Example Programs

Example programs are provided below for your use.

Environment Variable

Set this:

```
export HABANA_LOGS=~/.habana_logs
```

Copy Examples

Habana provides examples of some well known AI applications under the path. Clone the repository to your home directory.

```
cd ~/
git clone https://github.com/HabanaAI/Model-References.git
cd Model-References
git checkout 1.6.1
```

Set PYTHONPATH

Find the Python path using the command:

```
which python3
```

The output will look something like:

```
/usr/bin/python3

# Or inside a virtual environment
/home/CELSGCEUserID/my_env/bin/python3
```

Set PYTHONPATH

```
export PYTHONPATH=/path/to/python3
```

Then

```
export PYTHONPATH=/path/to/Model-References:$PYTHONPATH
```

If **Model-References** is in your root directory, the command would be

```
export PYTHONPATH=~/Model-References:$PYTHONPATH
```

Common Arguments

Below are some of the common arguments used across most of the models in the example code.

Argument	Default	Help
-b	1	Batch size for training
-batch-size		
--epochs	1	Number epochs for training
--world_size	1	--world_size
--distributed		Distribute training
--hpu		Gaudi training
--data_type		Specify data type to be either bf16 or fp32.
--distributed		whether to enable distributed mode
		and run on multiple devices

MNIST

Change directory

```
cd ~/Model-References/PyTorch/examples/computer_vision/hello_world
```

Demo_mnist Arguments

Argument	Default	Help
--lr	1.0	Learning rate for training
--gamma	0.7	Learning rate step gamma

Argument	Default	Help
--data-path	'../data'	input data path for train and test

Single Gaudi Run Commands

Single Gaudi FP32 eager mode run command:

```
python mnist.py --hpu
```

Single Gaudi BF16 eager mode run command:

```
python mnist.py --hpu --hmp
```

Single Gaudi FP32 lazy mode run command:

```
python mnist.py --hpu --use_lazy_mode
```

Single Gaudi BF16 lazy mode run command:

```
python mnist.py --hpu --hmp --use_lazy_mode
```

Multi-HPU run commands

NOTE: These run examples use PyTorch DDP, Distributed Data Parallel. This is accomplished by using model replica, splitting of data, and averaging the gradient through allreduce. The HCCL, Habana Collective Communication Library, is used to perform these operations.

There are eight Gaudi cards on the system.

On 8 HPU, 1 HLS and in FP32 eager mode, run the following command:

```
mpirun -n 8 --bind-to core --map-by slot:PE=7 --rank-by core --report-
bindings --allow-run-as-root $PYTHON mnist.py --batch-size=64 --epochs=1 --
lr=1.0 --gamma=0.7 --hpu
```

On 8 HPU, 1 HLS and in BF16 eager mode, run the following command:

```
mpirun -n 8 --bind-to core --map-by slot:PE=7 --rank-by core --report-
bindings --allow-run-as-root $PYTHON mnist.py --batch-size=64 --epochs=1 --
```

```
lr=1.0 --gamma=0.7 --hpu --hmp --hmp-bf16=ops_bf16_mnist.txt --hmp-
fp32=ops_fp32_mnist.txt
```

On 8 HPU, 1 HLS and in FP32 lazy mode, run the following command:

```
mpirun -n 8 --bind-to core --map-by slot:PE=7 --rank-by core --report-
bindings --allow-run-as-root $PYTHON mnist.py --batch-size=64 --epochs=1 --
lr=1.0 --gamma=0.7 --hpu --use_lazy_mode
```

On 8 HPU, 1 HLS and in BF16 lazy mode, run the following command:

```
mpirun -n 8 --bind-to core --map-by slot:PE=7 --rank-by core --report-
bindings --allow-run-as-root $PYTHON mnist.py --batch-size=64 --epochs=1 --
lr=1.0 --gamma=0.7 --hpu --hmp --hmp-bf16=ops_bf16_mnist.txt --hmp-
fp32=ops_fp32_mnist.txt --use_lazy_mode
```