

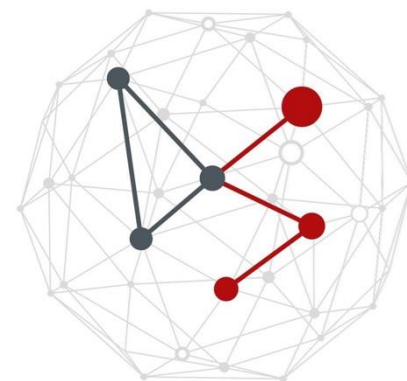
ML FOR HUMAN DATA ANALYTICS: A.Y. 24/25

Michele Rossi

michele.rossi@unipd.it

Francesca Meneghello (labs)

francesca.meneghello.1@unipd.it



ML4HDA

- A course on **Applied ML for Human Sensing Systems**
 - Centered around human-generated signals
- **Master Degrees**
 - Data science (DM)
 - ICT for Internet and Multimedia (DEI)
- **Teacher**
 - Prof. Michele Rossi - theory & apps
 - <https://www.dei.unipd.it/~rossi/>

Michele Rossi



Lab. classes

- **Lab** classes (14 hours)
 - Will be held *in presence*
 - **how-to:**
 - install & use environments
- Lab. classes for self study
 - Preliminary material 1/2 videos
+ supporting material
 - Advanced Python machine learning
 - Deep learning

Francesca
Meneghello



Eleonora
Cicciarella

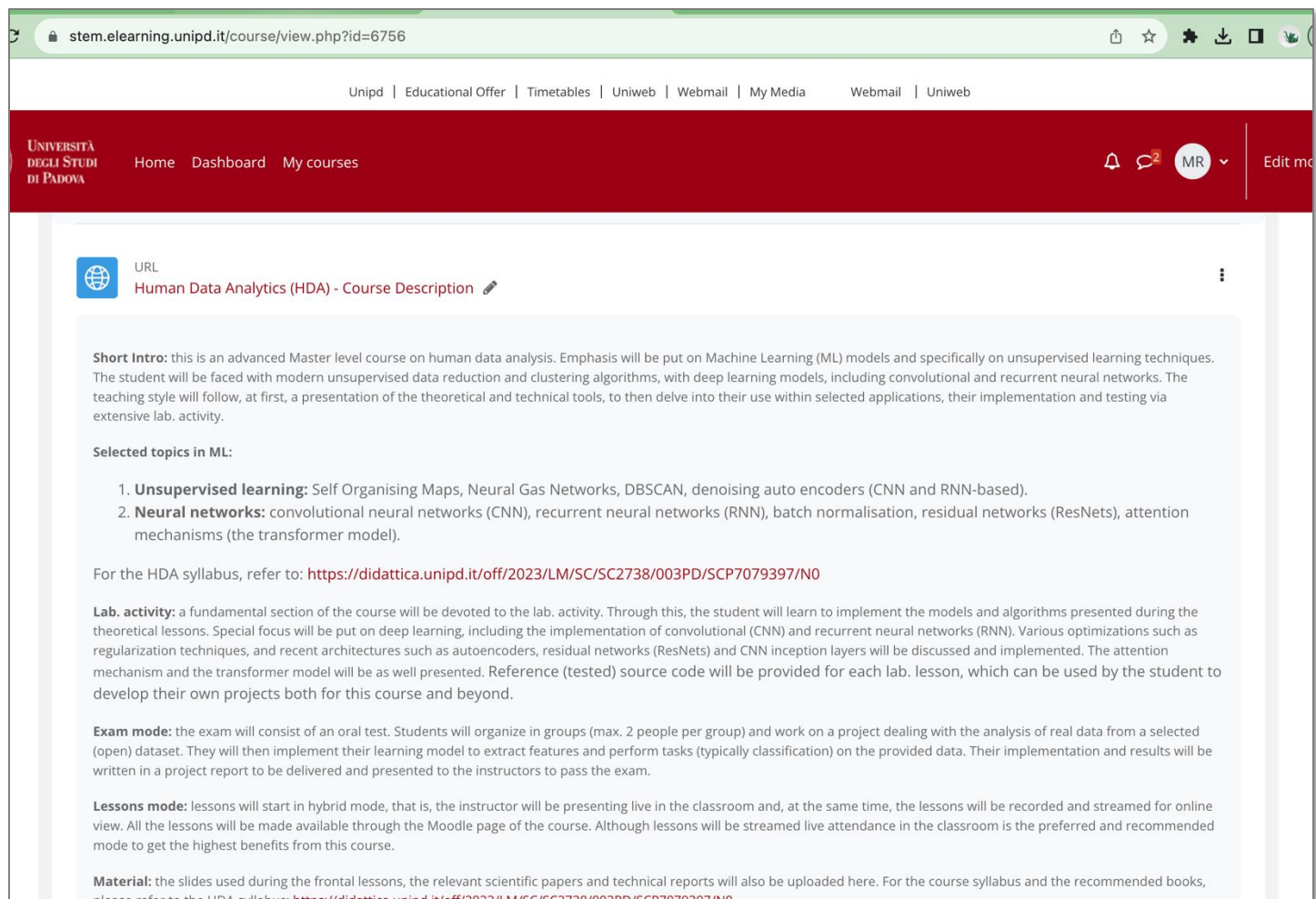


Course material

- Course material
 - ML for HUMAN DATA ANALYTICS 2024/2025
 - Integrated Moodle platform
 - <https://stem.elearning.unipd.it/>
 - Slides
 - Video lessons (**no videos** for the labs)
 - Lab classes (14 hours)
 - preliminary material
 - videos of assignments
 - software with solutions

<https://stem.elearning.unipd.it/course/view.php?id=9864>

ML for HUMAN DATA ANALYTICS 2024-2025



The screenshot shows a web browser displaying the course description for 'Human Data Analytics (HDA) - Course Description' on the Unipd e-learning platform. The URL in the address bar is <https://stem.elearning.unipd.it/course/view.php?id=6756>. The page header includes navigation links for Unipd, Educational Offer, Timetables, Uniweb, Webmail, My Media, and another Webmail/Uniweb link. The University of Padua logo is visible on the left, and a user profile 'MR' is on the right. The main content area has a title 'URL Human Data Analytics (HDA) - Course Description' with a globe icon. The description includes a 'Short Intro' about the course's focus on Machine Learning (ML) models and unsupervised learning techniques. It lists 'Selected topics in ML' with two main points: '1. Unsupervised learning' (covering Self Organising Maps, Neural Gas Networks, DBSCAN, and denoising auto encoders) and '2. Neural networks' (covering CNN, RNN, batch normalisation, ResNets, and the transformer model). It also provides a link to the HDA syllabus. The 'Lab. activity' section describes a fundamental section devoted to lab work, and the 'Exam mode' section describes an oral test and project-based assessment. The 'Lessons mode' section describes a hybrid delivery format. The 'Material' section mentions that slides and technical reports will be uploaded.

stem.elearning.unipd.it/course/view.php?id=6756

Unipd | Educational Offer | Timetables | Uniweb | Webmail | My Media | Webmail | Uniweb

UNIVERSITÀ DEGLI STUDI DI PADOVA

Home Dashboard My courses

MR

Edit mo

URL

Human Data Analytics (HDA) - Course Description

Short Intro: this is an advanced Master level course on human data analysis. Emphasis will be put on Machine Learning (ML) models and specifically on unsupervised learning techniques. The student will be faced with modern unsupervised data reduction and clustering algorithms, with deep learning models, including convolutional and recurrent neural networks. The teaching style will follow, at first, a presentation of the theoretical and technical tools, to then delve into their use within selected applications, their implementation and testing via extensive lab. activity.

Selected topics in ML:

1. **Unsupervised learning:** Self Organising Maps, Neural Gas Networks, DBSCAN, denoising auto encoders (CNN and RNN-based).
2. **Neural networks:** convolutional neural networks (CNN), recurrent neural networks (RNN), batch normalisation, residual networks (ResNets), attention mechanisms (the transformer model).

For the HDA syllabus, refer to: <https://didattica.unipd.it/off/2023/LM/SC/SC2738/003PD/SCP7079397/N0>

Lab. activity: a fundamental section of the course will be devoted to the lab. activity. Through this, the student will learn to implement the models and algorithms presented during the theoretical lessons. Special focus will be put on deep learning, including the implementation of convolutional (CNN) and recurrent neural networks (RNN). Various optimizations such as regularization techniques, and recent architectures such as autoencoders, residual networks (ResNets) and CNN inception layers will be discussed and implemented. The attention mechanism and the transformer model will be as well presented. Reference (tested) source code will be provided for each lab. lesson, which can be used by the student to develop their own projects both for this course and beyond.

Exam mode: the exam will consist of an oral test. Students will organize in groups (max. 2 people per group) and work on a project dealing with the analysis of real data from a selected (open) dataset. They will then implement their learning model to extract features and perform tasks (typically classification) on the provided data. Their implementation and results will be written in a project report to be delivered and presented to the instructors to pass the exam.

Lessons mode: lessons will start in hybrid mode, that is, the instructor will be presenting live in the classroom and, at the same time, the lessons will be recorded and streamed for online view. All the lessons will be made available through the Moodle page of the course. Although lessons will be streamed live attendance in the classroom is the preferred and recommended mode to get the highest benefits from this course.

Material: the slides used during the frontal lessons, the relevant scientific papers and technical reports will also be uploaded here. For the course syllabus and the recommended books,

Teaching approach

- Teaching mode

- All the theoretical lessons will be live in the classroom
- Recorded for online view
 - Posted later for offline view/study

- Lab activity

- In presence, in the lab
- Lab experiences
 - will not be recorded
 - will not be streamed
- Project assignments, software & solutions
 - will be posted for offline use

HDA Selling Points (1/2)

- Unsupervised learning techniques
 - Clustering, classification
 - Seldom studied in MS level courses
 - Very important for applications
 - Will be key to future research/developments
- Neural networks
 - Theoretical lessons (architectures, learning)
 - Lab. activity + hands-on activity
 - Set up and use environments (mainly Python, TensorFlow)
 - Learn to code your neural network (CNN, RNN, etc.)
 - Case studies based on real datasets

HDA Selling Points (2/2)

- Usually
 - Focus of basic machine learning courses is on i.i.d. data samples
 - Problems are typically: regression, classification
- This course
 - Is often concerned about modeling **complex data sequences**
 - Some (spatio-temporal) correlation exists among data points
- **Our focus is on (temporal/spatial) pattern analysis**
 - For correlated data (space, time)
 - Often such data is generated by sensing applications

Exams (1/2)

- **Project based**
 - Project assignment from instructors
 - Design/implement a machine learning model
 - Testing it on a public dataset
 - Testing it on own-collected data from wearable sensors
- **Outcome**
 - Written project report (max. 15 pages)
 - Python implementation of the processing pipeline
 - Power Point presentation (small demo at exam is appreciated)
- **Groups**
 - Max. 2 students per group

Guidelines about: public database, task to be performed and project report structure will be provided when we start the laboratory activity

Exams (2/2)

Exam dates for a.y. 24/25

- 28-29 January 2025
- 18-19 February 2025
- 18-19 June 2025
- 2-3 July 2025
- 18-19 Sept 2025

Enrolment on an exam

- <https://uniweb.unipd.it/>

Exam mode

- Upload technical report in pdf (template will be provided)
- Upload your code in advance
- Present your work using slides (20 minutes)

TOPICS

ML4HDA in a nutshell

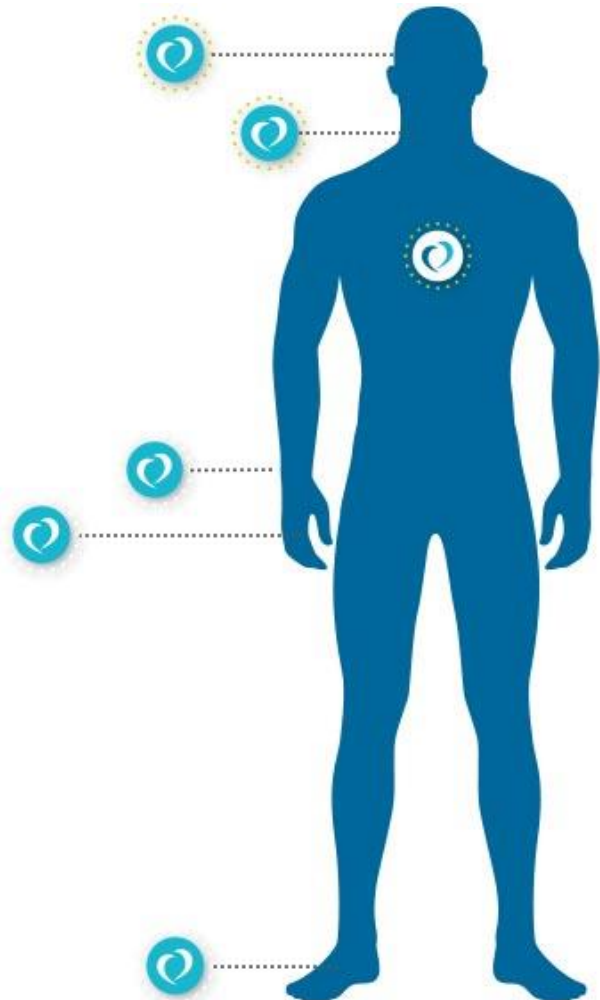
- Tools

- Dimensionality reduction: PCA
- Unsupervised Clustering: K-means, SOM, GNG, DB-SCAN
- Neural networks:
 - Feed Forward (FFNN), Convolutional (CNN),
 - Recurrent Neural Networks (RNN), Autoencoders, Spiking Nets
 - Implementation tricks: batch normalization, dropout, inception layers, attention mechanism, autoencoders
- Times series analysis: RNN, RNN+attention, transformer model

- Applications

- ECG signals, pulse oximeter (blood oxygenation), speech analysis
- Inertial signals (motion analysis)
 - authentication, activity recognition (heterogeneous data)
- Medical images

Human sensing



Activity Trackers



Smart watches



Smartphones



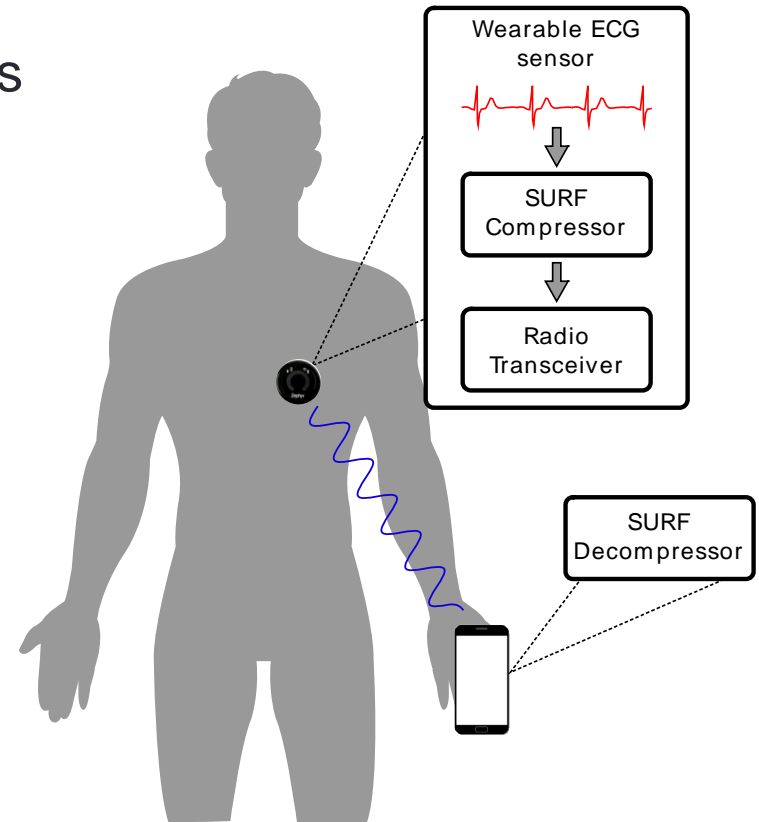
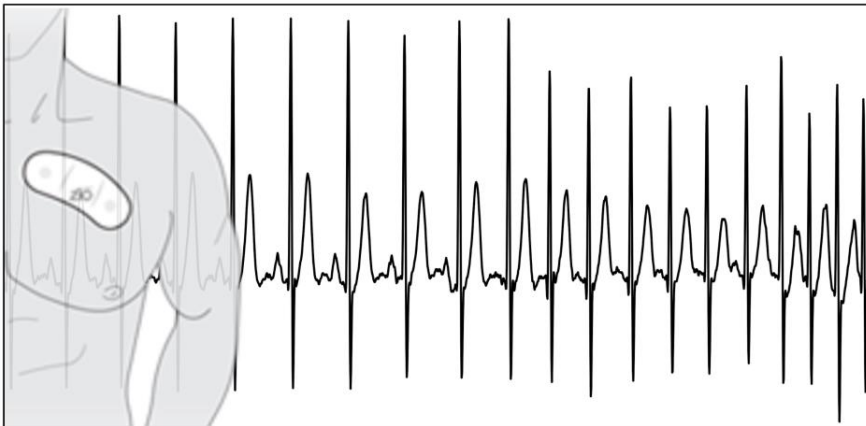
Interactive shoes



Smart clothing

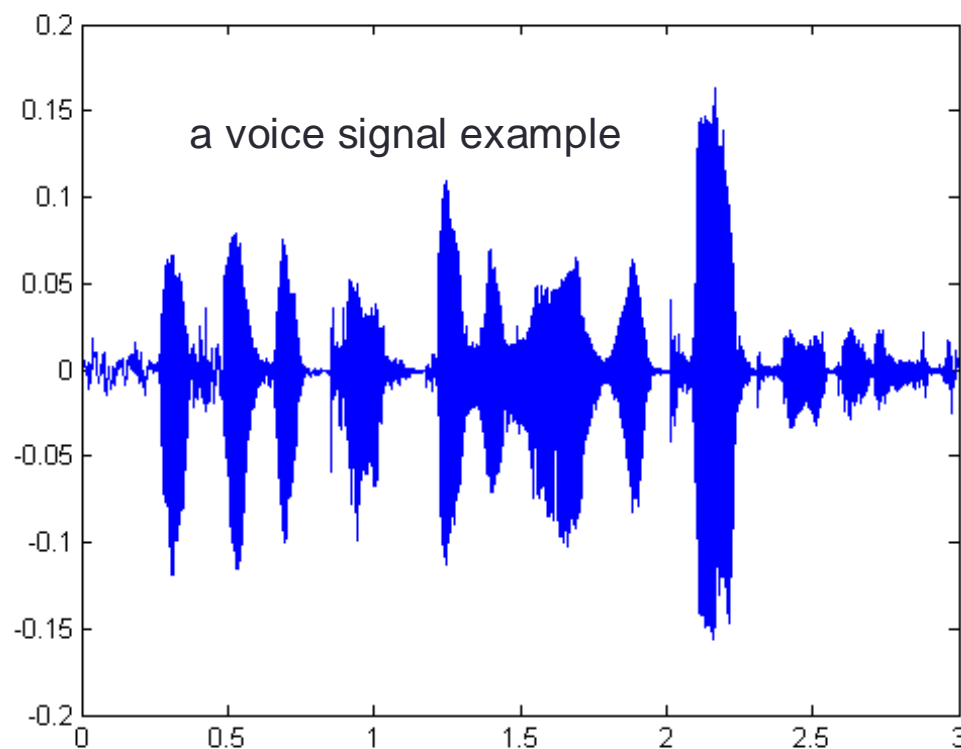
Modeling ECG signals

- Useful for many reasons
 - Efficient TX in resource limited systems
 - Automated detection of arrhythmia
 - User identification / authentication



Statistical modeling of time series

- How to reliably decode words and sentences (speech)
 - CNN, RNN, attention mechanism



Automatic speech recognition

- How to decode voice

- Feature extraction
- Cepstral features

- The tools

- CNN, RNN
- Attention mechanism



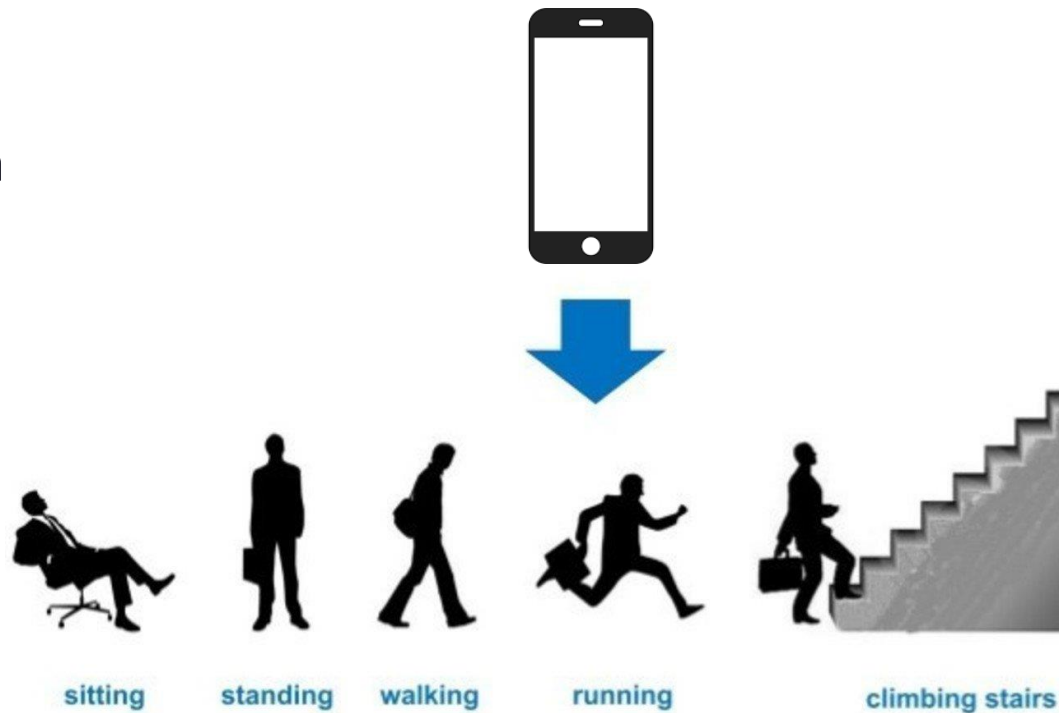
Inertial signals

- Accelerometer and Gyroscope signals

- From Inertial Measurement Units (IMU)
 - Chest-band, wrist-band, smart watches, smart phones, etc.

- Applications

- User authentication
- Activity recognition



Course structure

1) Tools: dimensionality reduction & clustering

Apps: Electrocardiography (ECG) signal

2) Tools: Neural networks (CNN, RNN, autoencoders, residual networks, attention mechanism/transformers)

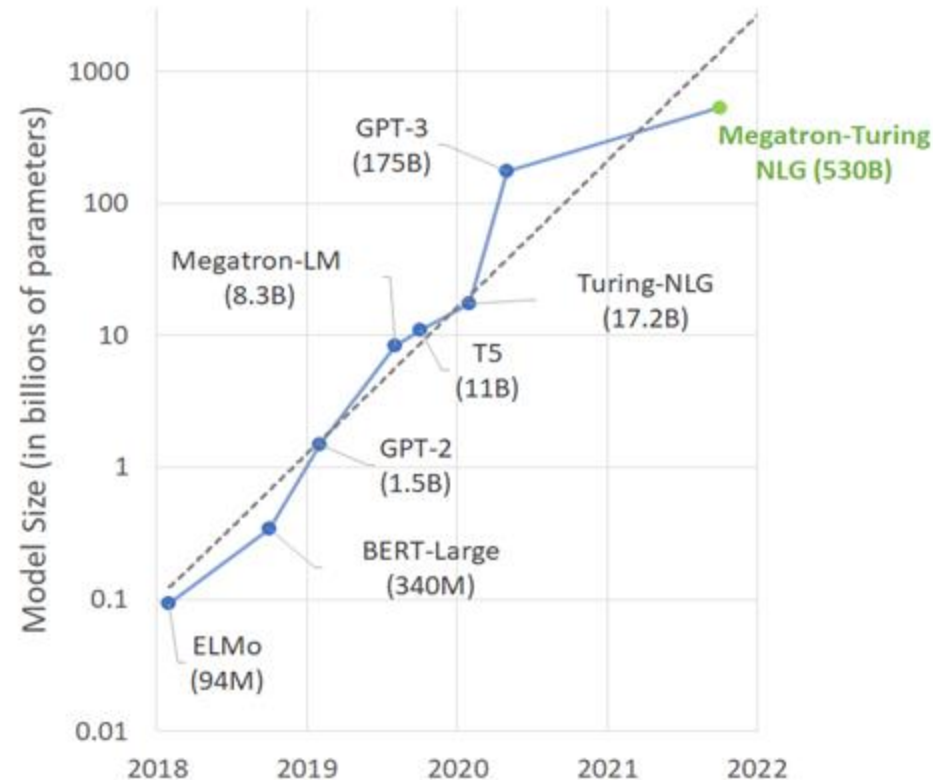
Apps:

- ECG signal
- speech analysis
- inertial signal (motion analysis)
- medical images: lymphoma classification

Spiking Neural Networks: Motivations

Deep Learning models are getting bigger and bigger

- Long training times
- Huge requirements in terms of energy consumption
- Implementation on energy-constrained devices is infeasible
- Need for efficient DL architectures



Spiking Neural Networks: Motivations

GPT-3, contains 175 billion learnable parameters, estimated to consume roughly 190,000 kWh to train (12 Million USD\$ in energy bills)

Imagine we wanted to train the brain as an ANN:

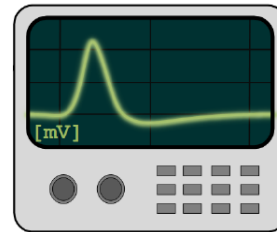
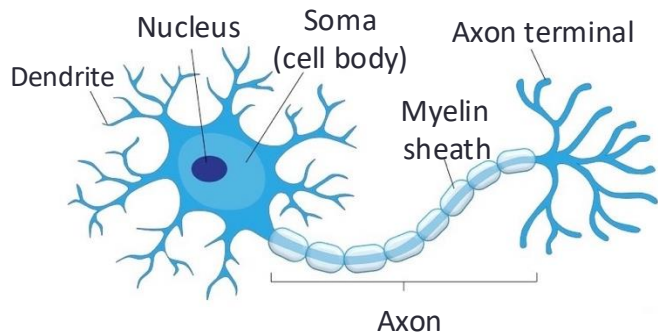
Simon Thorpe, Embedded Systems Research Group 2021

$$\underset{\text{neurons}}{8.6^{10}} \cdot \underset{\text{synapses}}{7.0^3} \cdot \underset{\substack{\text{time resolution} \\ (\text{ms})}}{1.0^3} = 600 \text{ PetaFLOPS}$$

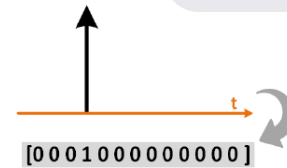
- Most powerful existing supercomputer: 442 PetaFLOPS at **30MW**
- Meanwhile, our brains operate within **~12-20 W** of power

Can we take inspiration from the brain?

Spiking Neural Networks



≈



Spikes are modeled
as **all-or-nothing**
event

- Biological neurons communicate via **action potentials**, or *spikes*
- The form of the action potential does not carry any information
- It is the **number** of spikes and their **timing** which matter
- Event-based processing

LABORATORIES

Lab. Classes

- Will use several tools



- Guided coding sessions to build machine learning apps
 - Build everything from scratch (simple applications)
 - Use some popular libraries (more complex projects)



TensorFlow



Jupyter

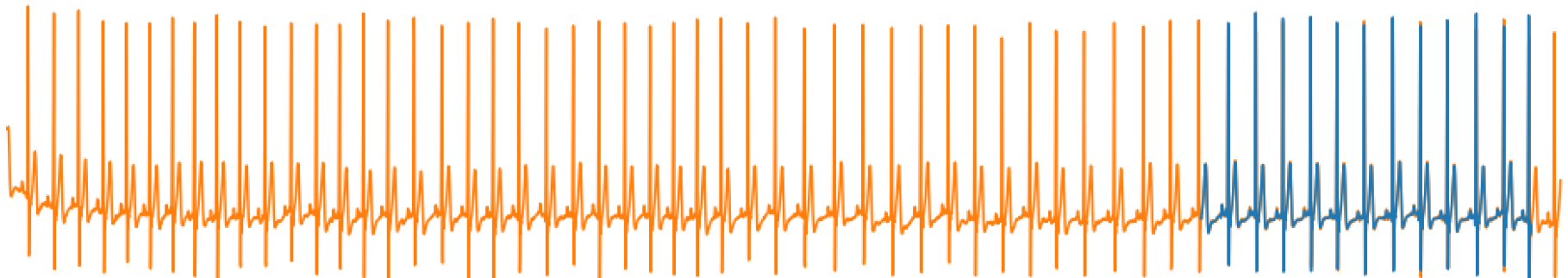


Lab 0 – preparatory material

- For self-study
 - Guided coding sessions to build machine learning apps
 - You will learn to:
 - use Jupyter notebooks and JupyterLab
 - use math and NumPy Python libraries
 - define your own functions in Python

Lab 1 – Oct. 29

- Machine learning tools for dimensionality reduction: PCA and clustering
 - The challenge:
 - ECG and PPG signals dimensionality reduction
 - You will learn to:
 - implement a dimensionality reduction algorithm based on PCA from sketch and using the implementation from Python libraries
 - implement a dimensionality reduction algorithm based on clustering
 - use **Scikit-learn** Python library



Lab 2 – Nov. 12



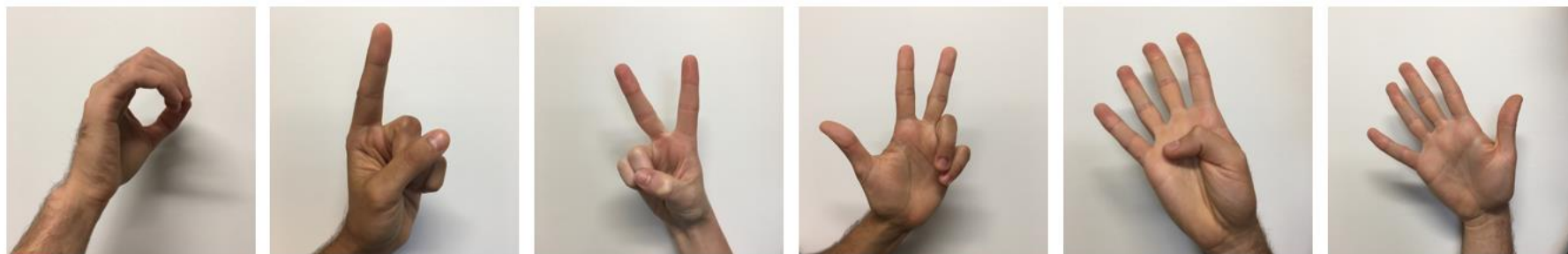
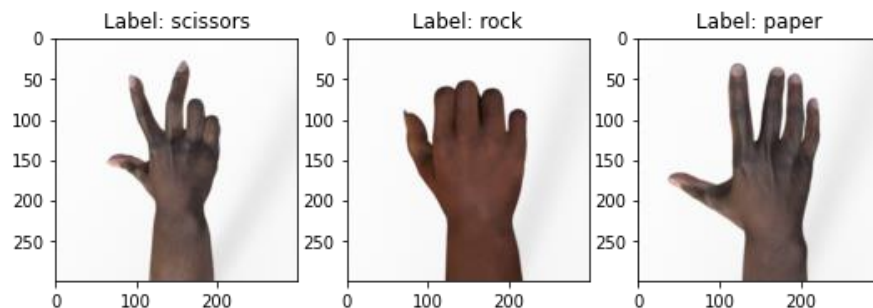
- CNN based classifier

- The challenge:

- hand gesture recognition

- You will learn to:

- implement a CNN based classifier using TensorFlow
 - train the classifier and test its performance



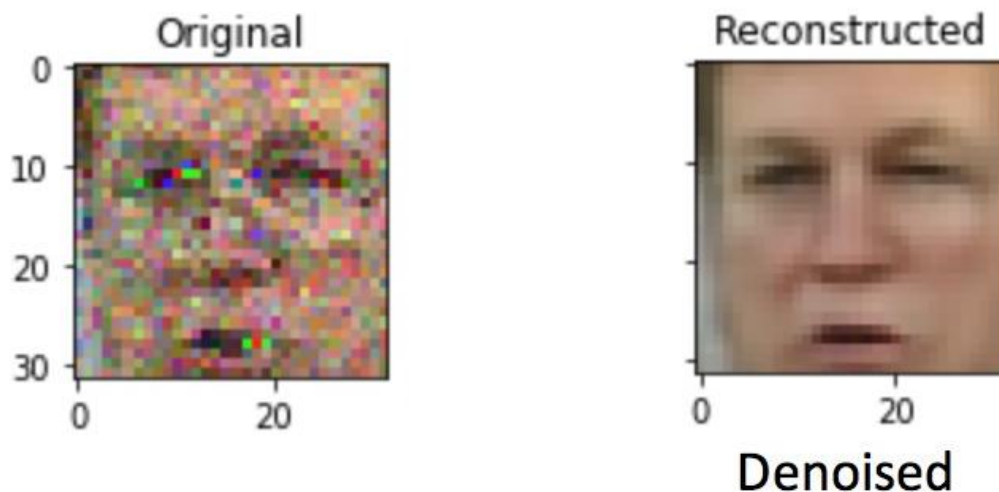
Lab 2a – optional for self study

- FFNN based classifier
 - Step-by-step implementation
 - The challenge:
 - classification of cat images
 - You will learn to:
 - implement a feed forward neural network classifier, defining the model, the forward and backward propagation steps and the update rule – TensorFlow is not used here!
 - train the classifier and test its performance

Lab 3 – Nov. 19



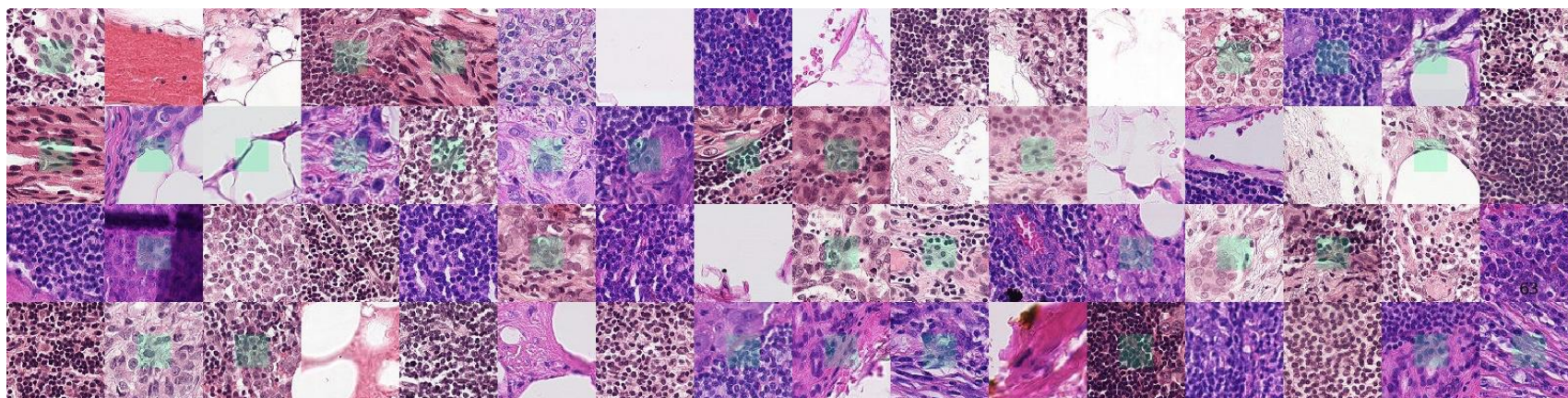
- CNN based autoencoder
 - The challenge:
 - image dimensionality reduction and denoising
 - You will learn to:
 - implement a CNN based autoencoder using TensorFlow
 - use the autoencoder to denoise noisy images



Lab 4 – Nov. 27



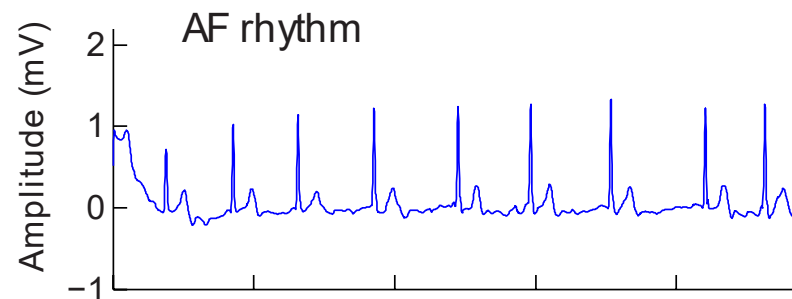
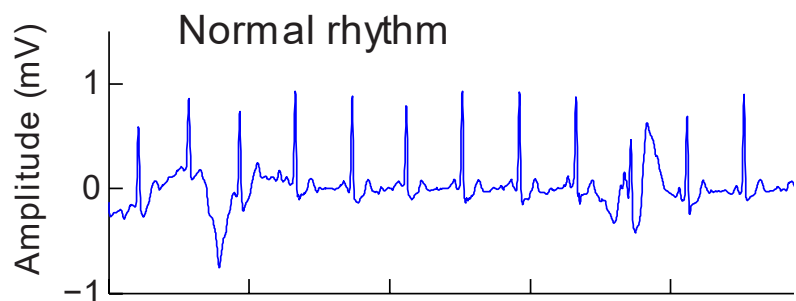
- Inception-V4 network for classification
 - The challenge:
 - metastatic tissue identification
 - You will learn to:
 - implement the basic building blocks of Inception-v4
 - combine them to shape the classifier



Lab 5 – Dec. 4



- CNN-RNN based classifier
 - The challenge:
 - abnormal heart rhythm recognition
 - You will learn to:
 - use **Pandas** Python library to handle data structures in Python
 - use **Scikit-learn** and **SciPy** Python libraries
 - preprocess the input data & implement a caching
 - implement a **CNN-RNN** based classifier using **TensorFlow**
 - train the classifier and test its performance using different metrics



Lab 6 – Dec. 11



- Attention mechanisms & sequence-to-sequence
 - The challenge:
 - neural machine translation
 - You will learn to:
 - integrate attention mechanism into a neural network architecture
 - use attention to create complex deep learning models
 - define custom `tf.keras.layers.Layers` and `tf.keras.Model` through the sub-classing strategy.
 - implement the encoder and decoder blocks, integrating an attention mechanism in the decoder part
 - train the model through the teacher-forcing approach on an Italian-English dataset
 - assess the performance of the translator on real sentences

Lab 6a – optional for self study

- Transformers



- The challenge:

- neural machine translation

- You will learn to:

- focus on the attention mechanism
 - define custom `tf.keras.layers.Layer`s and `tf.keras.Model` through the sub-classing strategy
 - implement the encoder and decoder blocks, integrating an attention mechanism into the decoder part
 - train the model through the teacher-forcing approach on an Italian-English dataset
 - assess the performance of the translator on real sentences

Lab 7 – Dec. 18



- Spiking neural networks

- The challenge:

- classification task

- You will learn to:

- implement a spiking neural network in snnTorch
 - assess the performance of the classifier



ML FOR HUMAN DATA ANALYTICS: A.Y. 24/25

Michele Rossi

michele.rossi@unipd.it

Francesca Meneghello (labs)

francesca.meneghello.1@unipd.it

