

Leonardo - Scientific Python user environment and tools for AI: the CINECA Artificial Intelligence project

The bulk of the "cineca-ai" package, provided by the deeplrn profile, includes (for example) Tensorflow, Pytorch, XGBoost, and other related packages and dependencies.

This module has been personalized by CINECA AI experts and its installations have been performed through the "spack" tool. You can find different cineca-ai modules in profile/deeplrn. The "module help" reports the versions of the main components (pytorch, tensorflow etc.). For a complete list load the module and launch the "python -m pip list" command.

The CINECA AI project can be used in several ways, depending on the method more suited to your needs and on the availability of conda/pip packages.

1. Loading cineca-ai module

The way to use the installations of the cineca-ai environment goes through the loading of the module:

```
module load profile/deeplrn
module av cineca-ai
module load cineca-ai/<version>

# list all available python installations of cineca-ai
python -m pip list

# use a specific listed installation
python -c "import PACKAGE"
```

If you need to use a package not included in the list of those provided by the cineca-ai modules, you can always rely on the cineca-ai environment for the dependencies and install what you need within a personal virtual environment and/or a conda environment.

1.1 Install additional packages within a virtual environment

If you wish to install additional python packages to use together with the cineca-ai suite, you can create a personal virtual env and install what you need in the following way :

```
# create the virtual env loading cineca-ai module
module load profile/deeplrn
module av cineca-ai
module load cineca-ai/<version>
python -m venv <myvenv> --system-site-packages

# activate the created virtual env to install your python packages.
source <myvenv>/bin/activate
pip list
pip install PACKAGE
deactivate
```

NOTES:

- <myvenv>: choose an arbitrary, up-to-you name for your personal virtual env
- the *--system-site-packages* flag gives the virtual environment access to the system site-packages directory (otherwise you cannot access the cineca-ai environment)
- it is advised to create your personal envs in your \$WORK area, since the \$HOME disk quota is limited to 50 GB
- to test the installation launch: `python -c "import PACKAGE"`
- to use the installed PACKAGE, just *source* your env (`source <myvenv>/bin/activate`): you will access your packages AND those of the cineca-ai environment.
- if PACKAGE is not available to pip, see Section 2

Example: torch-scatter

It requires torch, which will be taken from the cineca-ai env (i.e., no need to install it):

```
$ module load profile/deeplrn
$ module load autoload cineca-ai/<version>
$ python -m venv torch_venv --system-site-packages
$ source torch_venv/bin/activate
$ pip install torch-scatter
```

1.2 Install additional packages within a conda environment

If you wish to install additional conda packages to use together with the cineca-ai suite, you can create a personal conda env and install what you need in the following way (we strongly suggest using a python *venv* if possible) :

```
# create the conda env loading cineca-ai module
module load anaconda3/<version>
module load profile/deeplrn
module av cineca-ai
module load cineca-ai/<version>
conda create -p <path>/my_env -c conda-forge --override-channels

# activate the created conda env to access cineca packages and install your conda packages.
conda activate <path>/my_env
python -m pip list
python -m pip install my_packages
conda deactivate
```

2. Loading spack module

An alternative way to use specific installations of cineca-ai environment or install additional packages starting Cineca-ai environment goes through the loading of the spack module that CINECA staff used to perform cineca-ai installations

```
# find a specific installation of cineca-ai module
module load spack
spack find <CINECA-AI PACKAGE>
e.g. spack find py-torch

# use the specific installation of cineca-ai module loading it via spack
spack loads <CINECA-AI PACKAGE>
e.g. spack loads py-torch

For checking the loaded packages and eventually unload them
spack find --loaded
spack unload --all

# use the specific installation of cineca-ai module to install additional packages via spack
spack install <YOUR PACKAGE> ^<CINECA-AI PACKAGE>
e.g. spack install <YOUR PACKAGE> ^py-torch
```