

Getting started with Open Cognitive Environment (OpenCE, former WMLCE)

- [Open Cognitive Environment](#)
- [Simple Example with TensorFlow](#)
 - [Interactive mode](#)
 - [Batch mode](#)
 - [Visualization with TensorBoard](#)
 - [Interactive mode](#)
 - [Batch mode](#)
 - [Start the TensorBoard session](#)
- [Simple Example with Pytorch](#)
 - [Interactive mode](#)

Open Cognitive Environment

Welcome to the **OpenCE** project. The project contains everything that is needed to build conda packages for a collection of machine learning and deep learning frameworks. All packages created for a specific version of OpenCE have been designed to be installed within a single conda environment.

| Environment | opence-v1.3.1 | opence-v1.2.2 | opence-v1.1.2 | opence-v1.0.0 |
|--------------|---------------|-----------------|----------------|----------------|
| python | 3.8.0 | 3.8.0 | 3.8.12 | 3.8.12 |
| cuda | 11.2.2 | 11.0.221 | 10.2.89 | 10.2.89 |
| cudnn | 8.1.1 | 8.1.1 | 7.6.5 | 7.6.5 |
| nccl | 2.8.3 | 2.7.8 | 2.7.8 | 2.7.8 |
| openmpi | 4.1.1 | 3.1.3 | 3.1.3 | 3.1.3 |
| apex | 0.1 | N/A | N/A | N/A |
| hdf5 | 1.10.4 | 1.10.6 | 1.10.4 | 1.10.4 |
| horovod | 0.21.3 | 0.21.0 | 0.21.0 | 0.19.5 |
| ipython | 7.27.0 | 7.29.0 | 7.29.0 | 7.28.0 |
| matplotlib | 3.4.3 | 3.4.3 | 3.4.3 | 3.4.2 |
| mpi4py | 3.1.1 | N/A | N/A | N/A |
| numpy | 1.21.2 | 1.19.5 | 1.19.5 | 1.19.2 |
| onnx | 1.7.0 | 1.6.0 | 1.6.0 | 1.6.0 |
| opencv | 3.4.14 | 4.5.0 | 3.4.10 | 3.4.10 |
| pandas | 1.3.2 | 1.3.4 | 1.3.4 | 1.2.4 |
| pytorch | 1.8.1 | 1.7.1 | 1.7.1 | 1.6.0 |
| scikit-learn | 0.24.2 | 1.0.1 | 1.0.1 | 1.0.1 |
| scipy | 1.7.1 | 1.7.1 | 1.4.1 | 1.4.1 |
| tensorflow | 2.5.1 | 2.4.1 | 2.4.1 | 2.3.1 |
| tensorboard | 2.5.0 | 2.4.1 | 2.4.1 | 2.3.0 |
| transformers | 4.4.2 | 2.1.1 | 4.12.2 | 4.12.2 |

Simple Example with TensorFlow

Interactive mode

Get a node for interactive use:

```
swrun -p gpux1
```

Once on the compute node, load PowerAI module using one of these:

```
module load opence
module load opence-v1.3.1
```

Copy the following code into file "mnist-demo.py":

```
import tensorflow as tf
mnist = tf.keras.datasets.mnist

(x_train, y_train),(x_test, y_test) = mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0

model = tf.keras.models.Sequential([
    tf.keras.layers.Flatten(input_shape=(28, 28)),
    tf.keras.layers.Dense(512, activation=tf.nn.relu),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(10, activation=tf.nn.softmax)
])
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

model.fit(x_train, y_train, epochs=5)
model.evaluate(x_test, y_test)
```

Train on MNIST with keras API:

```
python ./mnist-demo.py
```

Batch mode

The same can be accomplished in batch mode using the following [tf_sample.swb](#) script:

```
wget https://wiki.ncsa.illinois.edu/download/attachments/82510352/tf_sample.swb
sbatch tf_sample.swb
squeue
```

Visualization with TensorBoard

Interactive mode

Get a node for interactive use:

```
swrun -p gpux1
```

Once on the compute node, load PowerAI module using one of these:

```
module load opence
module load opence-v1.3.1
```

Download the code [mnist-with-summaries.py](#) to \$HOME folder:

```
cd ~
wget https://wiki.ncsa.illinois.edu/download/attachments/82510352/mnist-with-summaries.py
```

Train on MNIST with TensorFlow summary:

```
python ./mnist-with-summaries.py
```

Batch mode

The same can be accomplished in batch mode using the following `tfd_sample.swb` script:

```
wget https://wiki.ncsa.illinois.edu/download/attachments/82510352/tfd_sample.swb
sbatch tfd_sample.swb
squeue
```

Start the TensorBoard session

After job completed the TensorFlow log files can be found in "`~/tensorflow/mnist/logs`", start the TensorBoard server on hal-ondemand, detail refers [Getting started with HAL OnDemand](#).

Simple Example with Pytorch

Interactive mode

Get a node for interactive use:

```
srun -p gpux1
```

Once on the compute node, load PowerAI module using one of these:

```
module load opence
module load opence-v1.3.1
```

Install samples for Pytorch:

```
pytorch-install-samples ~/pytorch-samples
cd ~/pytorch-samples
```

Train on MNIST with Pytorch:

```
python ./examples/mnist/main.py
```