Installing TensorFlow on macOS

Determine how to install TensorFlow

You must pick the mechanism by which you install TensorFlow. The supported choices are as follows:

- virtualenv
- "native" pip
- Docker
- installing from sources, which is documented in a separate guide.

We recommend the virtualenv installation. Virtualenv is a virtual Python environment isolated from other Python development, incapable of interfering with or being affected by other Python programs on the same machine. During the virtualenv installation process, you will install not only TensorFlow but also all the packages that TensorFlow requires. (This is actually pretty easy.) To start working with TensorFlow, you simply need to "activate" the virtual environment. All in all, virtualenv provides a safe and reliable mechanism for installing and running TensorFlow.

Native pip installs TensorFlow directly on your system without going through any container or virtual environment system. Since a native pip installation is not walled-off, the pip installation might interfere with or be influenced by other Python-based installations on your system. Furthermore, you might need to disable System Integrity Protection (SIP) in order to install through native pip. However, if you understand SIP, pip, and your Python environment, a native pip installation is relatively easy to perform.

Docker completely isolates the TensorFlow installation from pre-existing packages on your machine. The Docker container contains TensorFlow and all its dependencies. Note that the Docker image can be quite large (hundreds of MBs). You might choose the Docker installation if you are incorporating TensorFlow into a larger application architecture that already uses Docker.

In Anaconda, you may use conda to create a virtual environment. However, within Anaconda, we recommend installing TensorFlow with the `pip install` command, not with the `conda install` command.

NOTE: The conda package is community supported, not officially supported. That is, the TensorFlow team neither tests nor maintains the conda package. Use that package at your own risk.

Installing with virtualenv

Take the following steps to install TensorFlow with Virtualenv:

1. Start a terminal (a shell). You'll perform all subsequent steps in this shell.
2. Install pip and virtualenv by issuing the following commands:

```
$ sudo easy_install pip
$ pip install --upgrade virtualenv
```

3. Create a virtualenv environment by issuing a command of one of the following formats:

```
$ virtualenv --system-site-packages targetDirectory # for Python 2.7
$ virtualenv --system-site-packages -p python3 targetDirectory # for Python 3.n
```

   where *targetDirectory* identifies the top of the virtualenv tree. Our instructions assume that *targetDirectory* is `~/tensorflow`, but you may choose any directory.

4. Activate the virtualenv environment by issuing one of the following commands:

```
$ source ~/tensorflow/bin/activate      # If using bash, sh, ksh, or zsh
$ source ~/tensorflow/bin/activate.csh  # If using csh or tcsh
```

The preceding `source` command should change your prompt to the following:

```
(tensorflow)$
```

5. Ensure pip ≥8.1 is installed:

```
(tensorflow)$ easy_install -U pip
```

6. Issue one of the following commands to install TensorFlow and all the packages that TensorFlow requires into the active Virtualenv environment:

```
(tensorflow)$ pip install --upgrade tensorflow      # for Python 2.7
(tensorflow)$ pip3 install --upgrade tensorflow     # for Python 3.n
```

7. Optional. If Step 6 failed (typically because you invoked a pip version lower than 8.1), install TensorFlow in the active virtualenv environment by issuing a command of the following format:

```
$ pip install --upgrade tfBinaryURL   # Python 2.7
$ pip3 install --upgrade tfBinaryURL  # Python 3.n
```

where *tfBinaryURL* identifies the URL of the TensorFlow Python package. The appropriate value of *tfBinaryURL* depends on the operating system and Python version. Find the appropriate value for *tfBinaryURL* for your system here. For example, if you are installing TensorFlow for macOS, Python 2.7, the command to install TensorFlow in the active Virtualenv is as follows:

```
$ pip3 install --upgrade \
https://storage.googleapis.com/tensorflow/mac/cpu/tensorflow-1.4.1-py2-none-any.whl
```

If you encounter installation problems, seeCommon Installation Problems.

Next Steps

After installing TensorFlow,validate your installationto confirm that the installation worked properly.

Note that you must activate the virtualenv environment each time youuse TensorFlow in a new shell. If the virtualenv environment is notcurrently active (that is, the prompt is not `(tensorflow)`, invokeone of the following commands:

```
$ source ~/tensorflow/bin/activate      # bash, sh, ksh, or zsh
$ source ~/tensorflow/bin/activate.csh  # csh or tcsh
```

Your prompt will transform to the following to indicate that yourtensorflow environment is active:

```
(tensorflow)$
```

When the virtualenv environment is active, you may runTensorFlow programs from this shell.

When you are done using TensorFlow, you may deactivate theenvironment by issuing the following command:

```
(tensorflow)$ deactivate
```

The prompt will revert back to your default prompt (as defined by `PS1`).

Uninstalling TensorFlow

If you want to uninstall TensorFlow, simply remove the tree you created. For example:

```
$ rm -r ~/tensorflow
```

Installing with native pip

We have uploaded the TensorFlow binaries to PyPI.Therefore, you can install TensorFlow through pip.

TheREQUIRED_PACKAGES section of setup.pylists the packages that pip will install or upgrade.

Prerequisite: Python

In order to install TensorFlow, your system must contain one of the following Python versions:

- Python 2.7
- Python 3.3+

If your system does not already have one of the preceding Python versions,install it now.

When installing Python, you might need to disableSystem Integrity Protection (SIP) to permit any entity other thanMac App Store to install software.

Prerequisite: pip

Pip installsand manages software packages written in Python. If you intend to installwith native pip, then one of the following flavors of pip must beinstalled on your system:

- `pip`, for Python 2.7
- `pip3`, for Python 3.n.

`pip` or `pip3` was probably installed on your system when youinstalled Python. To determine whether pip or pip3 is actuallyinstalled on your system, issue one of the following commands:

```
$ pip -V  # for Python 2.7
$ pip3 -V # for Python 3.n
```

We strongly recommend pip or pip3 version 8.1 or higher in orderto install TensorFlow. If pip or pip3 8.1 or later is notinstalled, issue the following commands to install or upgrade:

```
$ sudo easy_install --upgrade pip
$ sudo easy_install --upgrade six
```

Install TensorFlow

Assuming the prerequisite software is installed on your Mac,take the following steps:

1. Install TensorFlow by invoking **one** of the following commands:

```
 $ pip install tensorflow      # Python 2.7; CPU support
 $ pip3 install tensorflow     # Python 3.n; CPU support
If the preceding command runs to completion, you should now
 validate your installation.
```

2. (Optional.) If Step 1 failed, install the latest version of TensorFlow by issuing a command of the following format:

```
 $ sudo pip  install --upgrade tfBinaryURL   # Python 2.7
 $ sudo pip3 install --upgrade tfBinaryURL   # Python 3.n
```

where *tfBinaryURL* identifies the URL of the TensorFlow Python package. The appropriate value of *tfBinaryURL* depends on the operating system and Python version. Find the appropriate value for *tfBinaryURL* here. For example, if you are installing TensorFlow for Mac OS and Python 2.7 issue the following command:

```
 $ sudo pip3 install --upgrade \
 https://storage.googleapis.com/tensorflow/mac/cpu/tensorflow-1.4.1-py2-none-any.whl
```

If the preceding command fails, see installation problems.

Next Steps

After installing TensorFlow,validate your installationto confirm that the installation worked properly.

Uninstalling TensorFlow

To uninstall TensorFlow, issue one of following commands:

```
$ pip uninstall tensorflow
$ pip3 uninstall tensorflow
```

Installing with Docker

Follow these steps to install TensorFlow through Docker.

1. Install Docker on your machine as described in the Docker documentation.
2. Launch a Docker container that contains one of the TensorFlow binary images.

The remainder of this section explains how to launch a Docker container.

To launch a Docker container that holds the TensorFlow binary image,enter a command of the following format:

```
 $ docker run -it -p hostPort:containerPort TensorFlowImage
```

where:

- *-p hostPort:containerPort* is optional. If you'd like to run TensorFlow programs from the shell, omit this option. If you'd like to run TensorFlow programs from Jupyter notebook, set both *hostPort* and *containerPort* to `8888`. If you'd like to run TensorBoard inside the container, add a second `-p` flag, setting both *hostPort* and *containerPort* to 6006.
- TensorFlowImage

  is required. It identifies the Docker container. You must specify one of the following values:
- `gcr.io/tensorflow/tensorflow`: TensorFlow binary image.
- `gcr.io/tensorflow/tensorflow:latest-devel`: TensorFlow Binary image plus source code.

`gcr.io` is the Google Container Registry. Note that someTensorFlow images are also available atdockerhub.

For example, the following command launches a TensorFlow CPU binary imagein a Docker container from which you can run TensorFlow programs in a shell:

```
$ docker run -it gcr.io/tensorflow/tensorflow bash
```

The following command also launches a TensorFlow CPU binary image in aDocker container. However, in this Docker container, you can runTensorFlow programs in a Jupyter notebook:

```
$ docker run -it -p 8888:8888 gcr.io/tensorflow/tensorflow
```

Docker will download the TensorFlow binary image the first time you launch it.

Next Steps

You should nowvalidate your installation.

Installing with Anaconda

The Anaconda installation is community supported, not officially supported.

Take the following steps to install TensorFlow in an Anaconda environment:

1. Follow the instructions on the Anaconda download site to download and install Anaconda.
2. Create a conda environment named `tensorflow` by invoking the following command:

```
$ conda create -n tensorflow python=2.7 # or python=3.3, etc.
```

3. Activate the conda environment by issuing the following command:

```
$ source activate tensorflow
 (tensorflow)$  # Your prompt should change
```

4. Issue a command of the following format to install TensorFlow inside your conda environment:

```
(tensorflow)$ pip install --ignore-installed --upgrade TF_PYTHON_URL
```

   where *TF_PYTHON_URL* is the URL of the TensorFlow Python package. For example, the following command installs the CPU-only version of TensorFlow for Python 2.7:

```
(tensorflow)$ pip install --ignore-installed --upgrade \
https://storage.googleapis.com/tensorflow/mac/cpu/tensorflow-1.4.1-py2-none-any.whl
```

Validate your installation

To validate your TensorFlow installation, do the following:

1. Ensure that your environment is prepared to run TensorFlow programs.
2. Run a short TensorFlow program.

Prepare your environment

If you installed on native pip, virtualenv, or Anaconda, thendo the following:

1. Start a terminal.
2. If you installed with virtualenv or Anaconda, activate your container.
3. If you installed TensorFlow source code, navigate to any directory *except* one containing TensorFlow source code.

If you installed through Docker, start a Docker container that runs bash.For example:

```
$ docker run -it gcr.io/tensorflow/tensorflow bash
```

Run a short TensorFlow program

Invoke python from your shell as follows:

```
$ python
```

Enter the following short program inside the python interactive shell:

```
# Python
import tensorflow as tf
hello = tf.constant('Hello, TensorFlow!')
sess = tf.Session()
print(sess.run(hello))
```

If the system outputs the following, then you are ready to beginwriting TensorFlow programs:

```
Hello, TensorFlow!
```

If you are new to TensorFlow, seeGetting Started with TensorFlow.

If the system outputs an error message instead of a greeting, seeCommon installation problems.

Common installation problems

We are relying on Stack Overflow to document TensorFlow installation problemsand their remedies. The following table contains links to Stack Overflowanswers for some common installation problems.If you encounter an error message or otherinstallation problem not listed in the following table, search for iton Stack Overflow. If Stack Overflow doesn't show the error message,ask a new question about it on Stack Overflow and specifythe `tensorflow` tag.

Stack Overflow Link	Error Message
42006320	`ImportError: Traceback (most recent call last):File ".../tensorflow/core/framework/graph_pb2.py", line 6, in from google.protobuf import descriptor as _descriptorImportError: cannot import name 'descriptor'`
33623453	`IOError: [Errno 2] No such file or directory: '/tmp/pip-o6Tpui-build/setup.py'`
35190574	`SSLError: [SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed`
42009190	`Installing collected packages: setuptools, protobuf, wheel, numpy, tensorflow Found existing installation: setuptools 1.1.6 Uninstalling setuptools-1.1.6: Exception: ... [Errno 1] Operation not permitted: '/tmp/pip-a1DXRT-uninstall/.../lib/python/_markerlib'`
33622019	`ImportError: No module named copyreg`
37810228	During a `pip install` operation, the system returns: `OSError: [Errno 1] Operation not permitted`
33622842	An `import tensorflow` statement triggers an error such as the following:`Traceback (most recent call last): File "", line 1, in File "/usr/local/lib/python2.7/site-packages/tensorflow/__init__.py", line 4, in from tensorflow.python import * ... File "/usr/local/lib/python2.7/site-packages/tensorflow/core/framework/tensor_shape_pb2.py", line 22, in serialized_pb=_b('\n,tensorflow/core/framework/tensor_shape.proto\x12\ntensorflow\"d\n\x10TensorShapeProto\x12-\n\x03\x64im\x18\x02 \x03(\x0b\x32 .tensorflow.TensorShapeProto.Dim\x1a!\n\x03\x44im\x12\x0c\n\x04size\x18\x01 \x01(\x03\x12\x0c\n\x04name\x18\x02 \x01(\tb\x06proto3') TypeError: __init__() got an unexpected keyword argument 'syntax'`
42075397	A `pip install` command triggers the following error:`...You have not agreed to the Xcode license agreements, please run'xcodebuild -license' (for user-level acceptance) or'sudo xcodebuild -license' (for system-wide acceptance) from within aTerminal window to review and agree to the Xcode license agreements.... File "numpy/core/setup.py", line 653, in get_mathlib_info raise RuntimeError("Broken toolchain: cannot link a simple C program")RuntimeError: Broken toolchain: cannot link a simple C program`

The URL of the TensorFlow Python package

A few installation mechanisms require the URL of the TensorFlow Python package.The value you specify depends on three factors:

- operating system
- Python version

This section documents the relevant values for Mac OS installations.

Python 2.7

```
https://storage.googleapis.com/tensorflow/mac/cpu/tensorflow-1.4.1-py2-none-any.whl
```

Python 3.4, 3.5, or 3.6

```
https://storage.googleapis.com/tensorflow/mac/cpu/tensorflow-1.4.1-py3-none-any.whl
```

Protobuf pip package 3.1

You can skip this section unless you are seeing problems relatedto the protobuf pip package.

NOTE: If your TensorFlow programs are running slowly, you mighthave a problem related to the protobuf pip package.

The TensorFlow pip package depends on protobuf pip package version 3.1. Theprotobuf pip package downloaded from PyPI (when invoking`pip install protobuf`) is a Python-only library containingPython implementations of proto serialization/deserialization that can run**10x-50x slower** than the C++ implementation. Protobuf also supports abinary extension for the Python package that contains fastC++ based proto parsing. This extension is not available in thestandard Python-only pip package. We have created a custom binarypip package for protobuf that contains the binary extension. To installthe custom binary protobuf pip package, invoke one of the following commands:

- for Python 2.7:

```
$ pip install --upgrade \
https://storage.googleapis.com/tensorflow/mac/cpu/protobuf-3.1.0-cp27-none-macosx_10_11_x86_64.whl
```

- for Python 3.n:

```
$ pip3 install --upgrade \
https://storage.googleapis.com/tensorflow/mac/cpu/protobuf-3.1.0-cp35-none-macosx_10_11_x86_64.whl
```

Installing this protobuf package will overwrite the existing protobuf package.Note that the binary pip package already has support for protobufslarger than 64MB, which should fix errors such as these:

```
[libprotobuf ERROR google/protobuf/src/google/protobuf/io/coded_stream.cc:207]
A protocol message was rejected because it was too big (more than 67108864 bytes).
To increase the limit (or to disable these warnings), see
CodedInputStream::SetTotalBytesLimit() in google/protobuf/io/coded_stream.h.
```