LearnI!

ARTIFICIAL INTELLIGENCE SYLLABUS

WEEK-1:

Artificial intelligence fundamentals

The course aims to offer a view of the classical/symbolic approach to Artificial Intelligence and serves as a basis for more in-depth treatment of specific theories and technologies for building complete A.I. systems integrating different approaches and methods.

- Advanced search
- Constraint satisfaction problems
- Knowledge representation and reasoning
- Non-standard logics
- Uncertain and probabilistic reasoning (Bayesian networks, fuzzy sets)
- Foundations of semantic web: semantic networks and description logics
- Rules systems: use and efficient implementation
- Planning systems

WEEK-2:

Machine learning

We introduce the principles and the critical analysis of the main paradigms for learning from data and their applications. The course provides the Machine Learning basis for both the aims of building new adaptive Intelligent Systems and powerful predictive models for intelligent data analysis.

- Computational learning tasks for predictions, learning as function approximation, generalization concept.
- Linear models and Nearest-Neighbours (learning algorithms and properties, regularization).
- Neural Networks (MLP and deep models, SOM).
- Probabilistic graphical models.
- Principles of learning processes: elements of statistical learning theory, model validation.
- Support Vector Machines and kernel-based models.
- Introduction to applications and advanced models.

Applicative project: implementation and use of ML/NN models with emphasis to the rigorous application of validation techniques.

WEEK-3:

Human language technologies

The course presents principles, models and the state-of-the-art techniques for the analysis of natural language, focusing mainly on statistical machine learning approaches and Deep Learning in particular. Students will learn how to apply these techniques in a wide range of applications using modern programming libraries.

- Formal and statistical approaches to NLP.
- Statistical methods: Language Model, Hidden Markov Model, Viterbi Algorithm, Generative vs Discriminative Models
- Linguistic essentials (tokenization, morphology, PoS, collocations, etc.).
- Parsing (constituency and dependency parsing). - Processing Pipelines.
- Lexical semantics: corpora, thesauri, gazetteers.
- Distributional Semantics: Word embeddings, Character embeddings.
- Deep Learning for natural language.
- Applications: Entity recognition, Entity linking, classification, summarization.
- Opinion mining, Sentiment Analysis.
- Question answering, Language inference, Dialogic interfaces.
- Statistical Machine Translation.
- NLP libraries: NLTK, Theano, TensorFlow.

Distributed systems: paradigms and models

WEEK-4:

Intelligent Systems for Pattern Recognition

The course introduces students to the design of A.I. based solutions to complex pattern recognition problems and discusses how to realize applications exploiting computational intelligence techniques. The course also presents fundamentals of signal and image processing. Particular focus will be given to pattern recognition problems and models dealing with sequential and time-series data.

- Signal processing and time-series analysis
- Image processing, filters and visual feature detectors
- Bayesian learning and deep learning for machine vision and signal processing
- Neural network models for pattern recognition on non-sectorial data (physiological data, sensor streams, etc)
- Kernel and adaptive methods for relational data
- Pattern recognition applications: machine vision, bio-informatics, robotics, medical imaging, etc.
- ML and deep learning libraries overview: e.g. scikit-learn, Keras, Theano A final project will introduce students to the implementation of a pattern recognition application or to the development of computational intelligence applications.

WEEK-5:

Smart applications

The course aim is to explore methods and technologies for the development of smart connected applications, i.e. applications which exhibit intelligent behaviour -- through the use of artificial intelligence techniques introduced in other courses -- and that are deployed in immersive environments, including smart objects (as embodied by Internet of Things devices), mobile devices (smartphones, tablets), wearables (smartwatches, fitness trackers), home automation devices, web technologies, and cloud services and infrastructure. As such, applications considered for the course will include elements of context-awareness, sensor intelligence, spoken-language interfaces, The course will be based around a single case study for a novel smart application; students will cooperate as a single team, under the leadership of the instructor, in the design and implementation of a complete solution. In addition to standard lectures, classroom activities will include workshop-like sessions, where alternative designs are discussed, decisions are taken, and tasks are assigned. Weekly homework on the various phases of the joint project will be assigned to the team, and results reviewed the following week. The final goal is the delivery of a fully-functioning prototype of a smart application addressing the initial problem.

While the specific technologies adopted for each case study will vary based on needs and opportunities, the following general themes will be explored in lectures (examples of specific subjects are noted next to each theme):

Introduction to the course and to the case study

- ✓ examples: a voice-activated ambient assistant to answer student queries about the logistics of lectures in a classroom building, or autonomous software for a robotic rover for exploring inaccessible environments

Common designs for smart applications

- ✓ examples: fuzzy logic in control systems or cloud analysis of field sensors data streams

Make or buy: selecting appropriate procurement strategies

- ✓ example: writing your own RRN architecture vs. using cloud services

Development platforms for smart objects

- ✓ examples: Brillo (IoT devices) or Android TV (Smart TVs)

Development platforms for smart architectures

- ✓ examples: TensorFlow (server-side RNNs), or the Face Recognition API (mobile)

Cloud services for smart applications

- ✓ examples: Google Cloud Machine Learning API, Google Cloud Vision API, Google Cloud Speech API, or Deploying Deep Neural Networks on Microsoft Azure GPU VMs

Deployment and operations

- ✓ examples: cloud hosting vs. device hosting, or harnessing user feedback to drive improvement

Measuring success: methods and metrics

- ✓ examples: defining user engagement and satisfaction metrics, or assessing the naturalness of smart interactions