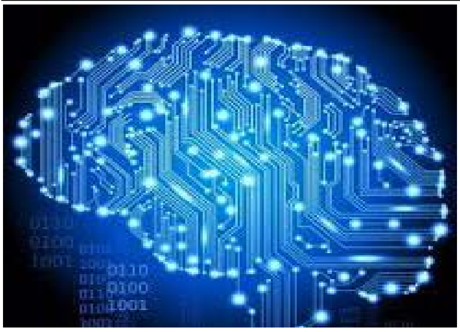
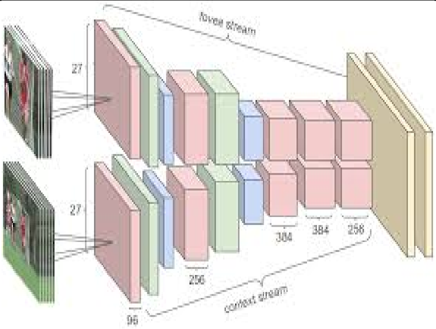
TELECOMMUNICATIONS, COMPUTER ENGINEERING, AND PHOTONICS INSTITUTE

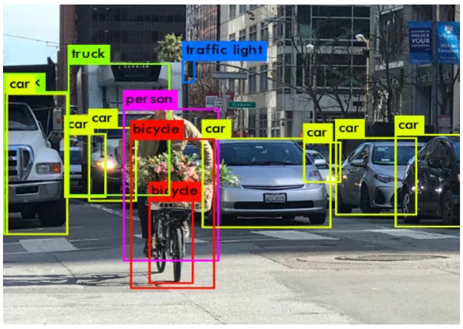
Sant'Anna School of Advanced Studies – Pisa

Retis Real-Time Systems Laboratory

Neural Networks and Deep Learning: Part 1: Theoretical Foundations







The first lecture is scheduled on January 13, 2026 at 9:00

Visit the course web page for registration and connecting to the channel.

http://retis.sssup.it/~giorgio/courses/neural/nn.html

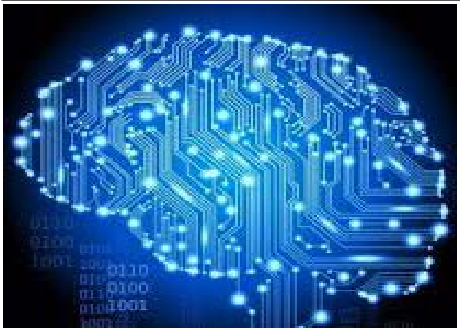
Giorgio Buttazzo

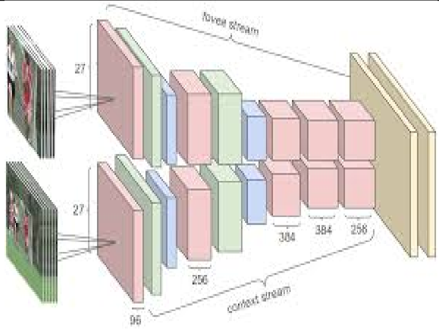
Objectives: The aim of this course is to provide key concepts and methodologies to understand neural networks, explaining how to use them for pattern recognition, image classification, signal prediction, system identification, and adaptive control.

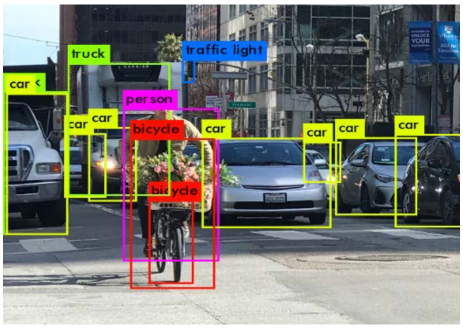
Course program

1. **Introduction to neural computing**. From classical AI to machine learning: a paradigm shift. Typical machine learning problems. The neural approach. Evolution of neural networks. Main network structures and learning paradigms. Hopfield networks and associative memories.
2. **Unsupervised learning**. The problem of high dimensionality. Dimensionality reduction and feature extraction: Principal component analysis and Self-organizing maps.
3. **Clustering algorithms**. Hard clustering algorithms: K-means, Hierarchical clustering, DBSCAN. Clustering evaluation metrics. Soft clustering algorithms: Fuzzy-C means, Gaussian Mixture Model.
4. **Reinforcement Learning**. Basic concepts and intuitions. Q-learning and SARSA algorithms. Examples of control applications. Model-based RL. The Dyna framework. Monte Carlo Tree Search.
5. **Supervised learning**. The Perceptron. Multi-layer networks. Backpropagation. Stochastic gradient descent. Applications to signal prediction, control, and system identification. Performance evaluation metrics. Managing overfitting and class imbalance. Radial basis functions (RBF) netowrks.
6. **Towards deep networks**. Problems in training deep networks: overfitting and vanishing gradient. Solutions for deep learning: regularization, and dropout methods. Softmax output layer. Autoencoders.
7. **Convolutional networks**. Basic principles and intuitions. Network architecture. Convolution operation. Convolution over a volume. Pooling layers. Normalization methods.
8. **Networks for classification**. Examples of CNNs and their key novelties: LeNet-5, AlexNet, VGG-Net, GoogLeNet, ResNet, Inception family, SqueezeNet, DenseNet, SENet, NASNet.
9. **Networks for object detection**. Two-stage detectors (the R-CNN family). Single-stage detectors (YOLO and SSD). Performance metrics. Pyramid networks (FPN, RetinaNet, the YOLO family).
10. **Networks for image segmentation**. Semantic, instance, and panoptic segmentation tasks. Fully Convolutional Network, U-Net, SegNet, ICNet, Mask R-CNN, YOLO-ACT.

Neural Networks and Deep Learning:
Part 2: Advanced Topics







The first lecture is scheduled on February 24, 2026 at 9:00
Visit the course web page for registration and connecting to the channel.
http://retis.sssup.it/~giorgio/courses/neural/nn.html

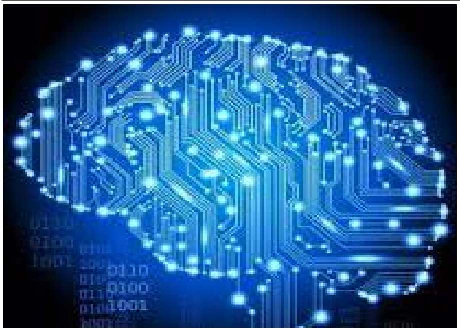
Giorgio Buttazzo

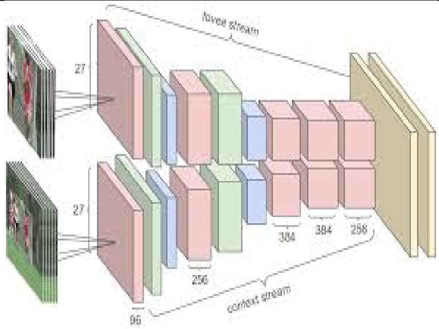
Objectives: This course focuses on methods for improving the performance of deep neural networks and overcome some of their limitations. Topics include recurrent neural networks, attention mechanisms, Transformers, deep reinforcement learning, semi-supervised learning, special deep learning models, neural networks for multi-object tracking, and generative networks.

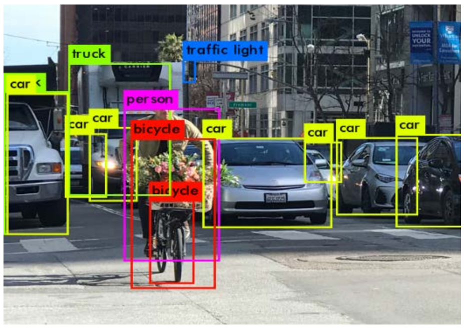
Course program

1. **Recurrent neural networks**. Sequence to sequence models. Backpropagation through time. Natural language processing: GRU, LSTM, and Bidirectional networks. Language modeling. Word embedding. Machine translation. The attention mechanism.
2. **Transformers**. Encoder-Decoder architectures. Positional encoding. Self-attention and multi-head attention mechanisms. Masked attention.
3. **Deep Reinforcement Learning**. Deep Q-learning models: DQL, Double-DQL, Dueling-DQL, Rainbow DQL, Recurrent DQL. Policy gradient and actor-critic methods: Reinforce, DDPG, PPO.
4. **Semi-supervised learning**. K-nearest neighbors. Self-training algorithms. Co-training. One-shot learning. Siamese networks. Zero-shot learning.
5. **Contrastive learning**. Principles of contrastive learning. Contrastive learning models: SimCLR, MoCo, Swav. Supervised Contrastive Learning.
6. **Model compression**. Weight quantization. Model pruning. Model distillation for transferring the knowledge of large DNNs to small DNNs. Optimized architectures for mobile devices.
7. **Neural networks for object tracking**. Generic object tracking. Tracking by detection: SORT, DeepSORT, FairMOT, BYTEtrack.
8. **Generative networks**. Generative autoencoders, Generative adversarial networks (GANs), GANs for style transfer. Variational Autoencoders, Diffusion models.

Neural Networks and Deep Learning:
Part 3: Trustworthy AI







The first lecture is scheduled on March 31, 2026 at 9:00
Visit the course web page for registration and connecting to the channel.
http://retis.sssup.it/~giorgio/courses/neural/nn.html

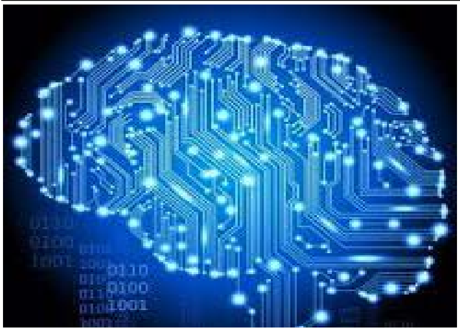
Giorgio Buttazzo, Federico Nesti, Giulio Rossolini, Daniel Casini

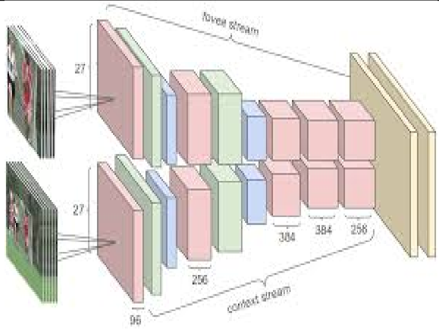
Objectives: This module presents methods for making neural networks more trustworthy, interpretable, safe, secure, and time predictable, in order to integrated them in safety critical systems as self-driving vehicles, autonomous robots, medical devices, and space probes.

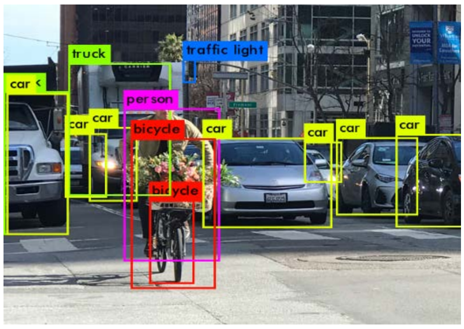
Course program

1. **Explainable and interpretable AI**. Definition and taxonomy. Common approaches for image processing and tabular data. Gradient-based methods and black-box methods.
2. **Anomaly detection**. Definition and taxonomy. Image-level and object-level out-of-distribution detection. Theoretical and empirical methods..
3. **Domain generalization**: Unsupervised and supervised domain adaptation. Adversarial learning for domain generalization. Teacher–student approaches. Augmentation and generative models for image enhancement.
4. **Attention mechanisms in computer vision**: CNN-based attention strategies and Vision Transformers. Recent Transformer architectures in computer vision.
5. **Adversarial attacks and defenses**. Introduction to adversarial machine learning. Taxonomy. Most common attacks. Defense methods to adversarial attacks.
6. **Real-worlds adversarial attacks and defenses**. Introduction to physical adversarial attacks. Simulated attacks. Practical considerations.
7. **Simulators for autonomous driving**. Features of existing simulators. The CARLA simulator. Hardware-in-the-loop simulators. Practical examples.
8. **Functional components on autonomous driving**. Basic blocks for perception, prediction, planning, control, and actuation.
9. **The Autoware framework for autonomous driving**. Overview of the framework. Neural networks in Autoware. Neural models for perception and prediction.

Neural Networks and Deep Learning:
Part 4: Implementation Issues







The first lecture is scheduled on May 5, 2026 at **9:00**
Visit the course web page for registration and connecting to the channel.
http://retis.sssup.it/~giorgio/courses/neural/nn.html

Giorgio Buttazzo, Alessandro Biondi, Daniel Casini, Federico Aromolo

Objectives: The aim of this course is to discuss practical and implementation issues useful to deploy neural networks on a variety of embedded platforms using different hardware platforms, programming languages, and development environments.

Course program

1. **Implementing Neural Networks from scratch in C**. General implementation principles. Implementing Self-Organizing Maps, Reinforcement Learning, and feed-forward networks.
2. **Development frameworks**. Overview of the existing frameworks. Common data sets. Common frameworks: Keras, Tensorflow, Caffe, and Pytorch. Examples of neural network implementations.
3. **DNN optimization for embedded platforms.** Weight quantization, pruning, distillation, to reduce execution times and contain memory footprints in resource constrained platforms.
4. **Accelerating deep networks on GPGPUs**. GPU programming in CUDA. Overview of the Nvidia TensorRT framework. Executing a DNN modeled in Caffe or PyTorch with TensorRT. Memory management in CUDA and TensorRT.
5. **GPU-based real-time neural vision**. Accelerating a neural network on TensorRT to detect objects from a video camera. Real-time multitasking of deep neural networks on GPU-based embedded platforms.
6. **Accelerating deep networks on FPGA**. Technologies and approaches. Overview of the AMD Deep Learning Processor Unit to execute convolutional networks. Multi-core DPU configurations.