41001 Modern Applied Machine Learning a.y. 2021-22

PhD in Statistics & Computer Science, Bocconi University
Instructor: Carlo Lucibello

The course aims at giving a broad understanding of state-of-the-art deep learning techniques and their applications. We will overview both the theoretical aspects and the practical details. The code examples will be given in the Julia programming language.

Prerequisites

- Bring your laptop to the lectures.
- Familiarity with the Julia programming language.
- Intermediate level Calculus and basic Probability.
- Basic Machine Learning.

Bibliography

Lecture notes, papers, and code will be presented during the course.

Lectures

- Intro on Learning Frameworks (2h - Lesson 1)
    - Supervised, unsupervised, self-supervised, semi-supervised, reinforcement learning, active learning, transfer learning, one or few shot learning, continual learning
    - Perceptron algorithm
- Deep Leaning Basics (6h - Lesson 2-5)
    - Multilayer perceptron (1h)
    - Backpropagation, AD (1h)
    - Flux.jl Tutorials (2h)
    - Signal Propagation in deep net (Ganguli et al.) (2h)
- CNN (2h - Lesson 5)
    - BatchNorm, Dropout
    - ResNets
    - UNet
- Geometric Deep Learning (4h - Lesson 6-7)
    - Symmetries
    - Graph Neural Networks
    - Applications and code examples
- Generative Models (6h - Lesson 8-10)
    - VAE (1h)
        - Application to Protein modeling
    - GAN (1h)
    - Autoregressive networks (2h)
        - Solving Stat. Mech.
    - Normalizing Flows (1h)

  - Energy Based Models (1h)
- Reinforcement Learning (4h - Lesson 11-12)
  - Q-Learning   (1h)
  - Policy Gradient (1h)
  - AlphaZero and MCTS (1h)
- NeuralODE and ScientificML (4h - Lesson 13-14)
- Natural Language Processing (2h - Lesson 15)
  - Attention mechanism
  - Transformers

Grading

Grading will be based on a deep learning project, to be approved in advance by the instructor. There are a few different options available.

Option 1: Personal Project

Apply deep learning techniques to your research work. The code has to be made publicy available in a GitHub repo.

Option 2: Replicate Published Paper

Replicate the result of some paper whose code is not yet available in julia. The code has to be made publicy available in a GitHub repo.

Option 3: Contribute to the Julia ecosystem

Make some relevant contribution to the Julia ML ecosystem: Flux.jl, FluxML/model-zoo, Zygote.jl, Transformers.jl, GraphNeuralNetworks.jl, SciML org, Turing.jl, MLJ org, ...