ACADEMIC PROGRAM

ARTIFICIAL INTELLIGENCE

B.F.A. IN **COMPUTER SCIENCE**

MODALITY: ON CAMPUS

ACADEMIC YEAR: 2022-2023

Name of the course:	Artificial Intelligence
Degree :	Computer Science
Location:	Centro Universitario de Tecnología y Arte Digital
Area:	Programming
Year:	3º
Teaching period:	1
Type:	OB
ECTS credits:	6
Teaching modality:	On campus
Language:	English
Lecturer / Email	-
Web page:	http://www.u-tad.com/

SUBJECT DESCRIPTION

Area description

This subject belongs to the programming subject. This subject is dedicated to the study of programming techniques and languages on which the software engineering degree studies will be based.

Subject description

This subject is dedicated to the study of that extensive, multidisciplinary and in some cases even controversial concept such as Artificial Intelligence (AI). AI can be defined as the set of capabilities that can be assimilated to human capabilities but developed by machines (for now, just "computers" as we know them). It is therefore about developing systems that perform "human-like" functions such as learning, solving problems, communicating with natural language, recognizing and identifying people or other components of an image and an increasingly extensive list of applications. Some existing, others imagined, some of which are probably possible in a short period of time, because if there is something to highlight about this technological field it is the enormous speed of its developments.

The enormous current development is undoubtedly due to the consolidation and popularization of the "deep learning" paradigm, helped by the availability of specific hardware for the calculation of enormous models and which is revolutionizing traditional areas of computing such as computer vision, natural language processing, and even artistic creation

The main effort of the subject will be to provide as complete an introduction as possible for the use of these technologies in any area of software development, as well as knowing some of their weak points (which of course they have) and important precautions regarding implementation. of this type of systems in specific contexts.

We will study the theoretical and technological bases that have enabled the enormous development of this field, especially in the last 20 years, both existing and realistic applications in the short term, in sectors such as image recognition, natural language processing (NLP), games, and a long etcetera.

With this objective, we will focus from the first moments on the latest developments of what is known as "Deep Learning", exploring a wide variety (within the possibilities of the course) of approaches within this field and always with specific cases of application. .

The focus of the course will therefore be mainly practical, based on the exhaustive development of examples on the most common development environment at the moment, which is Python and its specialized libraries.

COMPETENCIES AND LEARNING OUTCOMES

Competencies

BASIC AND GENERAL SKILLSs

CG1 - Ability to understand, schedule and solve problems trough software development

CG3 - Knowledge of the scientific fundamentals applicable to the resolution of computer problems

CG4 - Ability to simplify and optimize computer systems by understanding their complexity

CG9 - Ability to learn, modify and develop new software solutions

CG10 - Use of creative techniques to carry out computer projects

CB1 That students have demonstrated knowledge and understanding in an area of study that starts from the basis of general secondary education, and is usually at a level that, although it is supported by advanced textbooks, also includes some aspects that involve knowledge from the forefront of their field of study.

CB2 Students are able to apply their knowledge to their work or vocation in a professional manner and possess the competences usually demonstrated through the development and defense of arguments and problem solving within their field of study.

CB3 That students have the ability to gather and interpret relevant data (usually within their area of study) in order to make judgements that include reflection on relevant social, scientific or ethical issues.

CB4 Students are able to convey information, ideas, problems and solutions to both specialist and non-specialist audiences.

CB5 That students have developed those learning skills necessary to undertake further studies with a high degree of autonomy.

SPECIFIC SKILLS

CE1 - Knowledge of the structure of computers, the concepts of coding, manipulation, information processing and use of low-level languages

CE7 - Knowledge of the main types of data structures and use of libraries and algorithmic techniques associated with these structures together with the complexity orders that characterize these techniques

CE8 - Conocimiento de los distintos paradigmas detrás de los lenguajes de programación/ Knowledge of the different software

paradigms that underpin programming languages

CE9 - Knowledge of effective control structures, variables, programming syntax and memory management in the development of a computer application

CE10 - Ability to work with a release manager and generate application documentation automatically.

CE15 - Knowledge of fault tolerance, adaptability, load balancing and system predictability for distributed application development

CE17 - Knowledge of the parallelization characteristics of graphics cards and high-performance architectures for application development.

CE20 - Ability to test the operation and functionality of a computer application, develop test plans and use test-oriented design and programming techniques

CE23 - Knowledge of the principles of artificial intelligence and use of deterministic search algorithms and state machines

Learning outcomes

Upon completion of the degree, the graduate will be able to:

- To understand and handle the concept of dynamic memory

- To identify classes with the relevant data of a problem

- To instance classes and objects and manage them

- To understand and use the mechanisms of inheritance, polymorphism and operator overloading

- To identify class relationships in different use-cases.

- To master an object oriented programming language.

- To master programming patterns

- To know different problem solution strategies from an algorithmic view point: divide and conquer, dynamic programming, backtracking or genetic algorithms.

- To understand algorithmic complexity, assess it and search for optimal solutions

- To code a program able to find the optimal path between any pair of nodes of a graph

- To build neural networks to solve applied problems

CONTENTS

Amorphous computing, Self-stabilization and grid computing

Knowdledge representation

Production systems

Fuzzy reasoning

Pathfinding algorithms

Behaviour Trees

Task planners

SUBJECT SYLLABUS

1. Introduction: towards a definition of what AI is right now

1.1. How ChatGPT (Open AI GPT-3) was developed

1.2. AI in the media and fiction (Debate)

2. Environment for the development of the subject

2.1. Python and its libraries and environments

2.2. Interactive Python: jupyter notebooks (+ Google Colab)

23. Numpy

2.4. Deep learning libraries: Tensorflow, Keras, Pytorch

3. Methodological bases of Machine Learning

4. Artificial Neural Networks (ANN)

4.1. Mathematical basis of ANN computation

4.2. ANN with python + numpy

4.3. ANN with keras

5. Deep Learning (1)

5.1. Convolutional Networks (or Convolutional Neural Networks, CNN)

6. Deep Learning (2)

6.1. Natural Language Processing (NLP): Recurrent Neural Networks (RNN)

7. Reinforcement Learning

8. Unsupervised Deep Learning

8.1. Autoencoders

8.2. Generative Adversarial Networks (GAN)

9. Classic problem-solving approaches

9.1. CSP (Constraint Satisfaction Problems)

10. Ethics in AI and legal regulation

TRAINING ACTIVITIES AND TEACHING METHODOLOGIES

TRAINING ACTIVITIES

LEARNING ACTIVITIES	Total hours	Hours of presence
Theoretical / Expository classes	35,64	35,64
Practical classes	18,91	18,91
Tutorials	4,00	2,00
Independent study and autonomous work of the student	51,82	0,00
Elaboration of work (group or individual)	33,82	0,00
Evaluation Activities	5,82	5,82
TOTAL	150	62,37

Teaching methodologies

Expository method or master lesson

Case learning

Learning based on problem solving

Cooperative or collaborative learning

inquiry learning

Flipped classroom methodology

Gamification

Just in time Teaching (JITT) or classroom on time

Expository method or master lesson

Case method

Learning based on problem solving

Cooperative or collaborative learning

inquiry learning

Flipped classroom methodology

Gamification

TEMPORAL DEVELOPMENT

DIDACTIC UNITS / TOPICS TIME PERIOD

Topic 1 Introduction: towards a definition of what AI is right now Week 1

Topic 2 Environment for the development of the subject (python and libraries) Weeks 1 and 2

Topic 3 Methodological bases of Machine Learning Weeks 2 and 3

Topic 4 Artificial Neural Networks (ANN) Weeks 4 and 5

Topic 5 Deep Learning - CNNs Weeks 6 and 7

Topic 6 Deep Learning - NLP and RNNs Weeks 8 and 9

Topic 7 Reinforcement Learning (Reinforcement Learning) Week 10

Topic 8 Unsupervised Deep Learning: autoencoders and GANs Weeks 11 and 12

Topic 9 Classic problem-solving approaches Week 13

Topic 10 Ethics in AI and legal regulation Week 14

Partial Week 14 (last days of course in December)

EVALUATION SYSTEM

ASSESSMENT SYSTEM	MINIMUM SCORE RESPECT TO THE FINAL ASSESSMENT (%)	MAXIMUM SCORE RESPECT TO THE FINAL ASSESSMENT (%)
Assessment of participation in class, exercises or projects of the course	0	30
Assessment of assignments, projects, reports, memos	30	80
Objective test	10	60

GRADING CRITERIA

ASSESSMENT SYSTEM	ORDINARY EVALUATION	EXTRAORDINARY EVALUATION
Assessment of participation in class, exercises or projects of the course	10	10
Assessment of assignments, projects, reports, memos	60	60

Objective test	30	30

General comments on the evaluations/assessments

● The evaluation of participation in class, in practices or in projects of the subject will be carried out based on attendance and active participation in class and in the rest of the activities developed during the course. This aspect will represent 10% of the final grade for the subject in the ordinary call.

● Throughout the course, 2 activities or exercises associated with two basic Deep Learning topics will be proposed (also called “partial works” below):

o CNN for image classification

or RNN for NLP

● that must be delivered before the deadline and uploaded to the virtual platform (blackboard). Each of these activities will be graded strictly independently and that grade will be 30% of the final grade for the subject.

● In order to pass the subject, it will be an essential requirement to have at least a grade of 4 *in each of the two proposed partial works*. Review procedures will be enabled and, if any of the work is not approved, additional recovery work or works, different from the initial ones, must be carried out at a later date, always prior to the date of the ordinary call.

● If this recovery task(s) is suspended (<4 or when it does not average >= 5 with the rest of the grades), the ordinary call will be suspended and the student must take the extraordinary call (exams). theoretical and practical), regardless of the grades obtained in the ordinary exams, in addition to performing a recovery task(s) again.

● During the month of December, a partial objective exam (theoretical + practical) will be carried out to liberate this content. That is, if a pass is achieved (minimum grade 5) in this exam, it will not be necessary to take the official exam for the January 2024 call, but it will *always* be a requirement to have passed both the partial and final assignments or exercises, and that the simple average (arithmetic mean of all of them) is 5.

● As a final project, a project on the application of any of the technologies included in the subject is proposed.

o The final work is individual, although I may accept proposals for the formation of groups for projects that will strictly require the teacher's approval.

o This work will be presented without delay before the official date of the ordinary call in January 2024, and its grade will account for 30% of the total score for the subject, and so that this grade can be averaged with the rest of the grades must exceed a 4 (out of 10).

o There will be great (practically total) freedom over the topic of the work or project (theoretical, practical), as well as its presentation format. Optionally, this work - at any time during its development - can be presented to the teacher to receive comments and, of course, in case difficulties are found to try, in a common way, to solve them - whenever possible within the deadline. time of the subject.

o The grading of this work in its technical aspects, relevance and originality will be the responsibility of the teacher. All of this will be duly explained before the necessary presentations.

● To pass the subject in the ordinary call, the arithmetic average of all the grading elements (partial exams, final work or project and participation or attendance in classes) according to the following formulas

min(partial_work_scores) = 4

partial_work_grades = sum(partial_work_scores) / 2

min(midterm_exam) = 5

min(grade_final_work) = 4

qualifies_INAR_2024 = (qualifies_partial_works + partial_exam + qualifies_final_work) / 3

qualifies_final_INAR_2024 = (qualifies_INAR_2024 * 0.9) + attendance_participation_etc

● The final score qualifies_INAR_2024 must be equal to or greater than 5 (always out of 10) in order to consider the subject passed in the ordinary call.

● If any of the minimum score requirements for each of the components are not met, the student must present/repeat all those components that are below the minimum required and, in the case of the exam, must take the exam. ordinary call for January 2024.

● If the student does not obtain approval in the ordinary call, the student must re-sit the failed parts, maintaining the same evaluation criteria for the extraordinary call.

LIST OF REFERENCES (BOOKS, PUBLICATIONS, WEBSITES):

Basic and with which you can follow most of the course:

Jordi Torres (2020): Python Deep Learning. Practical Introduction with Keras and Tensorflow2. Marcombo Publishing House. ISBN: 978-84-267-2828-9

https://torres.ai/python-deep-learning/

François Chollet (2020): Deep Learning with Python, Second Edition. Manning.

https://www.manning.com/books/deep-learning-with-python-second-edition

(with their exercise notebooks - freely available):

https://github.com/fchollet/deep-learning-with-python-notebooks

(also available in Spanish):

https://anayamultimedia.es/libro/titulos-especiales/deep-learning-con-python-francois-chollet-9788441542259/

Recommended although not in their entirety or less complete or current than the previous ones:

Google Machine Learning “crash courses”:

https://developers.google.com/machine-learning

Deep Learning Basics:

https://colab.research.google.com/github/lexfridman/mit-deep-learning/blob/master/tutorial_deep_learning_basics/deep_learning_basics.ipynb

MIT Deep Learning and Artificial Intelligence Lectures:

https://deeplearning.mit.edu/

(extraordinarily detailed in mathematics but not in applications)

Ian Goodfellow, Yoshua Bengio, Aaron Courville (2016): Deep Learning - An MIT Press book.

https://www.deeplearningbook.org/

Alto, V. (2023): Generative artificial intelligence with ChatGPT and OpenAI models. Anaya Multimedia. (available from October 2023)

https://anayamultimedia.es/libro/titulos-especiales/inteligencia-artificial-generativa-con-modelos-de-chatgpt-y-openai-valentina-alto-9788441548961/

(these below are only recommended for the most “classic” AI):

Stuart J. Russell and Peter Norvig (2016): Artificial Intelligence. A Modern Approach. Third Edition. Pearson

Wolfgang Ertel (2017): Introduction to Artificial Intelligence. Second Edition. Springer

REQUIRED MATERIALS, SOFTWARE AND TOOLS

Type of classroom
Theory classroom

Board and projection system

Materials:
Personal Computer

Software:
Python:

Última versión de Python:https://www.python.org/downloads/

Como mínimo serán necesarias las siguientes librerías:

- virtual environments (virtualenv): https://virtualenv.pypa.io/en/latest/

- jupyter notebooks (https://jupyter.org/)- numpy (https: