

# Table of Contents

<b>TensorFlow</b> .....	<b>1</b>
TensorFlow for developing on Mac.....	1
TensorFlow mimicking the Grid env.....	1
Docker.....	1
Port forwarding.....	2
TensorFlow.....	2
Tensorboard.....	3

# TensorFlow

This is a quick and dirty explanation to get TensorFlow environment working within a Mac running 10.11 (el Capitan).

When submitting a job to the grid (which is the future goal), it will initialise a virtual Docker container which the code will run in. To develop in, this isn't ideal.

The first part shows you how to install TensorFlow locally without Docker so that developing code is more easy (don't have to copy to and from Docker containers, can host code in Git etc.).

The second part demonstrates installing TensorFlow in a Docker container on your mac, with the intention of mimicking what the grid environment will contain.

## TensorFlow for developing on Mac

Download pip and virtualenv:

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

Create the virtualenv so you don't mess with other python packages and versions on you mac:

- `virtualenv --system-site-packages ~/tensorflow`

Activate your env:

- `source ~/tensorflow/bin/activate`

and install TensorFlow:

- `pip install --upgrade`  
`https://storage.googleapis.com/tensorflow/mac/tensorflow-0.7.1-cp27-none-any.whl`

To get out of your virtualenv:

- `deactivate`

TensorFlow packages and examples are housed in `~/tensorflow/lib/python2.7/site-packages/tensorflow/models/image/cifar10`. Whenever you want to use TensorFlow you have to source your env: `source ~/tensorflow/bin/activate`.

## TensorFlow mimicking the Grid env

### Docker

So, to install docker:

Docker has a good instruction page here: <https://docs.docker.com/engine/installation/mac/> But essentially:

- Go here: <https://www.docker.com/products/docker-toolbox> and click Download, then follow instructions.

To start a docker environment, look for "docker Quickstart Terminal" in spotlight. To check that everything is hunky-dory, in this terminal that popped up, type: `docker run hello-world`. The output should look like:

```
Hello from Docker.  
This message shows that your installation appears to be working correctly.
```

To generate this message, Docker took the following steps:

1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal.

To try something more ambitious, you can run an Ubuntu container with:

```
$ docker run -it ubuntu bash
```

Share images, automate workflows, and more with a free Docker Hub account:

```
https://hub.docker.com
```

For more examples and ideas, visit:

```
https://docs.docker.com/userguide/
```

When starting a new Docker image, one can provide interactive flags (-i). To detach from this shell without it exiting, a Ctrl-d will do the trick (`docker info`) will tell you how many containers are running. To list all containers (stopped and running) do `docker ps -a`.

Some more interactive commands can be found here:

<https://docs.docker.com/engine/quickstart/#running-an-interactive-shell>

## Port forwarding

To port forward within Docker an example could be: `docker run -p 8888:8888 -p 6006:6006 imageToLoad`. Then, open up VirtualBox, click the "default" image that's running, click "settings -> Network -> Port Forwarding". Host IP is 127.0.0.1, Host Port can be 8810, Guest Port is 8888. Now going to <https://localhost:8010> from your web browser should work.

So Docker is now installed, now for TensorFlow

## TensorFlow

Installation of TensorFlow in the Docker environment is incredibly easy. Type `docker run -it b.gcr.io/tensorflow/tensorflow:latest-devel` from the "docker Quickstart Terminal". On first run, this will download the TensorFlow image, subsequent runs use the same, fresh image. More info and a list of images can be found at [https://www.tensorflow.org/versions/r0.7/get\\_started/os\\_setup.html](https://www.tensorflow.org/versions/r0.7/get_started/os_setup.html).

To get actual tag names that you can download do:

- `wget https://gcr.io/v2/tensorflow/tensorflow/tags/list`

To test if TensorFlow properly installed do `python -m tensorflow.models.image.mnist.convolutional`. Output should look something like:

```
Successfully downloaded train-images-idx3-ubyte.gz 9912422 bytes.  
Successfully downloaded train-labels-idx1-ubyte.gz 28881 bytes.  
Successfully downloaded t10k-images-idx3-ubyte.gz 1648877 bytes.  
Successfully downloaded t10k-labels-idx1-ubyte.gz 4542 bytes.  
Extracting data/train-images-idx3-ubyte.gz  
Extracting data/train-labels-idx1-ubyte.gz
```

```
Extracting data/t10k-images-idx3-ubyte.gz
Extracting data/t10k-labels-idx1-ubyte.gz
Initialized!
Step 0 (epoch 0.00), 7.4 ms
Minibatch loss: 12.053, learning rate: 0.010000
Minibatch error: 90.6%
Validation error: 84.6%
Step 100 (epoch 0.12), 601.3 ms
...
```

A nice image recognition example for TensorFlow can be found here:

[https://www.tensorflow.org/versions/r0.7/tutorials/image\\_recognition/index.html](https://www.tensorflow.org/versions/r0.7/tutorials/image_recognition/index.html)

More examples/tutorials can be found here: <https://www.tensorflow.org/versions/r0.7/tutorials/index.html>

## Tensorboard

To start Tensorboard, use the full Tensorflow image: i.e. `docker run -it -p 8888:8888 -p 6006:6006 b.gcr.io/tensorflow/tensorflow-full`. Then do

- `tensorboard --logdir ./ --port 8888`

and you can now open a web browser to <http://localhost:8810/>.

-- JoshuaWyattSmith - 2016-04-01

---

This topic: Main > TensorFlow

Topic revision: r5 - 2016-04-16 - JoshuaWyattSmith



Copyright &© 2008-2022 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

or Ideas, requests, problems regarding TWiki? use [Discourse](#) or [Send feedback](#)