technische universität dortmund

PD Dr. Andriy Sokolov
Institut für Angewandte Mathematik, LS III
TU Dortmund
Vogelpothsweg 87, 44227 Dortmund
E-Mail: andriy.sokolov@math.tu-dortmund.de
Tel: +49-(0)231-755-3171
Fax: +49-(0)231-755-5933

**Veranstaltungsankündigung WS 25/26**

# Introduction to Deep Learning and Its Applications

**Zeit und Ort:** ???

**Veranstaltungstyp:** Vorlesung (??+??)

**Inhalt:**

The study of deep learning and neural networks is of growing importance in today's world, as artificial intelligence (AI) increasingly impacts many aspects of our lives — from autonomous vehicles and medical diagnostics to financial forecasting and robotics. At the core of these technologies lie mathematical methods that enable machines to learn from vast amounts of data. In this course, we delve into the theoretical aspects of deep learning, exploring its mathematical foundations, training algorithms, and network structures, while also addressing related problems such as convergence, stability, and interpretability.

We combine theory with practice by implementing neural networks ourselves, using the PyTorch framework — a powerful tool that lets us efficiently construct, train, and deploy deep models. During the practical sessions, we apply these methods to solve real-world problems, such as image recognition with convolutional neural networks and handwritten digit recognition using the MNIST dataset. This approach prepares students to understand the fundamental mathematical aspects of deep learning and to gain practical experience in implementing and testing AI-driven applications.

1) Neural Networks (NNs)
   a. Types and structures (feedforward, recurrent, convolutional, etc.).
   b. Approximation using various activation functions (ReLU, sigmoid, tanh, softmax, etc.).
   c. Architectural components (layers, nodes, connections, weights, biases).
   d. Loss functions and their role in training.
   e. Hyperparameter selection (number of layers, number of neurons, learning rate, batch size).

2) Algorithms for Training NNs
   a. Backpropagation algorithm (chain rule, error signals, weight update mechanisms).
   b. Optimizers (SGD, momentum, AdaGrad, RMSProp, Adam, etc.).
   c. Regularization techniques (dropout, L1, L2 penalties) to avoid overfitting.
   d. Validation and testing strategies (train-test split, cross-validation).
3) Implementation of NNs Using Torch (Python)
   a. Introduction to Torch's modules and components (torch.nn, torch.optim).
   b. Model definition, forward pass, and training loop.
   c. Loading data, data loaders, batches, augmentations.
4) Convolutional Neural Networks (CNNs) for Image Recognition
   a. Convolutional layers, pooling, padding, stride.
   b. Architectural patterns (ResNet, AlexNet).
   c. Applications to object recognition, face recognition and pose detection
5) Practical Implementation
   a. A simple deep learning example — handwritten digit recognition using the MNIST dataset.
   b. Image color recognition with a small custom dataset.
   c. End-to-end pipeline: data preparation, training, evaluation, and visualization of results.
   d. Implementing data augmentation and fine-tuning hyperparameters.
6) Various Mathematical and Practical Aspects of NNs and Deep Learning
   a. Loss landscape, convergence, and stability of training.
   b. Vanishing and exploding gradient problems and their solutions.
   c. Computational complexity.

**Sprache:** English

**Anforderung** Students are expected to have a solid understanding of Linear Algebra and Numerical Methods I, as well as a basic or intermediate knowledge of Python.

**Bemerkungen:** Students are recommended to have their own laptop with Python environments installed, preferably version 3 or higher, and a Python editor such as Jupyter Notebook, Visual Studio Code, or a similar tool. It is also desirable to have Git-related software installed.

**Lehrinhalte** In this course, students will learn to understand the mathematical foundations of deep learning and neural networks. They will acquire practical skills in implementing and training deep models using the PyTorch framework. Additionally, they will be able to apply these methods to solve real-world problems.

**Kompetenzen** These competencies can significantly facilitate students' future careers in both

| | |
|---|---|
| | science and industry, enabling them to apply deep learning methods effectively in a wide range of problems and innovations. |
| **Empfohlene Literatur:** | To be announced |
| **Übung zur Veranstaltung:** | Nummer der Übung ??? |