

**Module Description, available in: EN**

## Deep Learning

### General Information

**Number of ECTS Credits**

3

**Module code**

TSM\_DeLearn

**Valid for academic year**

2020-21

**Last modification**

2019-01-30

**Coordinator of the module**

Jean Hennebert (HES-SO, jean.hennebert@hefr.ch)

**Explanations regarding the language definitions for each location:**

- Instruction is given in the language defined below for each location/each time the module is held.
- Documentation is available in the languages defined below. Where documents are in several languages, the percentage distribution is shown (100% = all the documentation).
- The examination is available 100% in the languages shown for each location/each time it is held.

	Lausanne			Lugano	Zurich		
<b>Instruction</b>			X E 100%		X E 100%		
<b>Documentation</b>			X E 100%		X E 100%		
<b>Examination</b>			X E 100%		X E 100%		

**Module Category**

TSM Technical scientific module

**Lessons**

2 lecture periods and 1 tutorial period per week

### Entry level competences

**Prerequisites, previous knowledge**Linear algebra: vector and matrix operations, Eigenvectors and  $\lambda$ -values

Multivariate calculus: partial differentiation, chain rule, gradient, Jacobian and Hessian

Statistics and probability theory: discrete and continuous distributions, multi-variate distributions, probability mass and density functions, Bayes' Rule, maximum likelihood principle

Programming: Experience in a programming language with good understanding of loops and data structures such as arrays/lists and maps/dictionaries; understanding of object oriented programming concepts. The course is taught using Python.

## Brief course description of module objectives and content

Deep Learning is one of the most active subareas of Machine Learning and Artificial Intelligence at the moment. Gartner has placed it at the peak in its 2017 Hype Cycle and the trend is going on. Deep Learning techniques are based on neural networks. They are at the core of a vast range of impressive applications, ranging from image classification, automated image captioning, language translation such as Google Translate, to playing Go and arcade games.

This course focuses on the mathematical aspects of neural networks, their implementation (in Python), and their training and usage. Students will learn the fundamental concepts of Deep Learning and develop a good understanding of applicability of Deep Learning for Machine Learning tasks. After completing the course, students will have developed the skills to apply Deep Learning in practical application settings.

## Aims, content, methods

### Learning objectives and acquired competencies

Students will

- have a thorough **understanding of neural network architectures** including convolutional and recurrent networks.
- know **loss functions** (e.g. categorical cross entropy) that provide the optimization objective during training.
- understand the principles of **back propagation**.
- know the benefits of **depths and representation learning**.
- know some of the **recent advances** in the field and some of the **open research questions**.
- develop the ability to decide **whether Deep Learning is suitable** for a given task.
- gain the ability to **build and train neural network models** in a Deep Learning Framework such as TensorFlow.

### Contents of module with emphasis on teaching content

- **Introduction:** Logistic Neuron, training and cost functions.  
Architectures: Feed-forward and recurrent networks. Applications of neural networks.
- **Optimization strategies:** Minimization of loss functions, gradient descent, stochastic gradient descent, mini-batch gradient descent, implementation of gradient decent optimizers in Python.
- **Training of Deep Neural Networks:** Backpropagation, computational graphs, automatic differentiation, special optimizers, such as Nesterov accelerated gradient, AdaGrad, or RMSProp; tricks for faster training, batch normalization, gradient clipping, special activation functions such as non-saturating activation functions, regularization using dropout.
- **Multilayer Perceptron (MLP):** implementation of an MLP including backpropagation in Python.
- **Convolutional Neural Networks (CNNs):** Convolutional and pooling layers, data augmentation, popular CNN architectures, transfer learning, applications.
- **Practical Considerations and Methodology:** Deep Learning frameworks such as TensorFlow; gpu vs cpu; visualizations such as activation maximization, class activation maps, saliency maps; performance metrics, selecting hyper-parameters, debugging strategies.
- **Recurrent Neural Networks:** Vanishing and exploding gradients, special memory cells, such as Gated Recurrent Units (GRU) or Long short-term memory (LSTM), static and dynamic unrolling, sequence classifiers, sequence-to-sequence models, encoder-decoder for language translation.
- **Special and Current Research Topics** such as
  - Autoencoders: principal component analysis using autoencoders; special applications such as denoising auto-encoders.
  - Generative Adversarial Models.
  - Learning embeddings for word representations, attention mechanism, transformers.

### Teaching and learning methods

Classroom teaching; programming exercises

### Literature

- I. Goodfellow, Y. Bengio, A. Courville: "Deep Learning", MIT Press, 2016. ISBN: 978-0262035613.
- N. Buduma: "Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithms", O'Reilly, 2017. ISBN: 978-1491925614.
- A. Géron, Hands-on Machine Learning with Scikit-Learn and TensorFlow, O'Reilly, 2017 ISBN: 978-1491962299.
- C. M. Bishop: "Neural Networks for Pattern Recognition". Clarendon Press. 1996. ISBN: 978-0198538646.
- K. P. Murphy, "Machine Learning, A Probabilistic Perspective", MIT Press, 2012, ISBN: 9780262018029
- T. M. Mitchell, "Machine Learning", McGraw-Hill Science/Engineering/Math, 1997, ISBN: 0070428077

## Assessment

### Certification requirements

Module uses certification requirements

### Certification requirements for final examinations (conditions for attestation)

75% of handed-in homework passed

### Basic principle for exams

**As a rule, all the standard final exams for modules and also all resit exams are to be in written form**

### Standard final exam for a module and written resit exam

#### Kind of exam

written

#### Duration of exam

120 minutes

#### Permissible aids

*Aids permitted as specified below:*

#### Permissible electronic aids

No electronic aids permitted

#### Other permissible aids

1 A4 page (front and back) of handwritten notes

### Special case: Resit exam as oral exam

#### Kind of exam

oral

#### Duration of exam

30 minutes

#### Permissible aids

No aids permitted