
Ano Letivo 2019-20

Unidade Curricular REDES NEURONAIS E SISTEMAS DIFUSOS

Cursos ENGENHARIA INFORMÁTICA (2.º ciclo)
ENGENHARIA ELETRÓNICA E TELECOMUNICAÇÕES (Mestrado Integrado) (*)

(*) Curso onde a unidade curricular é opcional

Unidade Orgânica Faculdade de Ciências e Tecnologia

Código da Unidade Curricular 14741035

Área Científica CIÊNCIA DE COMPUTADORES

Sigla

Línguas de Aprendizagem Português-PT ou Inglês-EN

Modalidade de ensino Blended Learning

Docente Responsável António Eduardo de Barros Ruano

DOCENTE	TIPO DE AULA	TURMAS	TOTAL HORAS DE CONTACTO (*)
António Eduardo de Barros Ruano	T; TP	T1; TP1	30T; 30TP

* Para turmas lecionadas conjuntamente, apenas é contabilizada a carga horária de uma delas.

ANO	PERÍODO DE FUNCIONAMENTO*	HORAS DE CONTACTO	HORAS TOTAIS DE TRABALHO	ECTS
1º	S1	30T; 30TP	168	6

* A-Anual;S-Semestral;Q-Quadrimestral;T-Trimestral

Precedências

Sem precedências

Conhecimentos Prévios recomendados

Programação básica e Álgebra básica

Objetivos de aprendizagem (conhecimentos, aptidões e competências)

O aluno, no final da disciplina, deverá ser capaz de:

- Compreender a inspiração biológica das redes neuronais, suas principais características e aplicações;
- Conhecer as redes percetões multicamadas, de função de base radial e as técnicas mais comuns de treino e de adaptação supervisionados; saber utilizá-las no contexto de aproximação de sistemas e de classificação;
- Conhecer os métodos mais comuns de treino não-supervisionado, baseados nas regras de Hebb e em competição, e as redes de Kohonen; saber utilizá-las em problemas de agrupamento de dados e classificação.
- Conhecer as principais redes neuronais recorrentes, e seus algoritmos de treino; saber utilizá-las em aplicações práticas;
- Conhecer os principais mecanismos de aprendizagem profunda; saber utilizar redes de confiança profunda e convolucionais em aplicações práticas;

Conteúdos programáticos

1. Introdução às Redes Neurais Artificiais
2. Aprendizagem supervisionada para redes neuronais diretas
 1. Modelos
 2. Algoritmos de Aprendizagem
 1. Topologia fixa
 2. Algoritmos Construtivos
 3. Aplicações para aproximação de funções e reconhecimento de padrões
3. Aprendizagem não supervisionada
 1. Aprendizagem Não Competitiva
 1. Regras do tipo Hebbian
 2. Memórias Associativas Lineares
 2. Aprendizagem Competitiva
 1. Mapas auto-organizados
 2. Classificadores de redes competitivas
4. Redes Neurais Recorrentes
 1. Introdução aos sistemas dinâmicos não lineares
 2. Modelos
 3. Treino e Operação
 4. Aplicações de Redes Neurais Recorrentes
5. Aprendizagem Profunda
 1. Revisitando os algoritmos de descida de gradiente
 2. Máquinas restritas da Boltzmann e redes de confiança profunda
 3. Redes Neurais Convolucionais

Demonstração da coerência dos conteúdos programáticos com os objetivos de aprendizagem da unidade curricular

Cada um dos objetivos desta unidade curricular é lecionado na secção correspondente do conteúdo programático

Metodologias de ensino (avaliação incluída)

As aulas serão distribuídas por componentes Teórica (T) e Prática Laboratorial (PL). Nas aulas T será exposta a teoria, e apresentados exemplos de aplicação. Nas aulas PL os estudantes implementarão programas em Matlab ou Python, utilizando os módulos disponíveis em cada uma destas linguagens.

Nesta disciplina haverá um mini-projeto, com um peso de 50% da nota, e um exame, realizado na época normal para a disciplina, com igual peso para a nota final. Se a nota final for igual ou superior a 10, e ambas as componentes forem classificadas com uma nota superior a 8, o aluno passa.

Caso a nota final for inferior a 10 e a nota do mini-projeto for superior a 8, o aluno é admitido a exame de recurso, que substituirá a nota do exame da época normal.

Demonstração da coerência das metodologias de ensino com os objetivos de aprendizagem da unidade curricular

Esta disciplina tem um formato clássico, com aulas teóricas em anfiteatro e aulas práticas em laboratório de programação. As aulas teóricas exprimem a unidade da cadeira e subliminarmente pretendem conduzir os alunos na direção dos objetivos de aprendizagem. Mais explicitamente, na sua leção, o professor explica os problemas, os métodos computacionais utilizados para os resolver, e suas implementações em Matlab ou Python.

As aulas práticas funcionam em regime de laboratório aberto. Os trabalhos a realizar serão publicados na página Web da cadeira. Os alunos trabalham nas aulas, mas sobretudo fora das aulas. Para incentivar a participação coletiva, os alunos são encorajados a intervir nos fóruns da cadeira, para discutir questões relacionadas com os problemas propostos e outras questões gerais.

Bibliografia principal

Apontamentos on-line

Ruano, A. E. (Ed.). Intelligent Control using Intelligent Computational Techniques. 2005: The Institution of Electrical Engineers.

Gori, M., Machine Learning. A Constraint-based Approach. 2018: Morgan Kaufmann.

Haykin, S.O., Neural Networks and Learning Machines. 3 ed. 2008: Prentice Hall.

Aggarwal, C.C., Neural Networks and Deep Learning. A Textbook. 2018: Springer.

Buduma, N. and N. Locascio, Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithms. 2017: O'Reilly Media.

Skansi, S., Introduction to Deep Learning: From Logical Calculus to Artificial Intelligence. 2018: Springer International Publishing.

Academic Year 2019-20

Course unit NEURAL NETWORKS AND FUZZY SYSTEMS

Courses INFORMATICS ENGINEERING
ELECTRONIC ENGINEERING AND TELECOMMUNICATIONS (Integrated Master's) (*)

(*) Optional course unit for this course

Faculty / School FACULTY OF SCIENCES AND TECHNOLOGY

Main Scientific Area CIÊNCIA DE COMPUTADORES

Acronym

Language of instruction Português-PT or Inglês-EN

Teaching/Learning modality Blended Learning

Coordinating teacher António Eduardo de Barros Ruano

Teaching staff	Type	Classes	Hours (*)
António Eduardo de Barros Ruano	T; TP	T1; TP1	30T; 30TP

* For classes taught jointly, it is only accounted the workload of one.

Contact hours

T	TP	PL	TC	S	E	OT	O	Total
30	30	0	0	0	0	0	0	168

T - Theoretical; TP - Theoretical and practical ; PL - Practical and laboratorial; TC - Field Work; S - Seminar; E - Training; OT - Tutorial; O - Other

Pre-requisites

no pre-requisites

Prior knowledge and skills

Basic Programming and basic Algebra

The students intended learning outcomes (knowledge, skills and competences)

At the end of this lecture course the student should:

- Understand the biological inspiration of neural networks, their main characteristics and applications;
- To know the multi-layer perceptrons and radial base function networks, as well as the most common techniques of supervised training and adaptation; know how to use them in the context of systems approximation and classification;
- To know the most common unsupervised training methods, based on Hebbian rules and on competition, and the Kohonen networks; know how to use them in data clustering and classification problems.
- To know the main recurrent neural networks, and their training algorithms; know how to use them in practical applications;
- To know the main mechanisms of deep learning; know how to use deep belief and convolutional networks in practical applications;

Syllabus

1. Introduction to Artificial Neural Networks
 2. Supervised Learning for Feedforward Neural Networks
 - a. Models
 - b. Learning Algorithms
 - c. Applications to function approximation and pattern recognition
 3. Unsupervised Learning
 - a. Non-Competitive Learning
 - b. Competitive Learning
 4. Recurrent Neural Networks
 - a. Introduction to nonlinear dynamic systems
 - b. Models
 - c. Training and Operation
 - d. Applications of Recurrent Neural Networks
 5. Deep Learning
 - a. Revisiting Gradient Descent Algorithms
 - b. Restricted Boltzmann Machines and Deep Belief Networks
 - c. Convolutional Neural Networks
-

Demonstration of the syllabus coherence with the curricular unit's learning objectives

The objectives correspond to specific chapters / sections of the syllabus

Teaching methodologies (including evaluation)

Classes will be distributed by Theoretical (T) and Laboratory Practice (PL) components. In lectures T the theory will be exposed, and examples of application will be presented. In PL classes students will implement programs in Matlab or Python, using the available modules in each of these languages.

In this discipline there will be a mini-project, with a weight of 50% of the grade, and an exam, carried out in the normal period, with equal weight for the final grade. If the final grade is equal to or higher than 10, and both components are graded with a mark higher than 8, the student passes.

If the final grade is less than 10 and the mini-project grade is higher than 8, the student is admitted to the resource exam, which will replace the regular exam mark

Demonstration of the coherence between the teaching methodologies and the learning outcomes

This course has a classic format, with theoretical classes in amphitheater and practical classes in a programming laboratory. Theoretical classes express the unity of the course and subliminally intend to lead the students towards the learning objectives. More explicitly, the teacher explains the problems, the computational methods used to solve them, and their implementations in Matlab or Python.

The practical classes work in an open laboratory regime. The work to be done will be published on the website of the course. Students work in class, but especially outside of class. To encourage collective participation, students are encouraged to intervene in the course's forums to discuss issues related to proposed problems and other general questions

Main Bibliography

Apontamentos on-line

Ruano, A. E. (Ed.). Intelligent Control using Intelligent Computational Techniques. 2005: The Institution of Electrical Engineers.

Gori, M., Machine Learning. A Constraint-based Approach. 2018: Morgan Kaufmann.

Haykin, S.O., Neural Networks and Learning Machines. 3 ed. 2008: Prentice Hall.

Aggarwal, C.C., Neural Networks and Deep Learning. A Textbook. 2018: Springer.

Buduma, N. and N. Locascio, Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithms. 2017: O'Reilly Media.

Skansi, S., Introduction to Deep Learning: From Logical Calculus to Artificial Intelligence. 2018: Springer International Publishing.