Donders Institute
for Brain, Cognition and Behaviour

Neural Information Processing Systems

Course Reader

Neural Information Processing Systems

Organized by the Artificial Intelligence Department
Donders Institute for Brain, Cognition and Behaviour

Teachers
Dr. Umut Güçlü; E: u.guclu@donders.ru.nl
Prof. dr. Marcel van Gerven; E: m.vangerven@donders.ru.nl

Table of Contents

Aims

After successful completion of the course, students:

- Know about current advances in modern neural networks such as deep learning, recurrent neural networks, reinforcement learning and generative modeling.
- Are able to implement advanced neural networks in the Python programming language.
- Understand how neural networks can be used to model cognitive processes.

Content

A main objective of artificial intelligence is to build machines whose cognitive abilities match (or surpass) those of humans. This is also referred to as artificial intelligence. One way to achieve this goal is by developing cognitive architectures that mimic the algorithms used by our own brains. This success of such an approach relies on a continuous interplay between AI and neuroscience.

In this course, we will explore how modern neural networks provide us with the tools to model cognitive processes in artificial systems and understand cognitive processes in biological organisms.

The course consists of different components:

- During the lectures, students will get acquainted with the theoretical basis and practical development of advanced neural networks. This will be done via presentation and discussion of key papers.
- During the practical sessions, students will learn to implement neural network approaches, related to specific papers discussed in class. To this end, the Python programming language will be used.

Instructional modes

Lecture
Practical sessions

Test information

Practical assignments 100%

Prerequisites

- Python programming experience
- Mathematical skills at the AI Bachelor level (probability theory, calculus and linear algebra)
- Basic knowledge of neural networks

Schedule (subject to change)

- Week 1: **Introduction**
- Week 2, 3: **Computer vision** (example topics: object recognition, semantic segmentation, image super-resolution, texture synthesis - paper will be announced prior to the lecture)
- Week 4, 5: **Computer audition** (example topics: music information retrieval, music transcription, speech recognition, speech synthesis - paper will be announced prior to the lecture)
- Week 6, 7 (guest lecture by L. Ambrogioni): **Natural language processing** (example topics: automatic summarisation, question answering, machine translation, part-of-speech tagging - paper will be announced prior to the lecture)
- Week 8, 9 (guest lecture by Dr. Y. Güçlütürk): **Computational creativity** (example topics: algorithmic composition, natural language generation, procedural generation, style transfer - paper will be announced prior to the lecture)
- Week 10, 11: **Neural coding** (example topics: decoding - classification, identification, reconstruction - encoding - paper will be announced prior to the lecture)
- Week 12, 13: **(Toward) artificial general intelligence** (example topics: chatbots, game artificial intelligence, general game playing, intelligent agents - paper will be announced prior to the lecture)
- Week 14: **Conclusion**

Course load (168 hours)

- 12 hours preparation for lectures (8 weeks x 1.5 hours)
- 12 hours lecture attendance (8 weeks x 1.5 hours)
- 18 hours self study for lectures (6 weeks x 3 hours)
- 36 hours preparation for practicals (12 weeks x 3 hours)
- 18 hours practical attendance (12 weeks x 1.5 hours)
- 72 hours self study for practicas (12 weeks x 6 hours)

Lectures

The first lecture will introduce the course and give a broad overview of the topics. To prepare for this lecture, please read the following paper in advance:

http://www.biorxiv.org/content/early/2017/07/25/166785

In each of the lectures 2-7, a key paper dealing with artificial neural networks will be explained and discussed in depth. The papers will be selected from top conferences such as NIPS, ICML, ICLR, CVPR, ICCV, ECCV, etc. or journals such as Science, Nature, Journal of Neuroscience, PLOS Computational Biology, etc. You will be expected to prepare for these lectures by reading the papers in advance and actively participate in the ensuing discussion. The discussions will focus on one or more of the following questions:

- Algorithmic breakthroughs: What algorithmic advance in neural network research is proposed?
- Theoretical insight: What does the paper offer in terms of theoretical understanding of neural network models?
- Neural network applications: What problem does the paper solve and how does it compare to non neural network approaches?
- Understanding natural intelligence: What unique viewpoint does the paper offer about understanding natural intelligence?
- Achieving strong AI: What unique viewpoint does the paper offer about achieving strong AI?
- Disruptive technology: What kind of disruptive technology does the paper propose/advance?

The last lecture will discuss the topics and conclude the course.

Practical sessions

During the practical sessions you will work in groups of three and implement various artificial neural networks from the papers that will be explained/discussed in the lectures. Make sure to evenly divide the load between group members. You will preferably make use of the Chainer framework (https://chainer.org). You may already consult the documentation to prepare yourself. You are also allowed to make use of other frameworks such as PyTorch or TensorFlow. However, we provide minimal support for these frameworks in terms of implementation details.

Hand-in of assignments

Practical assignments must be handed in as a jupyter notebook file via Brightspace per group. You can modify your submission until the submission deadline has passed. Only the final version will be graded. In solving your assignments, you are free to consult resources from the Internet. However, your source code needs to be your own. Copy/paste of the source code of other groups or internet resources will be considered plagiarism.

You can expect the following from the practical assistants:

- General hints on how to solve an assignment in case you are stuck.
- Pointing out mistakes that are spotted during the practical
- Feedback on the correct solution of a previous assignment after the deadline for that assignment has passed

Practical assistants will not provide completely worked out solutions of previous assignments

Grading of the assignments

Grading of the practical assignments will be based on the following evaluation criteria:

1. Quality of the content: Does the report look clean and understandable? Are implementations theoretically solid and computationally efficient? Is the code well written, well documented and easy to understand? Assignments **must** be handed in as a jupyter

notebook.

2. Insight: do the answers to the assignments convey that the subject matter has been understood?
3. Independence: Could the assignments be completed without relying too extensively on input from the assistants?

Practical assistants

For any questions regarding the lectures and practicals, **use the questions forum**. The practical assistants can be approached in case forum questions are not addressed. The coordinator may be contacted for questions during the lectures (by email only in case issues persist or in case of personal issues).

Second chance policy

The second chance will involve working on a project, which investigates how human-like intelligence can be realized in synthetic brains by implementing brain-inspired computing machinery to solve problems related to this topic. Projects can be theoretical or applied in nature and geared towards AI or neuroscience.

The goal is to work from the inception of an idea towards a finalized research paper that addresses your research question. Prior to starting your project, the idea must be approved by the teachers. To this end, you have to submit a short proposal. In case dedicated hardware is needed to estimate models, resources can be provided depending on availability. Example topics could be:

- Any topic/idea that motivates you and provides a realistic path towards a final paper
- Replication of a classical model (e.g. sparse or predictive coding models)
- Replication of a new model proposed in a recent conference publication
- Analysis of open-source neural datasets using computational approaches
- Implementation of a neural network which is able to play games in the openAI environment
- Implementation of a neural network which captures some property of natural intelligence
- Investigation of theoretical properties of neural networks
- Implementation of spiking neural network models proposed in literature
- Improvement of existing architectures/algorithms

The final product will be a working and well-documented implementation which solves an open problem in computational neuroscience or artificial intelligence as well as a NIPS style paper (see nips.cc for guidelines). Your final grade will be based on the quality your paper and source code, which will be evaluated in terms of clarity, novelty, relevance and sophistication.

Special circumstances

In case of special circumstances please contact the course coordinator as soon as possible.

Reading material

Reading material will be announced prior to the respective lectures.

Background material

If you have no background in neural networks or Python, it is up to you to acquire basic knowledge. A Python tutorial will be provided on Brightspace.

For additional background reading on the course topic we can advise the following review papers:

1. Nilsson N. Human-level artificial intelligence? Be serious! AI Mag. 2005;26(4):68–75.
2. LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521, 436–444. doi:10.1038/nature14539
3. Lake, B.M., Ullman, T.D., Tenenbaum, J.B., Gershman, S.J., 2016. Building Machines That Learn and Think Like People 2, 1–44.
4. Gershman SJ, Horvitz EJ, Tenenbaum JB. Computational rationality: A converging paradigm for intelligence in brains, minds, and machines. 2015;349(6245).
5. James L McClelland, Matthew M Botvinick, David C Noelle, David C Plaut, Timothy T Rogers, Mark S Seidenberg, and Linda B Smith. Letting structure emerge: connectionist and dynamical systems approaches to cognition. Trends Cogn. Sci., 14(8):348–356, 2010
6. Hassabis, D., Kumaran, D., Summerfield, C., & Botvinick, M. (2017). Neuroscience-inspired artificial intelligence. *Neuron*, *95*(2), 245–258. http://doi.org/10.1016/j.neuron.2017.06.011.
7. van Gerven MAJ. A primer on encoding models in sensory neuroscience. J Math Psychol. Elsevier Inc.; 2017;76(B):172–83.
8. Blei, D. M., & Smyth, P. (2017). Science and data science, *114*(33), 8689–8692. http://doi.org/10.1073/pnas.1702076114

To learn more about neural networks, you should consult:

- http://playground.tensorflow.org
- http://cs231n.stanford.edu
- http://neuralnetworksanddeeplearning.com
- http://www.andreykurenkov.com/writing/a-brief-history-of-neural-nets-and-deep-learning/
- http://deeplearningbook.org

For new advances in neural network research you may also consult the following blogs:

- https://blog.openai.com
- https://deepmind.com/blog/