WASP
WALLENBERG AI,
AUTONOMOUS SYSTEMS
AND SOFTWARE PROGRAM

Syllabus
Deep learning for Natural Language Processing, 6hp

Issued by the WASP graduate school management group

Main field of study

AI/MLX

Course level

PhD student course

Course offered for

PhD Students in the WASP graduate school

Entry requirements

The participants are assumed to have a background in mathematics corresponding to the contents of the WASP-course “Introduction to Mathematics for Machine Learning”. The course requires solid programming experience in a high-level language; the programming assignments will use Python. Students are expected to be comfortable with modern deep learning techniques and frameworks, for instance as taught by the WASP course Deep Learning and GANs.
No previous knowledge of NLP is required.

Intended learning outcomes

Natural Language Processing (NLP) develops methods for making human language accessible to computers. The goal of this course is to provide students with a theoretical understanding of and practical experience with the advanced algorithms that power modern NLP. The course focuses on methods based on deep neural networks.

On completion of the course, the student should be able to

- explain and analyse state-of-the-art deep learning architectures for NLP
- implement such architectures and apply them to practical problems
- design and carry out evaluations of deep learning architectures for NLP
- use current approaches to NLP in the student’s own field of research

Course content

The course content is presented in three modules:

- Introduction to deep learning for NLP. Overview of large language models (LLMs). Tokenisation and embeddings. Attention. LLM architectures, including GPT and BERT.
- Introduction to LLM development. Pretraining. Instruction fine-tuning. Alignment with human preferences. Data. Societal, environmental and ethical aspects.
- Current research in NLP. Applications of NLP in other research areas.

Each module presents relevant validation methods, tools, software libraries, and datasets.

Teaching and working methods

Each module features:

- video lectures introducing technical concepts
- reading assignments
- a set of non-examinatory programming exercises and discussion tasks
- interactive sessions where students apply and deepen their understanding

For the first two modules, there are also examinatory programming assignments. The final assignment is a project where students apply their learning to their own field of research.

Examination

4 programming assignments, 1 self-defined project

Grades

Fail or Pass