

Introduction to the Labs

Emilio Dorigatti

Labs

- ▶ Theory-oriented labs: math with pen and paper first, then code
- ▶ Practice-oriented labs: no math, less coding, play around
- ▶ Handouts in R, Python is also accepted (and encouraged)
- ▶ Not graded! Only final exam counts
- ▶ Solutions posted on Moodle the next Tuesday
 - ▶ Presented on Friday only on popular demand
- ▶ Come to the lab to:
 1. Ask questions
 2. Discuss and collaborate

Theory-oriented labs

- ▶ Apply the math you see in the lectures
 1. Pen-and-paper exercises on simple cases
 2. Code the solution and test on simple problems
- ▶ Goal: create your own Deep Learning Framework from scratch
- ▶ Purpose:
 1. Understand and internalize how things work
 2. Get practical experience in implementing ML algorithms
 3. Get ready for the exam

Practice-oriented labs

- ▶ Play with neural networks
 - ▶ github.com/jjallaire/deep-learning-with-r-notebooks
 - ▶ github.com/fchollet/deep-learning-with-python-notebooks
- ▶ Purpose:
 1. Learn to use Keras
 2. Get a feeling for how neural networks behave

Our software stack

1. Keras for R
2. Keras for Python
3. Tensorflow (Python bindings)
4. Tensorflow (C++)
 - ▶ BLAS (C/Fortran): CPU computation
 - ▶ cuDNN (C): GPU computation

Today's lab

Goal: get dev environment ready

- ▶ <https://keras.rstudio.com/>
- ▶ Test with the first notebook of “Deep Learning with R”
- ▶ Setting up tensorflow to use the GPU is a bit painful
 - ▶ Try with miniconda first
 - ▶ <https://docs.conda.io/en/latest/miniconda.html>
 - ▶ `conda install tensorflow-gpu`
 - ▶ Verify from Python:

```
import tensorflow as tf
cfg = tf.ConfigProto(log_device_placement=True)
sess = tf.Session(config=cfg)
```
 - ▶ Manual installation of cuDNN
 - ▶ <https://docs.nvidia.com/deeplearning/sdk/cudnn-install/index.html>
 - ▶ Then conda again