

# MAI

# Deep Learning

# The Cluster and I



**Dario Garcia-Gasulla**  
***dario.garcia@bsc.es***

# The Cluster and I

Cluster basics  
Software  
Jobs

- One cluster account per **registered** student
  - It's *illegal* to share it
- Each student is responsible of its own account
- Account enabled during the whole semester
  - All data will be lost afterwards (31/01/20)
- BSC clusters sometimes have down time
  - We will inform you
  - Deadlines will be adapted if needed



# The Cluster and I

## Cluster basics

### Software

### Jobs



- Two types of cluster:
  - Compute clusters (MinoTauro or mt1)
  - Data clusters (data transfer or dt01)
- Clusters have no outside access. No internet!
- Check the cluster user manual online
  - e.g., google "Minotauro user guide"
- Use dt for moving data from your computer:
  - (DW) `scp username@dt01.bsc.es:gpfs\_path local_path`
  - (UP) `scp local_path username@dt01.bsc.es:gpfs\_path`
- Use mt1 for experimentation

# The Cluster and I

Cluster basics  
Software  
Jobs



- Connect through SSH
  - `ssh username@mt1.bsc.es`
  - `ssh username@dt01.bsc.es`
- Work on your home directory
  - `/home/nct01/username`
- First thing: CHANGE your password, using the dt node
  - `ssh username@dt01.bsc.es`
  - `passwd`

# The Cluster and I

**Cluster basics**  
**Software**  
**Jobs**



- There are many DL frameworks available
  - Caffe2 (by Berkeley)
  - Microsoft Cognitive Toolkit (aka CNTK)
  - MXNet (by Apache)
  - PyTorch (by Facebook)
  - TensorFlow (by Google)
  - PaddlePaddle (by Baidu)
  - Keras (on top of TF, Theano, CNTK)
  - ...
- You can use whatever you want. We provide examples in Keras for simplicity (beware of the software versions!)

# The Cluster and I

## Cluster basics

## Software

## Jobs



- In the cluster there are several installed
  - PyTorch, TensorFlow, Keras ...
- Software is organized in modules. Load a module to access the software
- Module commands
  - `module list`: currently loaded modules
  - `module avail`: currently available modules
  - `module purge`: remove all loaded modules
  - `module load X`: load module X
- Order of loading is important due to dependencies

# The Cluster and I

Cluster basics  
Software  
Jobs



- In MinoTauro, module `python/3.6.3_ML` contains most of the machine learning software. Including:
  - TensorFlow
  - PyTorch
  - Keras
  - Scikit
  - Numpy

# The Cluster and I

Cluster basics  
Software  
Jobs



- Cluster jobs
  - To execute things in the cluster you launch "jobs"
  - Jobs are enqueued and executed in order, defined by a quota, considering:
    - Resources requested
    - Time requested
    - Resources used so far
- Tips:
  - Don't leave all experiments for the last few days. You may get queued a while if the cluster is busy.
  - Queues are less crowded during weekends



# The Cluster and I

Cluster basics  
Software  
Jobs



- To run a job, first define a launcher file with the needed directives:
- Available queues: (bsc\_queues command)
  - training: max 48h
  - debug: max 1h (but faster to get in)
  - Specify the queue with the command:  
*#SBATCH --partition=debug*
- Giving a maximum execution time (hard limit!)  
*#SBATCH --time=HH:MM:SS*
- Initial execution path  
*#SBATCH --workdir=pathname*

# The Cluster and I

Cluster basics

Software

Jobs



- Error and log file
  - `#SBATCH --error=file_name_%j.err`
  - `#SBATCH --output=file_name_%j.out`
- Define resources to use
  - Just one CPU  
`#SBATCH --ntasks=1`
  - Just one GPU (unless you parallelize)  
`#SBATCH --gres gpu:1`
  - If you ask for 1 GPU, you need to ask for the associated 4 CPUs
- Define software needed (modules to load)
  - This includes Python 3.6, Keras and TF  
`module purge; module load K80/default`  
`impi/2018.1 mkl/2018.1 cuda/8.0`  
`CUDNN/7.0.3 python/3.6.3_ML`

# The Cluster and I

**Cluster basics**  
**Software**  
**Jobs**



- Example of a complete launcher file

*run some\_code.py for 2 minutes using 1 GPU*

```
#!/bin/bash
#SBATCH --job-name="test_job"
#SBATCH --workdir=.
#SBATCH --output=mnist_%j.out
#SBATCH --error=mnist_%j.err
#SBATCH --ntasks=4
#SBATCH --gres gpu:1
#SBATCH --time=00:02:00
```

```
module purge; module load K80/default impi/2018.1
mk1/2018.1 cuda/8.0 CUDNN/7.0.3 python/3.6.3_ML
```

```
python some_code.py
```

# The Cluster and I

Cluster basics  
Software  
Jobs

- Launching jobs
  - `sbatch launcher_file.cmd`
- Checking the status of jobs and the jobId
  - `squeue`
- Killing jobs
  - `scancel jobId`
- Interactive jobs
  - `squeue` (find the Id of the computing node)
  - `ssh id_node` (from within login node to connect to it)



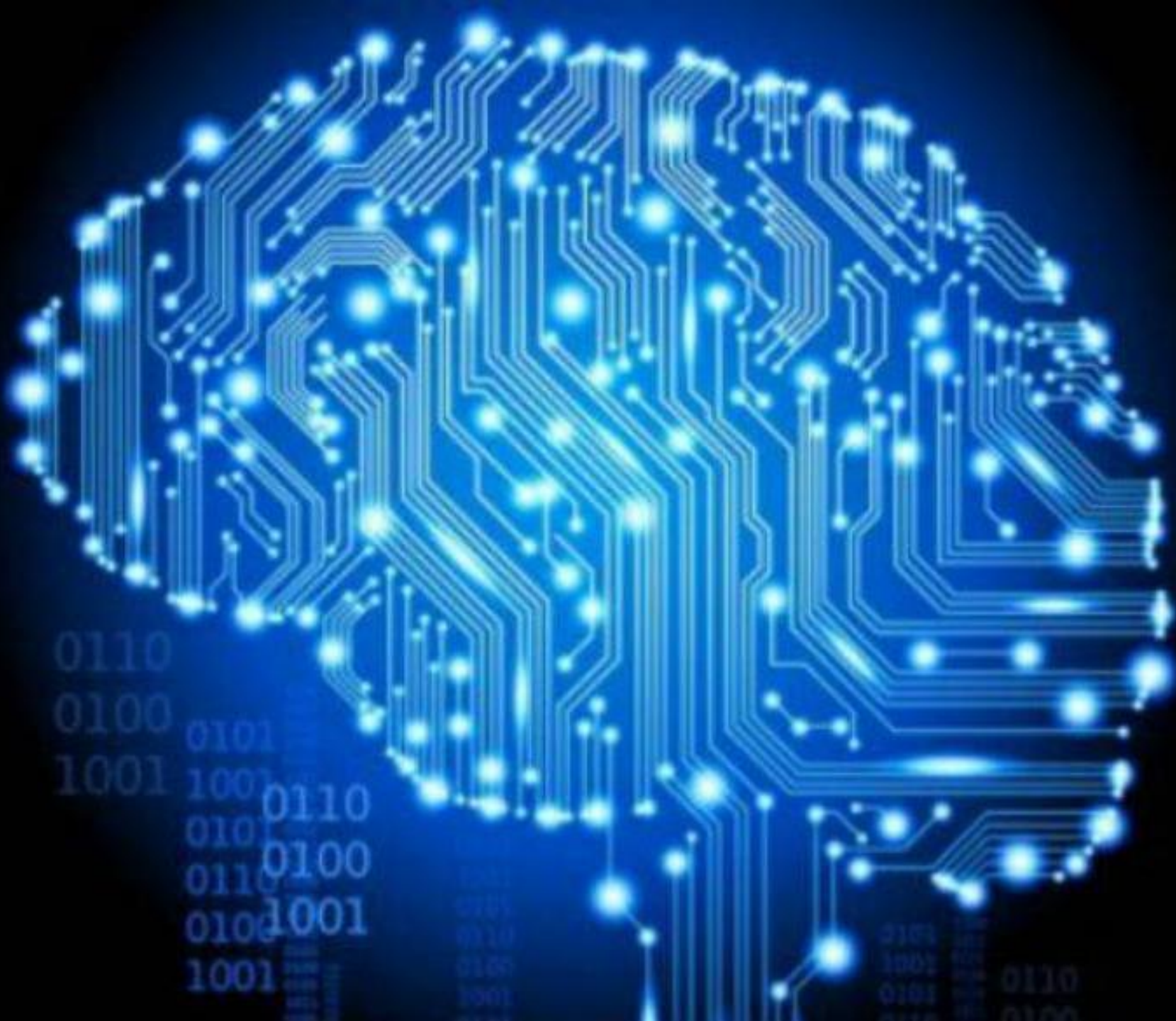
# The Cluster and I

**Cluster basics**  
**Software**  
**Jobs**



- Interactive sessions
  - Besides jobs, you can also execute things through interactive sessions
  - Limited to 1 hour
  - Allow you to connect to the computing node and to execute on it "live"
  - Appropriate for debugging and prototyping
  - It also consumes quota
  - Check out the online cluster user guide

# Basic working example



# Example: Basic steps

- Before the first lab session, make sure you can successfully run the following example



# Example: Basic steps

- Get the data from somewhere

```
wget https://s3.amazonaws.com/img-  
datasets/mnist.npz
```

- Upload it to the cluster

```
scp mnist.npz  
nctXXYYY@dt01.bsc.es:/home/nctXX/nctXXYYY/  
.keras/datasets/
```

You may need to create the directory first



# Example: Basic steps

- Write or upload the code and the launcher

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

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

- Submit job
  - sbatch launcher.sh

**Make sure this works before the first lab!**