libtorch (C++-only)

The core of pytorch does not depend on Python. A CMake-based build system compiles the C++ source code into a shared object, libtorch.so.

Building libtorch using Python

You can use a python script/module located in tools package to build libtorch

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\(pytorch-master)(docs)libtorch.rst, line 13)

Unexpected indentation.

```
cd <pytorch_root>

# Make a new folder to build in to avoid polluting the source directories
mkdir build_libtorch && cd build_libtorch

# You might need to export some required environment variables here.
Normally setup.py sets good default env variables, but you'll have to do
that manually.
python ../tools/build_libtorch.py
```

Alternatively, you can call setup.py normally and then copy the built cpp libraries. This method may have side effects to your active Python installation.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\(pytorch-master)(docs)libtorch.rst, line 26)

Unexpected indentation.

```
cd <pytorch_root>
python setup.py build

ls torch/lib/tmp_install # output is produced here
ls torch/lib/tmp_install/lib/libtorch.so # of particular interest
```

To produce libtorch.a rather than libtorch.so, set the environment variable *BUILD_SHARED_LIBS=OFF*.

To use ninja rather than make, set *CMAKE_GENERATOR="-GNinja" CMAKE_INSTALL="ninja install"*.

Note that we are working on eliminating tools/build_pytorch_libs.sh in favor of a unified cmake build.

Building libtorch using CMake

You can build C++ libtorch.so directly with cmake. For example, to build a Release version from the master branch and install it in the directory specified by CMAKE_INSTALL_PREFIX below, you can use

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\(pytorch-master)(docs)libtorch.rst, line 43)

Unexpected indentation.

```
git clone -b master --recurse-submodule https://github.com/pytorch/pytorch.git
mkdir pytorch-build
cd pytorch-build
cmake -DBUILD_SHARED_LIBS:BOOL=ON -DCMAKE_BUILD_TYPE:STRING=Release -DPYTHON_EXECUTABLE:PATH=`which python3` -DCMAKE_INSTALL_PREFIX:PA
cmake --build . --target install
```

To use release branch v1.6.0, for example, replace `master` with `v1.6.0`. You will get errors if you do not have needed dependencies such as Python3's PyYAML package.