

# PyTorch: Working with Sequential Networks

Jerónimo Arenas-García

Universidad Carlos III de Madrid

*jeronimo.arenas@uc3m.es*

January 15, 2019

# Background and goals

## Last sessions

- Pytorch basics
- Autograd
- Adversarial nets example

In this session we will restrict our selves to feed-forward sequential nets:

- Focus on solving exercises
- Some additional topics will also be covered

We will study Lesson 4 from the Udacity course

<https://classroom.udacity.com/courses/ud188>

## Downloads

- Videos are available, but we will just go through the notebooks:  
git clone <https://github.com/udacity/deep-learning-v2-pytorch.git>
- Download also data for Notebook 7: [https://s3.amazonaws.com/content.udacity-data.com/nd089/Cat\\_Dog\\_data.zip](https://s3.amazonaws.com/content.udacity-data.com/nd089/Cat_Dog_data.zip)  
Save dataset in intro-to-pytorch folder

## Lesson 4: Intro to PyTorch

The folder contains the following auxiliary python scripts

- `helper.py`: Auxiliary functions to print images, display networks accuracy, output classification, etc ...
- `fc_model.py`: Implements feed-forward generic network with just linear layers.

The notebooks

- Part 1: Tensors in PyTorch
- Part 2: Neural Networks
- Part 3: Training Neural Networks (go directly to Losses section).  
Optimizers, Losses, autograd.
- Part 4: Fashion-Mnist
- Part 5: Inference and Validation (dropout; evaluation and train modes)
- Part 6: Saving and Loading models
- Part 7: Loading and saving models (torchvision review)
- Part 8: Transfer Learning

## Official tutorials

You can now complete very easily pending tutorials from the official site. In particular, 2 tutorials that we did not go through during the last session (not a lot of typing required).