UAB Universitat Autònoma de Barcelona

Deep Learning

Code: 44737
ECTS Credits: 6

2025/2026

Degree	Type	Year
Research and Innovation in Computer based Science and Engineering	OP	1

Contact

Name: Silvana Silva Pereira

Email: silvana.silva@uab.cat

Teaching groups languages

You can view this information at the end of this document.

Prerequisites

It is recommended that students have knowledge and skills in:

- Programming in Python language
- Signal, Image and Video Processing
- Statistical validation

Objectives and Contextualisation

This course offers a practical introduction to neural network models and deep learning.

Students will consolidate and expand the theoretical knowledge acquired in previous machine learning courses, and complement them with new concepts related to the design of neural networks, current deep learning frameworks and the training process of these models.

Upon completion of the course, the student will have acquired:

- A solid knowledge of the different neural network architectures and the most common usage scenarios.
- The ability to critically select the most appropriate architecture and training mechanisms for each specific task.
- Practical experience in the use of deep learning libraries and environments to implement solutions to real problems.

Learning Outcomes

1. CA18 (Competence) Design the most appropriate neural network architecture in order to solve a given problem.
2. CA19 (Competence) Design computational solutions in multiple domains related to decision making based on the exploration of alternatives, uncertain reasoning and task planning.
3. KA23 (Knowledge) Describe the structure of convolutional and recurrent neural networks and to which environments they best suited for use.
4. KA24 (Knowledge) Describe the different data structuring and representation models.

5. KA25 (Knowledge) Describe advanced techniques for handling neural networks such as reinforcement learning, as well as adequately visualise the intermediate results of processing.
6. SA31 (Skill) Solve problems related to the analysis of large volumes of data by designing intelligent systems and using computational learning techniques.
7. SA32 (Skill) Solve specific problems using deep learning systems based on neural networks.
8. SA33 (Skill) Select the most appropriate neural network architecture according to the available data in order to obtain the expected results.
9. SA34 (Skill) Use neural network visualisation systems to evaluate possible optimisations that improve the system's performance.

Content

Introduction to neural networks: Perceptron, loss functions, training, and gradient backpropagation.

Convolutional neural networks (Deep Learning): Architectures for classification and segmentation (e.g., U-Net), and fine-tuning techniques for learning transfer.

Model validation and robustness: Reliable metrics, detection of bias in models, handling unbalanced datasets, and analysis of the level of generalization of models.

Model explainability: Visualization of activation and attention maps for interpreting model behavior.

Time series processing: Recurrent neural networks (LSTM), transformers applied to language (e.g., translators) and vision (e.g., Vision Transformers), and backbone architectures.

Unsupervised learning: Autoencoders, anomaly detection, and dimensionality reduction.

Generative models: Generative adversarial networks (GANs), variational autoencoders (VAEs), and other approaches to data generation.

Metric Learning (triplet loss, Barlow twins), one-shot approaches, Siamese networks.

Activities and Methodology

Title	Hours	ECTS	Learning Outcomes
Type: Directed			
Theoretical Explanations	20	0.8	
Type: Supervised			
Grupal Problem Resolution	30	1.2	
Type: Autonomous			
Personal work	90	3.6	

The course is based on the Project-Based Learning (PBL) methodology, aimed at enhancing student motivation and autonomy in their learning process.

At the beginning of the semester, teams of 4 or 5 students will be formed, who will develop a set of medium-complexity projects. These projects will be distributed throughout the course and weekly monitoring will be carried out, combining group and individual tutoring sessions.

- The projects will be proposed by the teaching staff and will meet the following requirements:
- They will be inspired by realistic situations or practical applications.
- They will be solvable with tools and knowledge accessible to students.

They will not have a known standard solution, in order to encourage creativity and critical analysis.

It is important to emphasize that the objective is not to find a universal optimal solution, but to propose a reasonable and well-justified solution. Often, in the real professional field, there is no single correct solution.

Each team will develop the projects with the greatest possible autonomy. The assigned tutor will have a supporting and supervisory role, avoiding directing or imposing solutions. Contributions must be original, although it is perfectly permissible - and even advisable - to consult bibliographic sources or resources available online. In these cases, however, the sources must be cited and their use must be explained both in the report and to the teaching staff.

The final project submission will consist of two parts:

- A written report, which describes the proposal, the process followed, the decisions made and the results obtained.
- An oral presentation, mainly addressed to a hypothetical entity that would have commissioned the project. Technical details should be reserved for annexes or specific sections of the report.

The oral presentation will be mandatory for the entire team, and the active participationof the rest of the class is expected, through questions and comments.

This subject allows the use of Artificial Intelligence (AI) technologies as an integral part of the development of projects. The student must clearly identify the parts generated with AI, specify the tools used and include a critical reflection on how these have influenced the process and the final result. The lack of transparency in the use of these technologies will be considered a lack of academic honesty.

Annotation: Within the schedule set by the centre or degree programme, 15 minutes of one class will be reserved for students to evaluate their lecturers and their courses or modules through questionnaires.

Assessment

Continous Assessment Activities

Title	Weighting	Hours	ECTS	Learning Outcomes
Class Co-evaluation Grade	10%	1	0.04	CA18, CA19, KA23, KA24, KA25, SA31, SA32, SA33, SA34
Group Grade	50%	6	0.24	CA18, CA19, KA23, KA24, KA25, SA31, SA32, SA33, SA34
Individual Grade	30%	2	0.08	CA18, CA19, KA23, KA24, KA25, SA31, SA32, SA33, SA34
Peer Co-evaluation Grade	10%	1	0.04	CA18, CA19, KA23, KA24, KA25, SA31, SA32, SA33, SA34

This subject does not include the single assessment system. Since the work revolves around a set of projects developed throughout the semester, the assessment is continuous and the final result is not recoverable.

Assessment methodology

The assessment is based on several instruments and activities:

1. Group assessment (0 to 10)

Carried out by the teaching staff (both tutors and non-tutors) based on the project submissions and presentations. It consists of:

- PROJECT PORTFOLIO: Document that includes the development of the project (approach, meeting minutes, technical justification, implemented application, tests carried out and user manual).
- PRESENTATION: Oral presentation (10-15 slides) of the project and the results obtained.
- FINAL REPORT: Written document that describes the program developed and the results obtained.
- DOCUMENTATION: Delivery, monitoring and compliance controls.

Group grade = 0.25 × Midterm report + 0.75 × Final report, where:

Final report = 0.8 × Written report + 0.2 × Presentation

2. Individual assessment (0 to 10)

Includes two components:

- Observation by the tutor during the tutored sessions, which assesses attitude, initiative, participation, attendance and punctuality.
- Individual portfolio, where each student documents their contributions: acts, decisions made, technical justifications, personal reflections and tasks carried out.

Individual grade = 0.7 × Tutor observation + 0.3 × Individual portfolio

3. Peer assessment (0 to 10)

At the end of each project, each group member fills out a peer assessment survey assessing the contributions of their peers.

4. Collective peer assessment (0 to 10)

Each group scores the others' projects in a public presentation session. The group rated as the best gets 10 points, the next gets 8, and so on.

5. Calculation of the project grade

Project grade =

0.5 × Group grade +

0.3 × Individual grade +

0.1 × Peer assessment +

0.1 × Collective peer assessment

Each project will contribute to the final grade as follows:

6. Final grade for the subject

Final grade = weighted average of the grades of all projects (all projects have the same weight).

7. Non-assessable

According to point 9 of article 266 of the UAB Academic Regulations, if the student does not provide sufficient evidence of learning throughout the course, the subject will be classified as non-assessable. In this case, it will be understood that the student has not participated sufficiently in the continuous assessment activities, such as:

- Not having submitted any project or having done so clearly insufficiently.
- Not having attended any class session or having participated in the follow-up activities.
- Not having presented the individual portfolio or having participated in the co-assessments.

This criterion will be applied objectively and will be reviewable according to the procedure established by the degree.

Bibliography

Books

Deep Learning, Ian Goodfellow, Yoshua Bengio, and Aaron Courville, MIT Press, 1st Ed. 2016

Pattern Recognition and Machine Learning, Christopher Bishop, Springer, 2011

Neural Networks for Pattern Recognition, Christopher Bishop, Oxford University Press, 1st ed., 1996

Books online:

Zhang, Z.C. Lipton, M. Li, A.J. Smola, *"Dive into Deep Learning", 2021* https://d2l.ai/

Michael Nielsen's *Neural Networks and Deep Learning* http://neuralnetworksanddeeplearning.com/

Links (Tutorials and Talks):

https://towardsdatascience.com

https://www.datacamp.com

https://medium.com

https://cs.stanford.edu/~sanmi/talks.html

Software

The course uses Python along with PyTorch, TensorFlow, and CUDA. Access to GPU and CPU clusters is facilitated whenever possible to support computational needs.

Groups and Languages

Please note that this information is provisional until 30 November 2025. You can check it through this link. To consult the language you will need to enter the CODE of the subject.

Name	Group	Language	Semester	Turn
(PLABm) Practical laboratories (master)	1	English	second semester	afternoon
(TEm) Theory (master)	1	English	second semester	afternoon