UAB
Universitat Autònoma de Barcelona

Deep Learning

Code: 44737
ECTS Credits: 6

2024/2025

Degree	Type	Year
4318303 Reseach and Innovation in Computer Based Science and Engineering	OT	0

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 subject aims to give a practical introduction to neural network models and deep learning.

The students will consolidate and extend their theoretical background by building on top of previous subjects on machine learning, complementing previous knowledge with new concepts on neural network design, deep learning frameworks, and the training process for such models.

The students should finish this subject having a broad knowledge of different neural network architectures and their typical use scenarios, and a proven ability to critically choose the right architecture and training mechanisms for each task.

Finally, the students will receive hands-on training and acquire practical experience on using current deep learning frameworks to solve specific tasks.

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 function, training, back propagation.

Introduction to deep learning convolutional neural networks: classification architectures, segmentation architectures (Unet), fine tuning.

Validation: reliable metrics, detection of biases in models, unbalanced problems, level of generalization of models. Explainability: activation and attention maps.

Processing of temporal data: recurrent networks (LSTMs), transformers (translators, visual transformers), backbone architectures

Unsupervised learning, autoencoders and anomaly detection.

Generative models, adversarial Networks, variational autoencoders.

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	KA23, KA24, KA25
Type: Supervised			
Grupal Problem Resolution	30	1.2	CA18, CA19, SA31, SA32, SA33, SA34
Type: Autonomous			
Personal work	90	3.6	CA18, CA19, KA23, KA24, KA25, SA31, SA32, SA33, SA34

The course will follow a learning methodology called Project Based Learning (PBL). The PBL methodology aims to empower and motivate the student in his/her learning process. Groups of 5 to 6 students will be arranged to carry out a set of projects (of medium size) throughout the semester. There will be weekly monitoring and both group and individual tutoring of the students.

The projects are set by the teaching staff so that they meet the following criteria: be as real as possible; be treatable using elementary tools; not have a standard algorithm associated as a solution.

On the other hand, it is essential to understand that it is not about finding an algorithm that works in 100% in all cases - often there is no such thing - but simply about "providing a reasonable solution proposal".

Projects must be developed by each team with maximum possible autonomy. Each team will be assigned a tutor who will follow the evolution but in principle will refrain from imposing his/her ideas. On the other hand, the student must be clear that it is not a question of looking for 'the' solution of the problem in other places, but of providing an original contribution. This does not mean that information that may exist in the bibliography or on the Internet should be given up; but when used, it should be properly reported in memory.

The realization of the project must end in a computer program and a final report. In addition to delivering in written form, the results of this report will be the subject of an oral presentation. Both, written memory and oral presentation, should be directed mainly to the entity, surely hypothetical, that would have proposed the problem. As a general rule, technicalities will be subsumed to specific sections of the written report.

In the oral presentations of the projects, the entire class is expected to attend, and all students should intervene through questions and observations.

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	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	KA23, KA24, KA25
Peer Co-evaluation Grade	10%	1	0.04	CA18, CA19, KA23, KA24, KA25

This subject does not contemplate the single evaluation system. Given that most of the work revolves around a set of projects that is developed throughout the course, the evaluation has a continuous nature, and its final result is not recoverable.

Evaluation Methodology

The minutes that the group will write after each tutored session will be evaluated, in which they will describe their discussions and agreements and the self-evaluation surveys that each student will take on their classmates and on themselves. At the end of each project, the students will make an oral presentation of the project and will deliver a report of the work carried out. Both will be evaluated by the teachers of the subject, whether or not they are tutors. Students will not take any written exams.

For the evaluation, the following INSTRUMENTS and ACTIVITIES will be used:

- An evaluation made by the teachers from the presentation of the projects carried out by the group (quality of work, presentation, memory delivered). Group grade (0 to 10) From:

STUDENT PORTFOLIO: Document where the development of the work done is explained: project approach, meeting minutes, information sought, explanation of the implemented application with a small user manual and tests and tests performed.

PRESENTATION: Oral presentation in 10-15 slides on the project developed and results obtained.

REPORT: Manuscript describing the developed program and the results obtained.

ACTS AND CONTROLS: Presentation of the documentation delivered.

- An individual evaluation based on the observations made by the tutors in the tutored sessions, where the attitude, initiative, participation, attendance and punctuality of the student in the group sessions will be taken into account. Individual mark (0 to 10).
- Co-evaluation and self-evaluation surveys among group members at the end of each project. Peer Coevaluation Note (0 to 10).
- Oral presentations are made before the students and the groups will assess the work of their classmates as a ranking. The group that is in 1st position will receive 10 points, the 2nd 8 points and so on. Class Coevaluation Note (0 to 10).

Grades

Each project will have a grade that will be calculated as follows:

Group Mark = 0.25 * Intermedial report + 0.75 * Final report

Final report = 0.8 * Report + 0.2 * Presentation

Project Grade = 0.5 * Group Grade + 0.3 * Individual Grade + 0.1 * Peer Coevaluation Grade + 0.1 * Class Coevaluation Grade

The final grade will come from the weighted average of the projects carried out. The weighting will be the same for all projects

To distinguish between 'failed' and 'no-show', a deadline is set for students to unsubscribe from the evaluation, in which case they will appear as 'no-show'. To unsubscribe, you must notify the teacher, in writing or by email, and obtain an acknowledgment of receipt.

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:

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

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

Links (Tutorials and Talks):

https://towardsdatascience.com

https://www.datacamp.com

https://medium.com

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

Software

Python with pytorch, tensorflow, cuda. As far as possible, we facilitate access to clusters of GPUs and CPUs.

Language list

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