samjabrahams / **tensorflow-on-raspberry-pi**

TensorFlow for Raspberry Pi

#raspberry-pi #tensorflow #machine-learning

70 commits | **5** branches | **10** releases | **3** contributors

Branch: **master** ▾ | New pull request | Find file | Clone or download ▾

samjabrahams Release 1.1.0 — Latest commit `c962a5e` on Apr 30

benchmarks/inceptionv3	Clean GUIDE, README, remove C++ benchmark	9 months ago
testlogs	Add testlogs 2016_07_17 for Py2.7 and Py3	a year ago
.gitignore	Initial changes for 0.11.0 release	11 months ago
GUIDE.md	Release 1.1.0	6 months ago
ISSUE_TEMPLATE.md	Add ISSUE_TEMPLATE.md	a year ago
LICENSE	Add licensing information	2 years ago
README.md	Release 1.1.0	6 months ago
TENSORFLOW_LICENSE	Release 1.1.0	6 months ago

README.md

Installing TensorFlow on Raspberry Pi 3 (and probably 2 as well)

Donate

If you find the binaries and instructions in this repository useful, please consider donating to help keep this repository maintained. It takes hours of work to compile each new version of TensorFlow, in addition to time spent responding to issues and pull requests.

Intro

If you're looking to run fully featured TensorFlow or Bazel on a Raspberry Pi 3, you're in the right place. This repo contains step-by-step instructions for installing TensorFlow from source using Bazel (which is also compiled from-scratch), as well as pre-built TensorFlow binaries.

As a quick note, if you're looking for officially supported TensorFlow/Raspberry Pi functionality, you can also check out using the Makefile contrib module. It builds a static C++ library instead of the standard Python library, but is very powerful.

Contents

Installing from Pip

Note: These are unofficial binaries (though built from the minimally modified official source), and thus there is no expectation of support from the TensorFlow team. Please don't create issues for these files in the official TensorFlow repository.

This is the easiest way to get TensorFlow onto your Raspberry Pi 3. Note that currently, the pre-built binary is targeted for Raspberry Pi 3 running Raspbian 8.0 ("Jessie"), so this may or may not work for you. The specific OS release is the following:

```
Raspbian 8.0 "Jessie"
Release: March 2, 2017
Installed via NOOBS 2.3
```

First, install the dependencies for TensorFlow:

```
sudo apt-get update

# For Python 2.7
sudo apt-get install python-pip python-dev

# For Python 3.3+
sudo apt-get install python3-pip python3-dev
```

Next, download the wheel file from this repository and install it:

```
# For Python 2.7
wget https://github.com/samjabrahams/tensorflow-on-raspberry-pi/releases/download/v1.1.0/tensorflow-1.1.0-cp27-none-l
sudo pip install tensorflow-1.1.0-cp27-none-linux_armv7l.whl

# For Python 3.4
wget https://github.com/samjabrahams/tensorflow-on-raspberry-pi/releases/download/v1.1.0/tensorflow-1.1.0-cp34-cp34m-
sudo pip3 install tensorflow-1.1.0-cp34-cp34m-linux_armv7l.whl
```

Finally, we need to reinstall the `mock` library to keep it from throwing an error when we import TensorFlow:

```
# For Python 2.7
sudo pip uninstall mock
sudo pip install mock

# For Python 3.3+
sudo pip3 uninstall mock
sudo pip3 install mock
```

And that should be it!

Docker image

Instructions on setting up a Docker image to run on Raspberry Pi are being maintained by @romilly here, and a pre-built image is hosted on DockerHub here. Woot!

Troubleshooting

This section will attempt to maintain a list of remedies for problems that may occur while installing from `pip`

"tensorflow-1.1.0-cp27-none-linux_armv7l.whl is not a supported wheel on this platform."

This wheel was built with Python 2.7, and can't be installed with a version of `pip` that uses Python 3. If you get the above message, try running the following command instead:

```
sudo pip2 install tensorflow-1.1.0-cp27-none-linux_armv7l.whl
```

Vice-versa for trying to install the Python 3 wheel. If you get the error "tensorflow-1.1.0-cp34-cp34m-any.whl is not a supported wheel on this platform.", try this command:

```
sudo pip3 install tensorflow-1.1.0-cp34-cp34m-linux_armv7l.whl
```

Note: the provided binaries are for Python 2.7 and 3.4 *only*. If you've installed Python 3.5/3.6 from source on your machine, you'll need to either explicitly install these wheels for 3.4, or you'll need to build TensorFlow from source. Once there's an officially supported installation of Python 3.5+, this repo will start including wheels for those versions.

Building from Source

Step-by-step guide

If you aren't able to make the wheel file from the previous section work, you may need to build from source. Additionally, if you want to use features that have not been included in an official release, such as the initial distributed runtime, you'll have to build from source. Don't worry, as we've figured out most of the quirks of getting it right. The guide will be updated as needed to be as correct as possible.

See the step-by-step guide here. **Warning: it takes a while.**

Non-Raspberry Pi Model 3 builds

There are numerous single-board computers available on the market, but binaries and build instructions aren't necessarily compatible with what's available in this repository. This is a list of resources to help those with non-RPi3 (or RPi 2) computers get up and running:

- ODROID
    - Issue thread for ODROID
        - NeoTitans guide to building on ODROID C2

Credits

While the final pieces of grunt work were done primarily by myself and @petewarden, this effort has been going on for almost as long as TensorFlow has been open-source, and involves work that spans multiple months in separate codebases. This is an incomprehensive list of people and their work I ran across while working on this.

The majority of the source-building guide is a modified version of these instructions for compiling TensorFlow on a Jetson TK1. Massimiliano, you are the real MVP. *Note: the TK1 guide was updated on June 17, 2016*

@vmayoral put a huge amount of time and effort trying to put together the pieces to build TensorFlow, and was the first to get something close to a working binary.

A bunch of awesome Googlers working in both the TensorFlow and Bazel repositories helped make this possible. In no particular order: @vrv, @damienmg, @petewarden, @danbri, @ulfjack, @girving, and @nlothian

Issue threads of interest:

- Initial issue for building Bazel on ARMv7l
- First thread about running TensorFlow on RPi
- Parallel thread on building TensorFlow on ARMv7l
    - This is where the most recent conversation is located

License

Subdirectories contained within the `third_party` directory each contain relevant licenses for the code and software within those subdirectories.

The file TENSORFLOW_LICENSE applies to the binaries distributed in the releases.

The file LICENSE applies to other files in this repository. I want to stress that a majority of the lines of code found in the guide of this repository was created by others. If any of those original authors want more prominent attribution, please contact me and we can figure out how to make it acceptable.

Back to top