

# Deep Learning - MAI

## Guided lab - CNNs

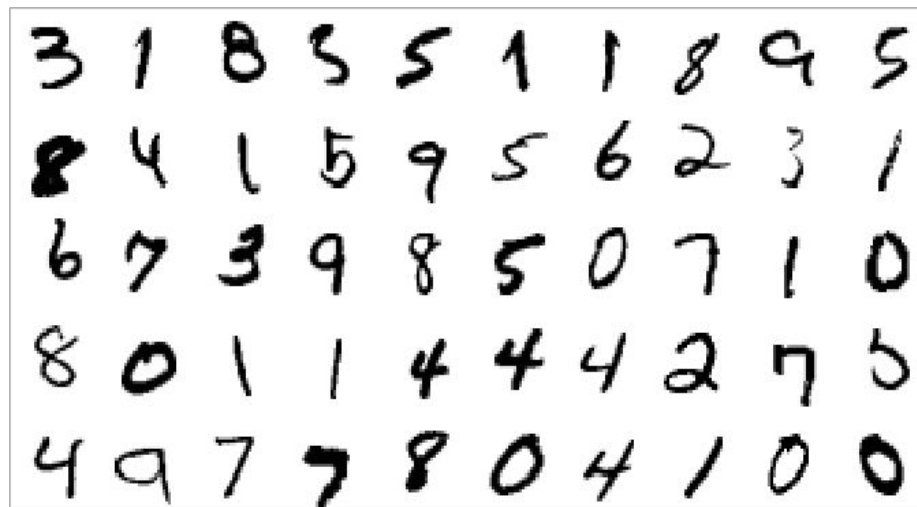
Dario Garcia Gasulla  
[dario.garcia@bsc.es](mailto:dario.garcia@bsc.es)

# Outline

1. **Fully connected** networks applied to **MNIST**
2. **CNNs** applied to **MNIST**
3. **CNNs** applied to **CIFAR10**

# MNIST

- ❖ MNIST is a *black and white* handwritten digit recognition dataset
- ❖ First testing ground for new AI techniques
- ❖ See how far you can get using a fully connected network



# CIFAR10

- ❖ CIFAR is a classification problem of low-resolution images (32x32)
- ❖ Version with 10 and 100 classes
- ❖ <https://www.cs.toronto.edu/~kriz/cifar.html>

**airplane**



**automobile**



**bird**



**cat**



**deer**



**dog**



**frog**



**horse**



**ship**



**truck**



# Let's look at the code

Get used to handling and loading data. It's a big part of any DL experiment.

Look into "*flow\_from\_directory*" from keras to avoid memory issues, when loading large datasets.

# Experiment 1 (FC & MNIST)

## ❖ Code

- [https://raw.githubusercontent.com/UPC-MAI-DL/UPC-MAI-DL.github.io/master/\\_codes/1.FNN-CNN/mnist\\_fnn\\_example.py](https://raw.githubusercontent.com/UPC-MAI-DL/UPC-MAI-DL.github.io/master/_codes/1.FNN-CNN/mnist_fnn_example.py)

## ❖ Launcher

- [https://raw.githubusercontent.com/UPC-MAI-DL/UPC-MAI-DL.github.io/master/\\_codes/1.FNN-CNN/launcher.sh](https://raw.githubusercontent.com/UPC-MAI-DL/UPC-MAI-DL.github.io/master/_codes/1.FNN-CNN/launcher.sh)

## ❖ Data

- <https://s3.amazonaws.com/img-datasets/mnist.npz>
- Within P9, store in ~/.keras/datasets

“wget” to download from internet to your pc  
“scp” to upload from your pc to P9

# Experiment 2 (CNN & MNIST)

## ❖ Code

- `https://raw.githubusercontent.com/UPC-MAI-DL/UPC-MAI-DL/master/\_codes/1.FNN-CNN/mnist\_cnn\_example.py`

## ❖ Launcher

- Adapt the launcher for experiment 1

# Experiment 3 (CNN & CIFAR10)

## ❖ Code

- Adapt the code from experiment 2
- Notice data dimensions

## ❖ Launcher

- Adapt the launcher for experiment 1

## ❖ Data

- <https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz>
- Rename to cifar-10-batches-py.tar.gz and store in ~/.keras/datasets



# Practical tips

- ❖ “tail -f file.out” to keep open for reading a live file
- ❖ model.summary()
  - track volumes
  - track complexity

Layer (type)	Output Shape	Param #
input_2 (InputLayer)	(None, 150, 150, 3)	0
conv2d_4 (Conv2D)	(None, 148, 148, 16)	448
max_pooling2d_4 (MaxPooling2)	(None, 74, 74, 16)	0
conv2d_5 (Conv2D)	(None, 72, 72, 32)	4640
max_pooling2d_5 (MaxPooling2)	(None, 36, 36, 32)	0
conv2d_6 (Conv2D)	(None, 34, 34, 64)	18496
max_pooling2d_6 (MaxPooling2)	(None, 17, 17, 64)	0
flatten_1 (Flatten)	(None, 18496)	0
dense_1 (Dense)	(None, 512)	9470464
dense_2 (Dense)	(None, 1)	513

**Dario Garcia-Gasulla (BSC)**  
*[dario.garcia@bsc.es](mailto:dario.garcia@bsc.es)*

