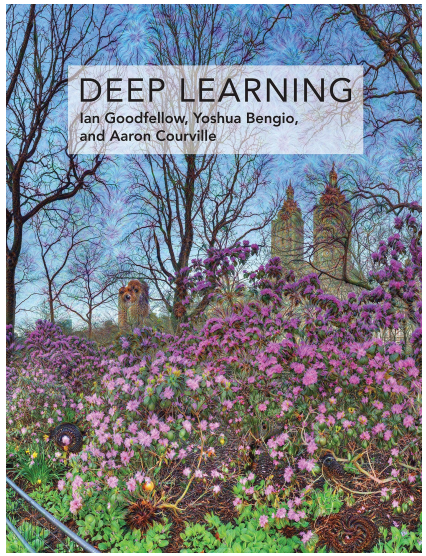
[1] Aurélien Géron. *Hands-On Machine Learning wth Scikit-Learn, Keras and TensorFlow*. O'Reilly, 2019

[2] François Chollet. *Deep Learning with Python*. Manning Pub. Comp., 2017

Complementary Bibliography







[3] Mohamed Elgendy. *Deep Learning for Vision Systems*.
Manning Pub. Comp., 2020

[4] Aston Zhang, Zachary Lipton, Mu Li, and Alexander Smola. *Dive into Deep Learning*.
OnLine, 2022

[5] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*.
MIT Press, 2016

Essential Libraries and Tools

In this course we will assume that you are familiar with Python programming language (version ≥ 3) and its scientific libraries, in particular NumPy, SciPy and Matplotlib. We will also make extensive use of the following Python libraries:

- Scikit-Learn: easy to use, machine learning library, with extensive and well-written documentation and examples.
- TensorFlow: created by Google, this library is for distributed numerical computation and supports many large-scale machine learning applications including several types of deep neural networks.
- Keras: a high-level Deep Learning API that makes training and running neural networks very simple. No need to install it since TensorFlow comes with its own implementation of this API (`tf.keras`).

The code examples given in this course and the assignments will be implemented using a Jupyter Notebook, a web-based interface for running several coding languages.

Course Evaluation

The course consists of the following two components:

1. In-class programming exercises - 30% to 40% of final grade.
   Done individually.
   - Expected number of programming exercises: 4-7.
   - Duration: 1h00mn to 1h30mn.
2. Machine and Deep Learning projects - 60% to 70% of final grade.
   Done in groups of two (2) students.
   - Expected number of projects: 3 or 4.
   - Oral discussion of the projects.

The answers to the in-class exercises and the project reports are to be implemented in a Jupyter-Notebook.