University of Pavia
Ph.D. School of Electrical and Electronics Engineering and Computer Science

Deep Learning and TensorFlow

A short course

May 3 - June 10, 2019

Instructor:
Marco Piastra, PhD, marco.piastra@unipv.it

Asssistants:
Mirto Musci, PhD, mirto.musci@unipv.it
Andrea Pedrini, PhD, andrea.pedrini@unipv.it
Gianluca Gerard, PhD candidate gianluca.gerard01@universitadipavia.it

The book “Deep Learning”, by I. Goodfellow, Y. Bengio and A. Courville, MIT Press, 2017, will be assumed as reference for these episodes. At present, an online, html version of the book is freely accessible online at http://www.deeplearningbook.org/.
For Keras and Python programming for Deep Learning in general: “Deep Learning with Python” by Francois Chollet, Manning, 2017
For TensorFlow, a good introductory book is “Hands-On Machine Learning with Scikit-Learn & TensorFlow” by Aurélien Géron, O’Reilly, 2017

Detailed Syllabus

1 - Lecture (3 hours), Friday May 3, 2:15 pm, Aula 4

Machine learning: Artificial Neural Networks

- An introductory example: linear regression
- Supervised machine learning: representation, evaluation, optimization
- Feed-forward networks as universal representation

- Evaluation function and numerical optimization
- Gradient descent, stochastic gradient descent, mini-batch gradient descent

- Feed-forward network as a flow graph
- Automatic differentiation: computing the gradient
- Tensorial representation, implicit operations (transposing, broadcasting)

2 - Lecture (3 hours), Friday May 10, 2:15 pm, Aula 4

Deep Learning for Neural Networks

- Deep network representation: fundamentals, potential advantages and major issues
- Layerwise representation as a flow graph (with layerwise gradient)
- From regression to classification: Softmax layer

- First order vs. second order optimization, potential advantages and major issues
- Quasi second-order methods: momentum, Nesterov, AdaDelta, AdaGrad, Adam

- The choice of non-linear functions: ReLU
- Empirical tricks: batch normalization, dropout

3 - Lecture (3 hours), Monday May 13, 2:15 pm, Aula 4

Deep Convolutional Neural Networks

- The ImageNet challenge
- Convolutional filters: representation and gradients

- Max pooling, normalizations: representation and gradients
- Advanced architectures: Inception and ResNet
- Transfer learning

- Visualizing representations in deep layers, merging
- Active learning, adversarial generation
- Beyond single-object classification: object location, segmentation (hints)

4 - Lecture (4 hours), Friday May 17, 2:15 pm, Aula 4

TensorFlow Fundamentals

- Sessions: computing flow graphs
- Tensors: constant, variables, placeholders and ops

- Tensor transformations: slicing, broadcasting, reshaping
- Optimization: loss functions, gradients, optimizers

5 - Lab (4 hours), Monday May 20, 2:15 pm, Aula B1

A first example with TensorFlow

- Graph and sessions
- Randomization
- Linear regression: stochastic gradient descent in detail and with an optimizer
- Running a session and visualizing results

7 - Lab (4 hours), Monday May 27, 2:15 pm, Aula B1

TensorFlow for DCNN (I)

- Logistic regression for MNIST
- DCNN for MNIST

- A bag of tricks: batch normalization, dropout, improved optimizers

6 - Lab (4 hours), Monday June 3, 2:15 pm, Aula B1
TensorFlow eager mode, Keras

- Graph mode and eager mode: differences
- A complete example: regression revisited
- From eager mode to graphs
- The Keras front end

8 - Lab (4 hours), Monday June 10, 2:15 pm, Aula B1
TensorFlow for DCNN (II)

- Dataset preparation: setup, tensor representation, mini batches
- Transfer learning: pre-trained models
- More tricks: data augmentation
- Fine tuning and network surgery: telling cats from dogs with Inception v3

Prerequisites (for the hands-on tutorial)

Previous knowledge of the topics below will be assumed as prerequisites:

- **Linux Ubuntu**: minimal survival skills
- **Python programming**: basic operations, lists, tuples, dictionaries, conditions, loops, functions, classes and instances, modules, exception handling (all code examples will be in Python 3)
- **Numpy basics**: array creation, basic operations and linear algebra (matmul), indexing, slicing, iteration, reshaping
- **PyCharm**: some experience is not mandatory but strongly recommended (with the debugger, in particular)

Organizers

Prof. Virginio Cantoni
Prof. Francesco Leporati

Ph.D. Coordinator

Prof. Paolo Di Barba