

Benchmarking tool for the autograd API

This folder contain a set of self-contained scripts that allow to benchmark the autograd with different common models. It is designed to run the benchmark before and after your change and will generate a table to share on the PR.

To do so, you can use `functional_autograd_benchmark.py` to run the benchmarks before your change (using as output `before.txt`) and after your change (using as output `after.txt`). You can then use `compare.py` to get a markdown table comparing the two runs.

The default arguments of `functional_autograd_benchmark.py` should be used in general. You can change them though to force a given device or force running even the (very) slow settings.

Sample usage

```
# Make sure you compile pytorch in release mode and with the same flags before/after
export DEBUG=0
# When running on CPU, it might be required to limit the number of cores to avoid
oversubscription
export OMP_NUM_THREADS=10

# Compile pytorch with the base revision
git checkout master
python setup.py develop

# Run the benchmark for the base
# This will use the GPU if available.
pushd benchmarks/functional_autograd_benchmark
python functional_autograd_benchmark.py --output before.txt

# Compile pytorch with your change
popd
git checkout your_feature_branch
python setup.py develop

# Run the benchmark for the new version
pushd benchmarks/functional_autograd_benchmark
python functional_autograd_benchmark.py --output after.txt

# Get the markdown table that you can paste in your github PR
python compare.py

popd
```

Files in this folder:

- `functional_autograd_benchmark.py` is the main entry point to run the benchmark.
- `compare.py` is the entry point to run the comparison script that generates a markdown table.
- `torchaudio_models.py` and `torchvision_models.py` contains code extracted from torchaudio and torchvision to be able to run the models without having a specific version of these libraries installed.

- `ppl_models.py` , `vision_models.py` and `audio_text_models.py` contain all the getter functions used for the benchmark.