CppFlow

Run TensorFlow models in c++ without Bazel, without TensorFlow installation and without compiling Tensorflow. Perform tensor manipulation, use eager execution and run saved models directly from C++.

```cpp
// Read the graph
cppflow::model model("saved_model_folder");

// Load an image
auto input = cppflow::decode_jpeg(cppflow::read_file(std::string("image.jpg")));

// Cast it to float, normalize to range [0, 1], and add batch_dimension
input = cppflow::cast(input, TF_UINT8, TF_FLOAT);
input = input / 255.f;
input = cppflow::expand_dims(input, 0);

// Run
auto output = model(input);

// Show the predicted class
std::cout << cppflow::arg_max(output, 1) << std::endl;
```

You can take a look to the examples to see a full example on how to load a deep network and feed it with a sample image.

CppFlow uses Tensorflow C API to run the models, meaning you can use it without installing Tensorflow and without compiling the whole Tensorflow repository with bazel, you just need to download the C API. With this project you can manage and run your models in C++ without worrying about void, malloc or free. With CppFlow you easily can:

- Open saved models created with Python
- Execute Tensorflow neural networks in C++
- Perform tensor manipulation directly from C++

How To Run It

Since it uses TensorFlow 2 C API you just have to download it, check the docs to see a guide on how to do it.

Afterwards, you can install the library:

```
git clone git@github.com:serizba/cppflow.git
cd cppflow/examples/load_model
mkdir build
cd build
cmake ..
make -j
make install
```

Now you can check the quickstart guide to run a program using cppflow.

Documentation

Check the docs at https://serizba.github.io/cppflow/.

There you can find quickstart guides and more information about how to install the library and run the examples.

Development

CppFlow is basically a wrapper over Tensorflow C API. The basic class, tensor is a wrapper of a TF eager tensor, and it just constains a pointer to its TF representation.

The TF C API provides the tools to call all the TF raw ops, but using them is confusing. CppFlow includes a facade over these functions, so they can be called easily as normal C++ functions. To achieve this, the file ops contains (mostly) all the TF raw ops functions, but with a simple C++ interface. This file has been generated automatically using a small script.

CppFlow also includes a wrapper on TF saved models, the model class, so they can be easily opened and executed.

Contributors

If you are willing to contribute to this project, please go ahead an visit the development roadmap of cppflow. Specially `contributor_wanted` labelled PR or issues are very welcome to new contributors.

Citation

If you use this code or find this work useful in your research, please cite us:

```
@software{
    izquierdo2019cppflow,
    author = {Izquierdo, Sergio},
    doi = {10.5281/zenodo.7107618},
    title = {{cppflow: Run TensorFlow models in C++ without
       installation and without Bazel}},
    url = {https://github.com/serizba/cppflow},
    version = {2.0.0},
    month = {5},
    year = {2019}
}
```

Style guide

We use the Google's C++ style guide using static code linker cpplint. We use the Google's Python style guide using static code linker pylint using attached pylintrc configuration.

Remark

CppFlow is not related with TensorFlow. The CppFlow icon is a modified version of the TensorFlow logo. TensorFlow, the TensorFlow logo and any related marks are trademarks of Google Inc.