COURSE DESCRIPTION CARD - SYLLABUS

Course name

Foundations of deep neural networks

Course

Field of study
Computing

Year/Semester
1/1

Area of study (specialization)
Edge Computing

Profile of study
general academic

Level of study
Second-cycle studies

Course offered in
Polish

Form of study
full-time

Requirements
compulsory

Number of hours

Lecture
20

Laboratory classes
30

Other (e.g. online)
0

Tutorials
0

Projects/seminars
0

Number of credit points

4

Lecturers

Responsible for the course/lecturer:

Szymon Szczęsny, PhD. Eng.

email: szymon.szczesny@put.poznan.pl

tel. 61 6652297

Faculty of Computing and Telecommunications

Piotrowo 3a st. 60-965 Poznan

Responsible for the course/lecturer:

Damian Huderek, MSc. Eng.

email: damian.huderek@put.poznan.pl

tel. 61 6652297

Faculty of Computing and Telecommunications

Piotrowo 3a st. 60-965 Poznan

Prerequisites

A student starting this course should have basic knowledge of computational logic, algorithm theory and complex data structures. An additional requirement is the ability to integrate basic knowledge from various fields related to computer science. It is also necessary to be able to expand knowledge and work in a team. Concerning social competencies, the student should be aware that knowledge in computer science quickly becomes obsolete and requires constant expansion. The student should present an attitude of honesty, creativity, reliability and cognitive curiosity.

Course objective

1. Passing knowledge in the field of artificial intelligence based on neural networks.

2. Presenting basic structures of neural networks used in embedded systems and edge computing.

3. Discussing using neural networks in data processing, analysis of signals from sensors, pattern classification, feature detection, etc.

4. Acquainting with basic tools for designing, training and implementing artificial neural networks.

5. Acquainting with limitations of hardware implementations of neural networks in edge systems.

6. Developing students' skills to process complex data using neural network algorithms.

7. Teaching the ability to develop dedicated IT systems based on elements of artificial intelligence, taking into account limitations resulting from the nature of interdisciplinary problems.

Course-related learning outcomes

Knowledge

1. has ordered and theoretically founded general knowledge related to key issues in the field of computer science [K2st_W2]

2. has advanced detailed knowledge of selected issues in the field of computer science [K2st_W3]

3. knows development trends and the most important new achievements of IT and other selected related scientific disciplines [K2st_W4]

4. knows advanced methods, techniques and tools used for solving complex engineering tasks and conducting research in selected areas of computer science [K2st_W6]

Skills

1. can obtain information from literature, databases and other sources (in Polish and English), integrate them, interpret and critically evaluate them, draw conclusions and formulate, as well as exhaustively justify opinions [K2st_U1]

2. can plan and carry out experiments, including measurements and computer simulations, interpret obtained results and draw conclusions, formulate and verify hypotheses related to complex engineering problems, as well as simple research problems [K2st_U3]

3. can use analytical, simulation and experimental methods to formulate and solve engineering tasks, as well as simple research problems [K2st_U4]

4. can - when formulating and solving engineering tasks - integrate knowledge from various areas of computer science (and, if necessary, also knowledge from other scientific disciplines) and apply a system approach, also taking into account non-technical aspects [K2st_U5]

5. can - using, among others, conceptually new methods - solve complex IT tasks, including non-standard tasks and tasks including a research component [K2st_U10]

Social competences

1. understands that in computer science knowledge and skills become obsolete very quickly [K2st_K1]

2. understands the importance of using the latest knowledge in the field of computer science in solving research and practical problems [K2st_K2]

3. understands the importance of popularizing the latest achievements in the field of computer science [K2st_K3]

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Formative assessment:

a) concerning lectures:

– based on answers to questions concerning the material discussed during previous lectures

b) concerning laboratories:

– based on assessment of current progress of implementation of tasks

Summative assessment:

a) concerning lectures, assumed learning outcomes are verified with:

– assessing knowledge and skills demonstrated in a problem-based written test

– discussing the results of the final test

b) concerning laboratories, assumed learning outcomes are verified with:

– evaluating the current progress in the implementation of tasks to be performed as part of laboratory restorative exercises (the student carries out the exercise according to the provided instructions);

– obtaining additional points for activity during classes, especially for: discussing additional aspects of the issue, the effectiveness of applying the acquired knowledge when solving a given problem, comments related to the improvement of didactic materials, indicating students' perceptual difficulties enabling the ongoing improvement of the teaching process.

Programme content

The lecture program covers the following topics:

Biological foundations of neural networks, history of research on biological networks, mathematical foundations of fuzzy systems, division of neural networks into three generations, models of neurons (perceptron, sigmoidal, radial, adaline, WTA), learning methods (gradient methods, Hebb's rule, pseudo-inversion matrix, Levenberg–Marquardt algorithm, back propagation, global optimization, genetic algorithms), unidirectional architectures of neural networks, radial networks, Hopfield associative memories, Hamming network, BAM network, recurrent networks (RMLP, Elman, RTRN), self-organizing networks and their learning algorithms (Kohonen, Sammon), deep neural networks (convolutional CNN, DBN, recurrent LSTM), complexity of the neural network algorithm, aspects of pipeline and parallel processing in neural networks, hardware implementation in digital systems (TinyML), hardware implementation in ASIC dedicated circuits, semiconductor implementations of second-generation neural networks, discussion of the application of neural networks in edge systems, neural networks in the task of analyzing data from sensors, implementation of preprocessors and classifiers, methods of classifier evaluation, error functions, ROC curves, confusion matrices.

Laboratory classes are conducted in the form of thirty hours of laboratory exercises, preceded by a 2-hour instructional session at the beginning of the semester. All laboratory classes are conducted by 2-person teams of students. The laboratory program covers the following topics: basics of TensorFlow and Keras tools, basic architectures and training algorithms of neural networks, main problems in training neural networks and how to solve them, methodology of developing models of neural networks, optimization of models of neural networks.

Teaching methods

1. Lecture: multimedia presentation supplemented with examples on a board

2. Laboratory exercises: solving tasks, practical exercises, data analysis, simulation, discussion, teamwork, case study, multimedia presentation.

Bibliography

Basic

1. S. Osowski, Sieci neuronowe do przetwarzania informacji, OWPW, 2020

2. A. W. Trask, Zrozumieć głębokie uczenie, PWN, 2019

3. R. Bharath, S. Leszek, Głębokie uczenie z TensorFlow : od regresji liniowej po uczenie przez wzmacnianie, Helion, 2020

Additional

1. S. J. Russell, P. Norvig, Artificial intelligence: a modern approach, Pearson, 2021

2. S. Szczęsny, 0.3 V 2.5 nW per channel current-mode CMOS perceptron for biomedical signal processing in amperometry, IEEE Sensor Journal 17(17), pp. 5399-5409, 2017

3. S. Szczęsny, High Speed and Low Sensitive Current-Mode CMOS Perceptron, Microelectronic Engineering 165, pp. 41-51, 2016

Breakdown of average student's workload

	Hours	ECTS
Total workload	100	4,0
Classes requiring direct contact with the teacher	50	2,0
Student's own work (literature studies, preparation for test) [1]	50	2,0

1 delete or add other activities as appropriate