Hands-on Machine Learning Training

Session 5 – Neural Networks

Preparation

In this session you will, step by step, implement a neural network yourself without using any deep learning libraries. After participating in the previous sessions you should already be familiar with Python as well as NumPy (a Python package for matrix calculations and scientific computing).

In order to prepare for the session you need to work through and understand the following web pages:

- Michael Nielsen: "Neural Networks and Deep Learning", Chapter 1. up to but not including the section "Implementing our network to classify digits" [1].

After working through the material you should understand:

- the structure of neural networks
- the step and sigmoid activation function
- (the formula for) the feed forward pass
- the mean square error criterion
- the idea of mini batches and stochastic gradient descent

Debugging

While implementing the exercise you will inevitably stumble upon Python's biggest strength, but discover it to be simultaneously Python's biggest weakness. Python is a weakly- and dynamically-typed language. This is obviously Python's biggest strength as a scripting language, since it allows one to quickly test ideas without worrying about the details of the data types. However, it can also be Python's biggest weakness, as implicit type coercion rules will inevitably lead to semantic bugs, which are difficult to debug. For example, a vector $\mathbf{x} \in \mathbb{R}^n$ is treated differently than a 2D matrix $\mathbf{X}^{n \times 1}$ by NumPy when applying broadcasting.

[1] `http://neuralnetworksanddeeplearning.com/chap1.html`

When implementing ML methods, you should perform sanity checks regularly, and never ever rely on unit-tests alone, since they cannot capture highly semantic bugs/the complex interplay at work when developing ML models.

Further Reading

For those of you interested in debugging Machine Learning models apart from catching dimension errors, we refer to:

- https://neptune.ai/blog/model-debugging-strategies-machine-learning