COURSE DESCRIPTION CARD - SYLLABUS

Course name
Data processing with GPU [S2Inf1-PB>PDGPU]

Course

Field of study Computing	Year/Semester 1/2
Area of study (specialization) Edge Computing	Profile of study general academic
Level of study second-cycle	Course offered in polish
Form of study full-time	Requirements compulsory

Number of hours

Lecture 20	Laboratory classes 15	Other (e.g. online) 0
Tutorials 0	Projects/seminars 15	

Number of credit points

4,00

Coordinators

mgr inż. Paweł Pietrzak
pawel.pietrzak@doctorate.put.poznan.pl

dr inż. Adam Turkot
adam.turkot@put.poznan.pl

Lecturers

Prerequisites

Before taking the course students should be acquainted with basics of neural networks and edge systems. Apart from that students should also be able to write programs in Python programming language with the use of Tensorflow framework. As for the social competencies, the ability to work in a team is necessary.

Course objective

The goal of the course is to teach students accelerated computing using a GPU. Computations to be accelerated are neural networks' inference and training. Data to be processed are images and video sequences. Students will learn GPU-based optimizations for neural networks as well as deployment, advanced convolutional neural networks architectures, image data preparation and strengthen existing knowledge of deep learning methodologies.

Course-related learning outcomes

Knowledge

1.Has advanced and deepened knowledge of broadly understood information systems, theoretical foundations of their construction, and methods, tools and environments
2. Has well-ordered and theoretically grounded general knowledge connected with the key issues of computer science
3. Knows advanced methods, techniques and tools applied in solving complex engineering tasks and carrying out research work in the selected field of computer science.

Skills
1. Is able to acquire information from literature, databases and other sources (in Polish and English), integrate it, interpret and critically evaluate it, draw conclusions and formulate and exhaustively justify opinions.
2. Is able to use analytical, simulation and experimental methods to formulate and solve engineering tasks and simple research problems
3. Is able to integrate knowledge from different fields of computer science (and if necessary from other scientific disciplines) and apply a system approach, taking into account also non-technical aspects, when formulating and solving engineering tasks
4. Is able to make a critical analysis of existing technical solutions and propose their improvements (enhancements)
5. Is able to solve complex information technology tasks, including atypical tasks and tasks with a research component, using new conceptual methods.
6. Is able to design a complex device, information system or process to a given specification taking into account non-technical aspects and implement the design - at least in part - using appropriate methods, techniques and tools including adapting existing tools or developing new ones.

Social competences
1. Understands that in computer science knowledge and skills become obsolete very quickly.
2. Understands the importance of using the latest knowledge in computer science to solve research and practical problems.

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Formal evaluation: (a) for lectures: on the basis of answers to questions concerning the material discussed in previous
3
lectures (b) for laboratorys/exercises: based on an assessment of the current progress of the tasks,
Summary evaluation: a) in the scope of lectures, the verification of the assumed educational results is carried out by an electronic test on the Moodle platform; b) oral examination combined with the defence of a project prepared by students in a group

Programme content

GPU acceleration, Convolutional neural networks' architectures, data preparation, neural networks' models deployment, neural networks' models optimization, Tensorflow framework, TensorRT model optimizer, GPU-based edge devices, deep learning applications in computer vision domain

Teaching methods

lecture: multimedia presentation, presentation illustrated with examples given on the board, presentation of selected student solutions. laboratory/project: performing tasks and experiments on hardware and software provided for the laboratory, working in groups

Bibliography

Podstawowa 1. Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow, Géron, Aurélien, O'Reilly UK Ltd., 2019
Uzupełniająca 1. Mastering Computer Vision with TensorFlow 2.x: Build advanced computer vision applications using machine learning and deep learning techniques, Krishnendu Kar, Packt Publishing, 2020

Breakdown of average student's workload

	Hours	ECTS
Total workload	100	4,00
Classes requiring direct contact with the teacher	50	2,00
Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation)	50	2,00