Hands-on Machine Learning Training

Session 6 – Backpropagation

Preparation

Based on your knowledge from the last session as well as from the supplementary material, you will implement the backpropagation algorithm on your own in this session. In the end, you will write all necessary code to train a complete neural network from scratch. By now, you should already be familiar with the basic building blocks of a neural network, as well as with Python and NumPy.

In order to prepare for the session you should:

- Work trough and understand the following book chapter (including the optional sections): Michael Nielsen: "Neural Networks and Deep Learning", Chapter 2 [1].
- Understand the source code in this chapter
- Look at the exercises in this book chapter on your own or with your teammates

We will not control if you have worked through the exercises. However, understanding and solving these exercises could help you get a better understanding of the complicated backpropagation equations. This will definitely be of use during the quicktest and the following session, as you will need to implement the equations in the book chapter. After working through the material and exercises you should know about:

- The basic concept of backpropagation
- The hadamard product
- The four fundamental equations of backpropagation
- How to measure the error of a neural network
- The backpropagation algorithm in detail
- How to update a neural network based on gradient descent

[1] `http://neuralnetworksanddeeplearning.com/chap2.html`

Further Reading

If you would like a deep dive into the topic, this section could be the first suggestions for it. They are not mandatory in this laboratory.

- You can find the paper [2] by Rumelhart, Hinton and Williams from 1986 about the initial idea of the backpropagation implementation
- You can find the videos [3] about backpropagation (and neural networks) with nice visualization from 3Blue1Brown
- You can find the recorded lecture [4] and the teaching materials on backpropagation from Stanford, including slides [5] and notes [6]
- You can find the doc of how the gradients are automatically calculated in PyTorch [7] and the further reading of it [8]

[2] Rumelhart, D., Hinton, G. & Williams, R. Learning representations by back-propagating errors. Nature 323, 533–536 (1986). https://doi.org/10.1038/323533a0

[3] 3Blue1Brown: https://www.youtube.com/playlist?list=PLZHQObOWTQDNU6R1_67000Dx_ZCJB-3pi

[4] Lecture: https://www.youtube.com/watch?v=d14TUNcbn1k

[5] Slides (backpropagation starting from Slide 49): http://cs231n.stanford.edu/slides/2020/lecture_4.pdf

[6] Notes: https://cs231n.github.io/optimization-2/

[7] PyTorch autograd: https://pytorch.org/tutorials/beginner/basics/autogradqs_tutorial.html

[8] PyTorch autograd further reading: https://pytorch.org/docs/stable/notes/autograd.html