Neural Networks: Course Introduction

Danilo Comminiello and Simone Scardapane

Neural Networks 2023/2024

September 27, 2023

Sapienza
Università di Roma

Table of contents

1 About The Course

Teachers

General Information

Course Description

Expected Results

Syllabus

Class Schedule

Exam Assessment

Your teachers

- Danilo Comminiello
  danilocomminiello.site.uniroma1.it
  danilo.comminiello@uniroma1.it

- Simone Scardapane
  www.sscardapane.it
  simone.scardapane@uniroma1.it

General information

- Credits: 6 CFU
- Course language: English
- Offered programs: Master's Degrees in Artificial Intelligence and Robotics, Engineering in Computer Science, Communication Engineering
- Course webpage: https://sites.google.com/uniroma1.it/neuralnetworks2023/
- Classroom page: Download slides, videolectures, Python notebooks, homeworks and additional material. Access code: **e5t5q7v** . Participants are invited to register here.
- Offices: DIET Dept. (SPV), RM032, 1st floor
- Office hours: by appointment

  Laboratory: ISPAMM Lab, DIET Dept.

ISPAMM Research Lab

Figure 1: The **Intelligent Signal Processing and MultiMedia (ISPAMM)** group is an interdisciplinary research team at DIET. Website: www.ispamm.it

Course description

The course is intended as a broad overview to neural networks, as used today in a number of applicative fields.

It provides both theoretical and practical understanding of how neural networks are designed and implemented.

We will review the general paradigm of building differentiable models.

We will overview essential components to deal with images (*convolutive layers*), sequences (*recurrent layers*), and sets (*transformer layers*).

We will finally focus on a selection of advanced and recent topics, including *graph neural networks*, *continual learning*, and *generative models*.

Expected results

At the end of the course, a student is expected to have:

- a broad understanding of how neural networks work in practice,
- the capability of implementing new components from scratch,
- the capability of re-using existing models,
- the capability of designing new architectures for problems beyond the overview of the course,
- the ability to autonomously study new topics on the research frontier, and navigate the current scientific literature and software panorama.

What we expect from you

- Ask yourself what are your main interests in Neural Networks.
- Consider the practical part of this course as an opportunity for you to understand your attitudes.
- Attend classes, as it is the first step to become an AI master!

Prerequisites and background

This course requires basic knowledge of:

- machine learning fundamentals
- linear algebra
- probability theory and stochastic processes
- optimization
- Python

Anyway, sufficient recalls will be given and additional material can be provided, even on request, to fill some gaps.

Syllabus I

- **Introduction and Background**
  - Introduction to neural networks
  - Background recalls on linear algebra, probability and optimization
- **Neural Networks Fundamentals**
  - Linear regression and automatic differentiation
  - Multi-layer perceptron
- **Neural Networks for Images**
  - Deep learning computation
  - Convolutional neural networks and modern architectures

Syllabus II

- **Neural Networks for Sequences and Sets**
  - Recurrent neural networks and modern architectures
- **Transformer Network**
  - Attention mechanisms
  - Transformer architectures
- **Deep Generative Models**
  - Generative Adversarial Networks
  - Deep Latent Generative Models
- **Emerging Neural Approaches**

Course material

General information:

- On the NN course page you can find all the course information, including program and exam dates.

Main material:

- Lecture slides and video recordings, papers, notebooks, supplementary material, and news will be provided on the Classroom page (access code: e5t5q7v).

Main textbooks (available online):

- Simon J. D. Prince, “Understanding Deep Learning”, MIT Press, 2023.
- Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola, “Dive Into Deep Learning”, 2020.

Class schedule

The course starts on September 27, 2023, with the following schedule:

- Wednesday, 14:00–16:00, Classroom Viale Regina Elena, 295
- Thursday, 15:00–19:00, Classroom 21, SPV, Via Eudossiana, 18

The course is held in in-presence modality only.

Exam assessment and grade evaluation

- **Final project: 60% of the grade**
  - Projects on emerging and advanced topics can be assigned starting from the beginning of December.
  - You can choose one project among those available.
  - Teams are possible up to 2 students.
  - Projects must be submitted at least 1 week before the exam date.
- **Questions: 40% of the grade**
  - Oral examination in scheduled exam dates.
  - Oral examination consists in two questions on the course program.

Lode is given only to students presenting an outstanding project (e.g., *publication quality*) and giving a brilliant oral argumentation.

Exam schedule

- **I Session:** January TBD, 2024
- **II Session:** February TBD, 2024
- **I Extraordinary Session:** March TBD, 2024
- **III Session:** June TBD, 2024
- **IV Session:** July TBD, 2024
- **V Session:** September TBD, 2024
- **II Extraordinary Session:** October TBD, 2024

2 What are neural networks?

Neural networks are (almost) everywhere

Defining a neural network

Let's start from a definition

Neural networks are

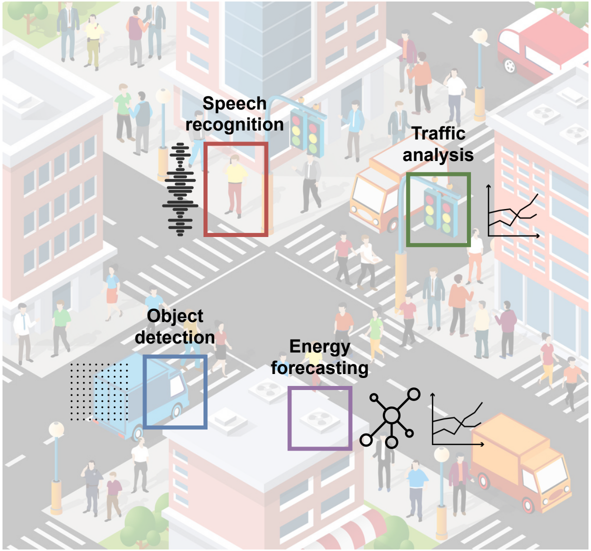
composable,

differentiable functions

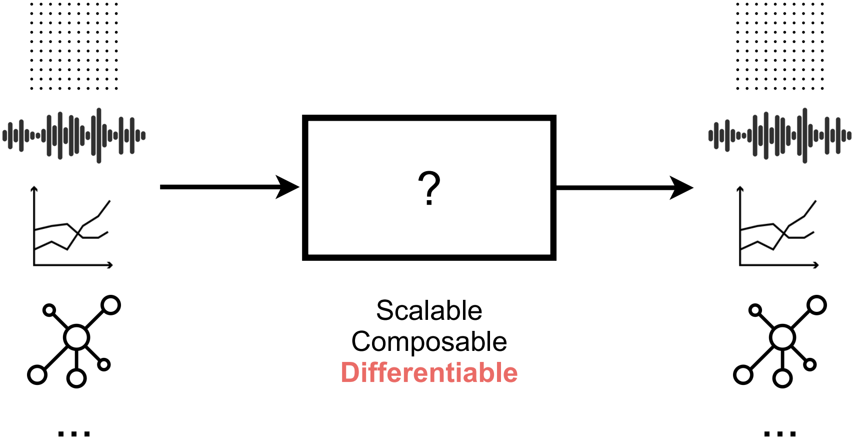
that can be **optimized end-to-end** numerically.

Neural networks are everywhere I

Neural networks are everywhere II



Processing complex data



Neural networks in five minutes

- All these inputs/outputs can be represented as tensors, i.e., large n-dimensional arrays of numbers.
- Neural networks are composed of multiple blocks (layers), each of which performs a simple manipulation on these tensors.
- The operation of a layer may involve another tensor, whose values can be chosen freely (e.g., a matrix multiplication). These are called parameters of the layer.
- All parameters can be optimized numerically (training) by maximizing the performance of the network on a set of examples (dataset).

“State-of-the-art”

Listing all notable applications of neural networks is almost impossible: think of a complex problem, and someone has probably developed a state-of-the-art model for it, ranging from neural translation to protein folding, videogame playing, neural rendering, physics simulations,...

Hint: browse paperswithcode.com/sota for a few examples.

Amazingly, all this is powered by a very small set of components and organizing principles (e.g., *differentiability*, *invariances and equivariances*, *sparsity*, *locality*).

3 Neural Network Applications

- Audio and Speech Application
- Computer Vision
- Smart Manufacturing
- AI Accelerators
- Safety and Security
- Neural Networks for Biomedical Applications
- Neural Networks for Science

Speech recognition

Automatic Speech Recognition is one of the most popular tasks involving devices equipped with microphone sensors.

Figure 2: Typical examples of home and robot assistants. Image sources: Cosmos Magazine, Ears.

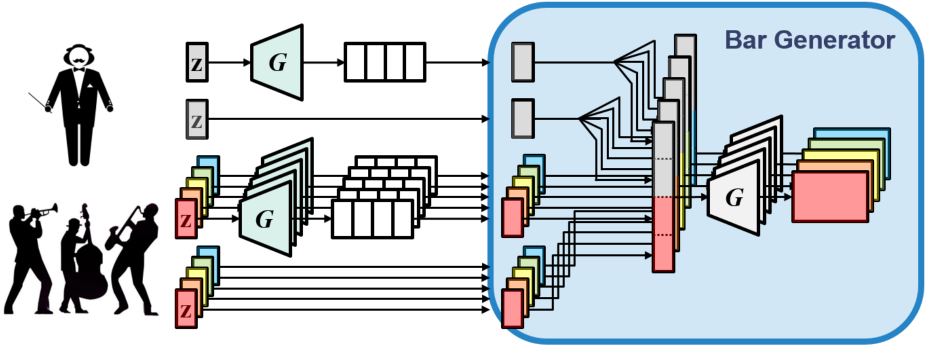


Figure 3: Neural networks are able to synthesize music. Image sources: MuseGAN.

Machines that dream

Learning your favorite art and applying it to repaint pictures.

Figure 4: Image source: DeepArt.

Generate data that do not exist

Advanced algorithms are trained to generate and/or detect text and news, and even audio, images or videos.

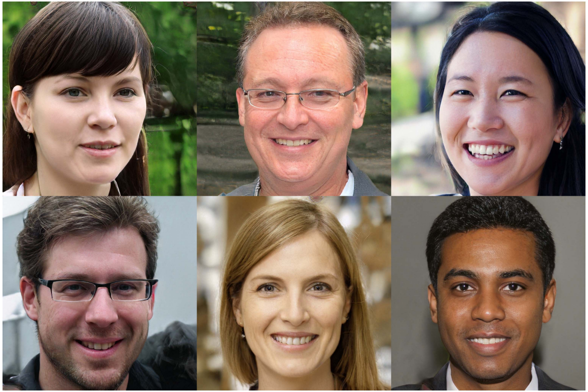


Figure 5: Realistic fake faces can be generated by deep generative models. Image source: ThisPersonDoesNotExist.

Self-driving cars involve several potential applications, such as:

- evaluation of driver condition, driving scenario classification, use of external and internal sensors, infotainment systems, speech and gesture recognition, among others.

Figure 6: Image source: Scientifist.com.

Smart manufacturing and IoT applications

Most companies are shifting towards automate manufacturing to increase productivity, minimize human errors, optimize production costs, improve efficiency.

This process involves Internet-of-Things paradigm and a wide range of sensors to capture signals to be processed by ML algorithms.

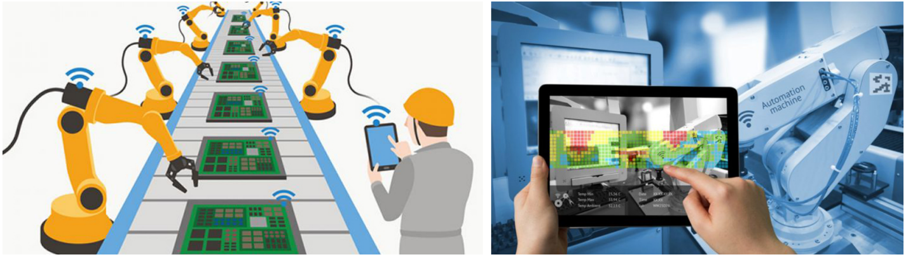


Figure 7: Image sources: MachineDesing, PacktHub.

AI accelerators

Modern artificial intelligence (AI) applications involve neural networks that require efficient microprocessors or computer systems designed as **hardware accelerator**.

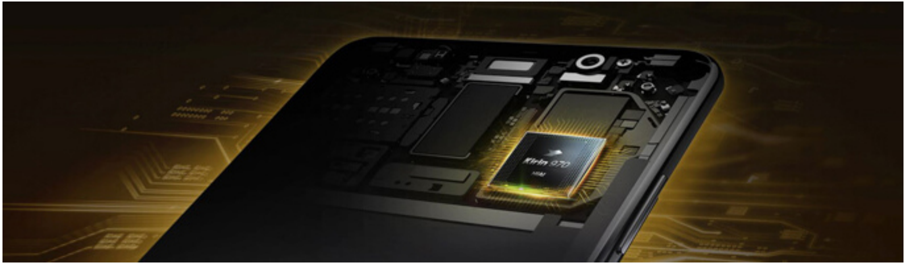


Figure 8: Most recent smartphones are provided by Neural Processing Unit. Image source: Huawei.

Crowd behavior analysis: detect abnormal events in crowded places, such as panic and acts of violence, from video sequences.

Figure 9: Image source: Active Vision Laboratory.

Neural networks can be applied to **biomedical signals** for several goals, such as:

- improve prediction of symptoms, procedures, diagnosis, medications or patient outcomes, learning from missing or imbalanced data, analysis of biomedical time series, scalable processing of clinical data, processing and making sense of clinical notes, among others.

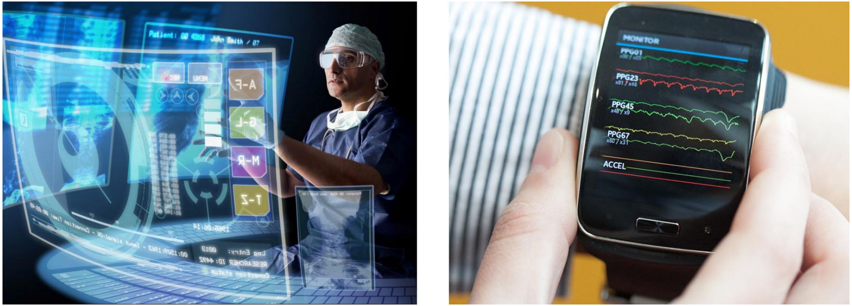


Figure 10: Image source: The Telegraph.

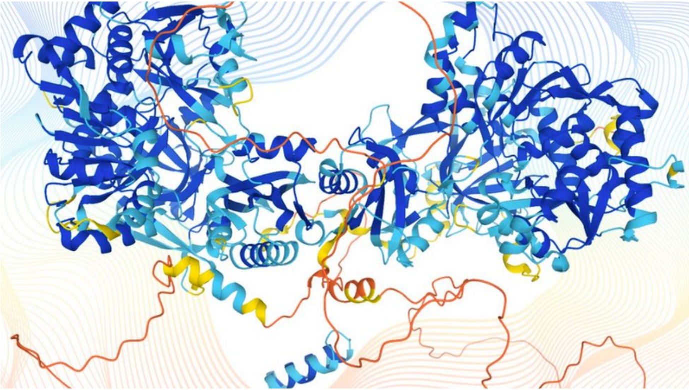


Figure 11: Neural networks empower scientific discovery, from personalized drugs to nuclear fusion control. Image source: BBC.

4 Taxonomy of Learning Approaches

- Learning Approaches
- Supervised Learning
- Unsupervised Learning
- Semi-Supervised Learning
- Self-Supervised Learning
- Reinforcement Learning
- Main Learning Tasks

Introduction to Learning Approaches I

Machine learning approaches can be grouped in four big families:

1. Supervised learning is an approach based on data learning.
2. Unsupervised learning is an approach aiming at clustering data even if no example is available. Self-supervised learning is an unsupervised approach that allows of implementing supervised learning tasks.
3. Semi-supervised learning is based on the availability of some examples, which are not enough to learn the model completely.
4. In reinforcement learning, the optimal learning process aims at maximizing a certain *reward* associated to each action.

Introduction to Learning Approaches II

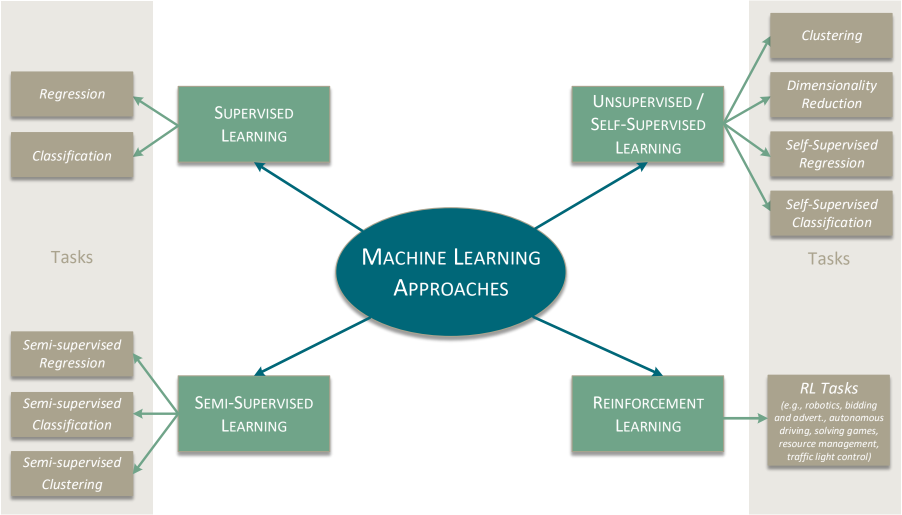


Figure 12: Representation of a taxonomy of machine learning approaches and tasks.

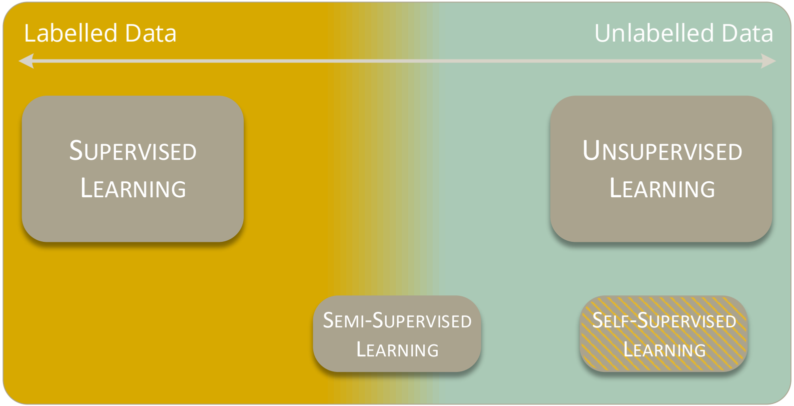


Figure 13: Representation of a taxonomy of machine learning approaches according to the availability of labelled or unlabelled data.

Supervised learning I

Supervised learning (SL) approach is based on the learning by data examples.

The examples, also denoted as *training data*, can be seen as the available previous experience, and based on this, one builds a model to make predictions for the future.

The main tasks in supervised learning are *classification* and *regression*, which will be extensively studied during the course.

In SL, or *learning with a teacher*, the network is provided with a correct answer (the output) for every input.

Supervised learning II

A concise description of the data is provided by the supervisor function that produces the output, given an input.

The supervisor function can be defined as a target signal $\mathbf{y}_k$ containing the desired output for a fixed set of inputs $\mathbf{x}_k$.

For this purpose, we define as *training set* a given a set of empirical data defined over input/output space $\mathcal{T} \subset (X, Y)$, formally defined as

$$[\mathbf{x}_k, \mathbf{y}_k]_{k=1}^{K} \in \mathcal{T}\,(\subset X, Y)\,, \quad \mathbf{x} \in X \subset \mathbb{R}^P, \quad \mathbf{y} \in Y \subset \mathbb{R}^Q$$

where X, Y are, respectively, the input and output vector spaces.

The goal of the SL is to find a suitable function that maps any input to an output space, in which a certain task can be accomplished.

This process is subject to the minimization of a certain error.

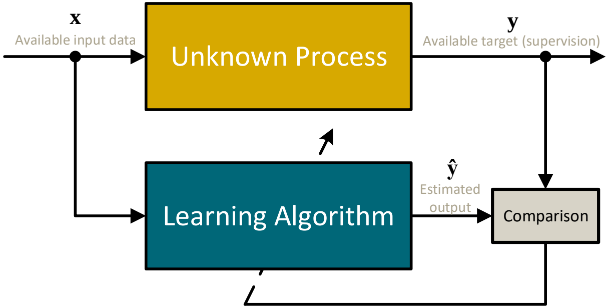


Figure 14: General scheme for supervised learning.

Unsupervised learning I

Unsupervised learning (UL) aims at recovering the clusters in which the available data can be grouped together without relying on the availability of any previous example.

In UL, or *learning without a teacher*, a correct answer associated to each input in the training set is not required.

UL explores the underlying *structure* in the data, or correlations between patterns in the data, and organizes input into categories from these correlations.

A general way to represent data is to specify a similarity between any pairs of objects. If they share any structure, it should be possible to reproduce the data from the same prototype.

Unsupervised learning II

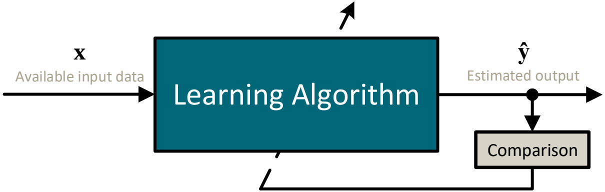


Figure 15: General scheme for unsupervised learning.

The main UL tasks are *clustering* and *dimensionality reduction*. Typical examples include *image and text segmentation* and *novelty detection* in process control.

Semi-supervised learning

Semi-supervised learning is a hybrid approach that combines supervised and unsupervised learning.

Input data is a mixture of labeled (i.e., *known*) examples (usually small quantity) and unlabeled examples (larger quantity).

SSL may also be referred to as either:

- transductive learning, whose goal is to infer the correct labels for the given unlabeled data,
- inductive learning, whose goal is to infer the correct mapping from X to Y.

Self-supervised learning

Self-supervised learning is not really a hybrid approach, like semi-supervised learning, but it can be considered as an *unsupervised learning approach* as no labels were given.

However, unlike unsupervised learning, it does not aim at clustering high-level patterns.

Indeed, self-supervised learning attempts to still solve tasks in a similar way to supervised learning, but without any label available.

This is possible by augmenting the input data using transformed versions of them.

This enables to learn group of similar classes, which make possible tasks that originally required fixed labels to learn, such as classification, without any ground truth provided.

Reinforcement learning I

The term **reinforcement learning** (RL) refers more to a philosophy for learning algorithms development, which finds inspiration from behaviorist psychology, rather than to a specific type of algorithm.

RL has its roots in *control theory*.

Indeed, the typical RL scenario involves a *dynamic environment* that results in state-action-reward triples as the data.

The learning algorithm is intended as an *agent* working in a certain *environment*, which can take actions within it.

Reinforcement learning II

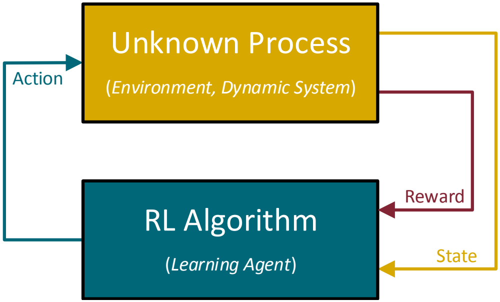


Figure 16: General scheme for reinforcement learning.

For every action, which is *randomly* performed, a certain reward can be related.

Randomness may be introduced either as noise in *measurements* or Monte Carlo randomness in the *search procedure*, or both.

Reinforcement learning III

The difference between *RL* and *SL* is that in RL no optimal action exists in a given state, but the learning algorithm must identify an action in order to *maximize the expected reward* over time.

Unlike SL, the learning algorithm in RL is not told which actions to take in a given situation.

Instead, the learner is assumed to gain information about the actions taken by some reward not necessarily arriving immediately after the action is taken.

The *concise description of data* is the strategy that maximizes the reward.

An example of RL problem is *learning to play chess*.

Reinforcement learning IV

One of the biggest *challenges* of an RL algorithm is to find a trade-off between exploration and exploitation:

- To maximize the reward, a learning algorithm must choose actions that have found to be *effective in the past* in producing a reward.
- To discover those actions the learning algorithm has to choose actions that was *not tried in the past*, thus exploring the state space.

There is *no general solution* to this dilemma, but neither of the two options can lead exclusively to an optimal strategy.

RL is studied and applied in many disciplines, such as *game theory*, *stochastic/heuristic optimization algorithms* involving random search and optimization, among others.

Main learning tasks

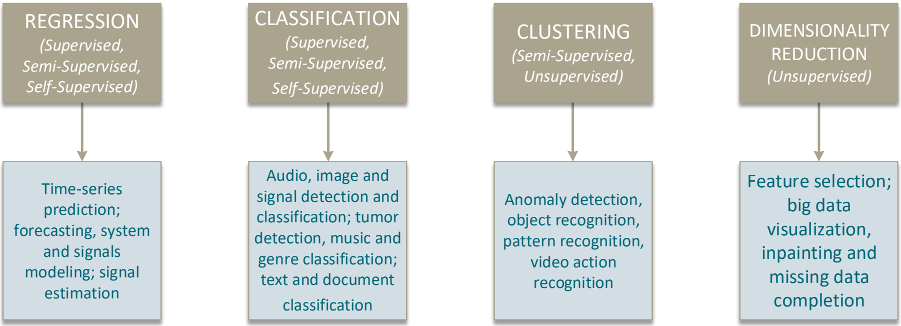


Figure 17: In this course, we focus on both supervised and unsupervised tasks. Each task has a variety of applications that can be found in the real world.

Next lecture

- Linear algebra refresh.
  - The notions of linear algebra that can be considered most relevant to neural networks will be introduced.
  - Basic operations will be shown to process and manipulate structured data.

Neural Networks: Course Introduction

Neural Networks 2023/2024

Danilo Comminiello and Simone Scardapane

https://sites.google.com/uniroma1.it/neuralnetworks2023

{danilo.comminiello, simone.scardapane}@uniroma1.it