

# Deep Learning

(6 CFU)

A.A. 2023-2024

## Programma del Corso

### Obiettivi formativi:

L'obiettivo del corso è quello di fornire competenze avanzate e specifiche nell'ambito delle architetture di reti neurali Deep. Il corso è costituito da una parte teorica e metodologica sui concetti fondamentali, e da una attività di laboratorio in cui tali concetti sono applicati nella risoluzione di problemi mediante recenti framework di sviluppo. Al termine del corso lo studente sarà in grado di: addestrare e ottimizzare in maniera adeguata reti neurali Deep; saper distinguere tra diverse soluzioni, e saper selezionare e personalizzare le architetture di reti più efficaci da utilizzare in ambiti applicativi reali, supervised, unsupervised o seguendo un approccio basato su un apprendimento per rinforzo.

### Programma:

#### 1. Introduzione al Corso

- Introduzione al deep learning.
- Algoritmi specifici per l'ottimizzazione e la regolarizzazione
- Richiami di feature engineering: One-hot encoding, Normalization, Standardization
- Valutazione delle performance dei modelli di DL

#### 2. Convolutional Neural Networks (CNN)

- Concetti base: Local receptive fields, shared weights, pooling, calcolo dei parametri
- Architettura LeNet
- Very deep convolutional networks per la Computer vision
- Architetture CNN (AlexNet, NiN, Residual Networks, VGG, Inception/GoogleNet, DenseNet)

#### 3. Tecniche di addestramento per reti Deep

- Inizializzazione dei parametri
- Batch normalization
- Funzioni di attivazione
- Transfer learning
- Optimizers (Momentum optimization, Algoritmi di Nesterov, AdaGrad e RMSProp; Adam optimization)
- Learning rate schedule
- Tecniche per affrontare l'overfitting (es. Early stopping,  $L1$  e  $L2$  regularization, Dropout, Max-norm regularization, Data Augmentation)

#### 4. Reti Ricorrenti (RNN) e architetture per time series

- Architetture RNN
- Memory cells: LSTM, GRU, Bidirectional RNN
- Predizione con RNN

- 1d convolution per time series
- Convolution layers per time series
- Dilated convolution
- RNN e residuals
- Deep RNN
- Quasi-Recurrent Neural networks

## 5. Autoencoders e Generative Adversarial Networks (GAN)

- Stacked autoencoders
- Convolutional autoencoders
- Recurrent autoencoders
- Denoising autoencoders
- Sparse autoencoders
- Variational autoencoders
- Architetture GAN
- Deep Convolutional GAN
- Architetture U-Nets e DeconvNet

## 6. Attention mechanisms e Transformers

- Beam search
- Attention mechanism
- Multi-head attention
- Self-attention
- Transformers
- Cenni su Word embeddings
- BERT
- GPT

## 7. Casi di Studio e Progetti

Si esporranno vari casi di studio e si proporranno progetti in cui applicare le nozioni apprese su vari domini d'interesse. In particolare le tematiche trattate potranno riguardare, tra l'altro, applicazioni di metodi e tecniche di DL nelle seguenti aree:

- **Social Media Analysis** (sentiment analysis, fake news detection, fake users detection, ecc.)
- **Analisi di segnali audio e video**
- **Time series analysis** (algorithmic trading, ecc.)
- **Recommender Systems** (social RecSys, cultural heritage RecSys, e-commerce RecSys, ecc.)
- **Data Science** (prediction functions per applicazioni pratiche, ecc.)
- **Visione Artificiale** (object detection, face detection and recognition, activity recognition, ecc.)
- **Bioinformatica** (riconoscimento di sequenze genetiche, ecc.)

### Materiale didattico:

Slide del corso.

Testi consigliati:

- A. Geron, "Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems", O'Reilly Media, Inc, USA, 2019.
- I. Goodfellow, Y. Bengio, and A. Courville, "[Deep Learning](#)", MIT Press, 2016.

- A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, "[Dive into Deep Learning](#)", 2020.
- M. Nielsen, "[Neural Networks and Deep Learning](#)", 2019.

**Metodi didattici:**

Lezioni frontali ed esercitazioni.

**Modalità di verifica dell'apprendimento:**

Scritto + progetto

**Strumenti a supporto della didattica:**

Librerie e tool di sviluppo per il deep learning:

- Python
  - Jupyter
  - Tensorflow e Keras
-