How to use Tensorflow on the GPU nodes

This page describes how to prepare for and run jobs which use Tensorflow on the GPU nodes.

First you need to clone the following repository:

```
$ git clone https://c4science.ch/source/tensorflow-pip-deneb.git
```

The currently available version is 1.9.0. New version will be added once they are available. The recommended setup uses Python virtual environments, as they enable Tensorflow to be installed in an isolated and stable environment.

### One time setup

The only supported Python at the moment is the system one, Python 3.6.5.

These commands have to be executed within a job in the `gpu` partition and QOS (alternatively it's also possible to load the GPU software environment in any other node).

```
$ ssh deneb2.epfl.ch
$ Sinteract -p gpu -q gpu -t 01:00:00 -m 4G
$ module purge
$ module load gcc cuda cudnn python
$ virtualenv --system-site-packages -p python3 tensorflow-1.9
$ source ~/tensorflow-1.9/bin/activate
$ pip3 install --upgrade pip setuptools
$ pip3 install --no-cache-dir --upgrade ~/tensorflow-pip-deneb/1.9-gpu/tensorflow-1.9.0-cp36-cp36m-linux_x86_64.whl
```

Additionally, if other Python packages are needed they should be installed inside the same Python virtual environment. Python packages installed in the system or the user home directory are not available within the Python virtual environment.

For example, if `pandas` is needed, use the following command (still within a job in the `gpu` partition and with at least the same modules loaded):

```
$ pip3 install --no-cache-dir --upgrade pandas
```

### Running a job in batch mode

Submit your job script

Example job script:

```
#!/bin/bash -l
#SBATCH --nodes=1
#SBATCH --time=1:0:0
#SBATCH --qos=gpu
#SBATCH --gres=gpu:1
#SBATCH --ntasks-per-node=1
#SBATCH --cpus-per-task=1
#SBATCH --partition=gpu

slmodules -s x86_E5v2_Mellanox_GPU
module load gcc cuda cudnn mvapich2 openblas
source ~/tensorflow-1.9/bin/activate
srun python your_input.py
```

### Running a job interactively

IPython is needed. It should be installed inside the same Python virtual environment as above using the following command:

```
pip3 install --no-cache-dir --upgrade ipython
```

Then

```
Sinteract -p gpu -q gpu_free -g gpu:1
```
module purge
module load gcc cuda cudnn python
source ~/tensorflow-1.9/bin/activate
ipython
copy/paste the lines from your_input.py

Related articles
- Python Virtual Environments
- How to use Tensorflow on the GPU nodes
- MPI4PY