TensorFlow ("TF")

What is TensorFlow (TF)?

TensorFlow is an open source offering from Google Brain Team.

It is a system that processes a dataFLOW graph, where the data that gets passed in and out of each node is a TENSOR (multi-dim array). In other words, it is a dataflow processor where ALL data is in the form of 'tensors'.

Why use TF?

Because CNNs involve pipelined neuron processing, where each neuron (a node in TF) processes arrays of inputs and weights (tensors in TF).

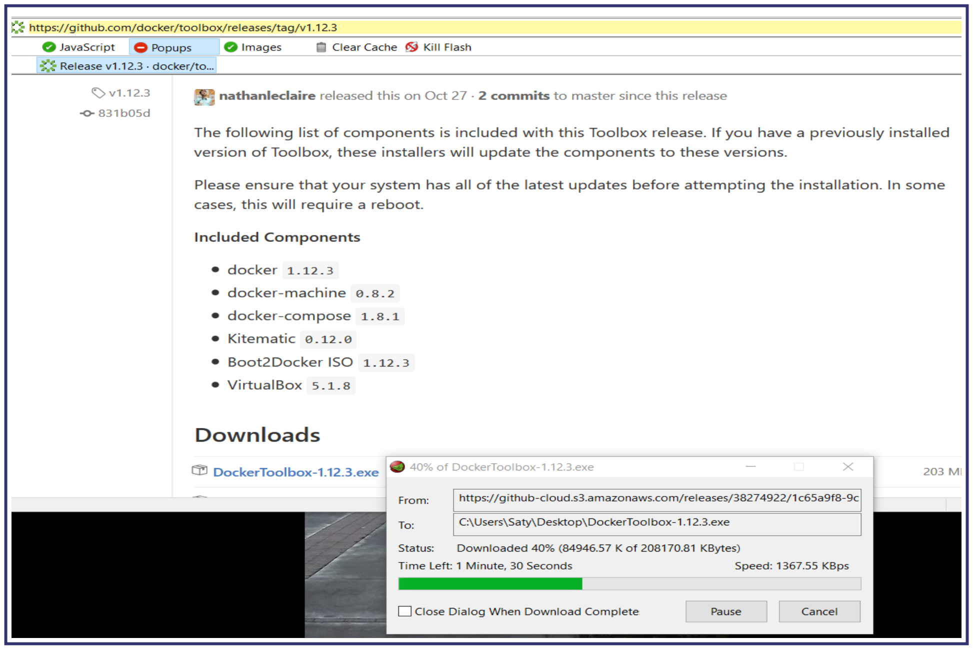
TF makes it possible to express neural networks as graphs (flexible), as opposed to a collection of chained function calls (rigid). The flexibility also allows the nodes to be processed in a variety of ways - in a CPU, GPU, cloud, mobile device..

Setting up TF

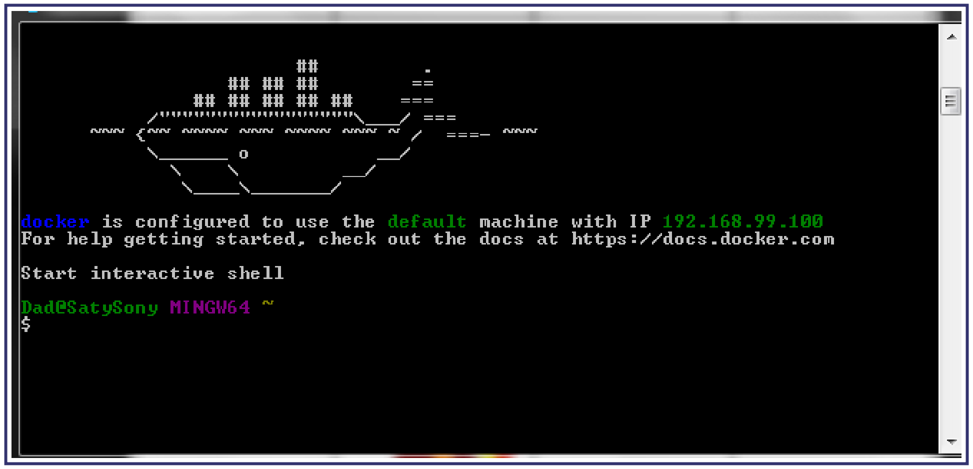
You can download TF from GitHub via your native Python environment (eg. using pip install), and use it in that environment.

Or you can use Docker! Very simply, Docker is a lightweight alternative to running a VM such as VirtualBox or VMWare. Docker allows for easy downloading and installing of software inside it, using git-style pull requests. For our purposes, we'll need to install Python first, then pull in tensorflow.

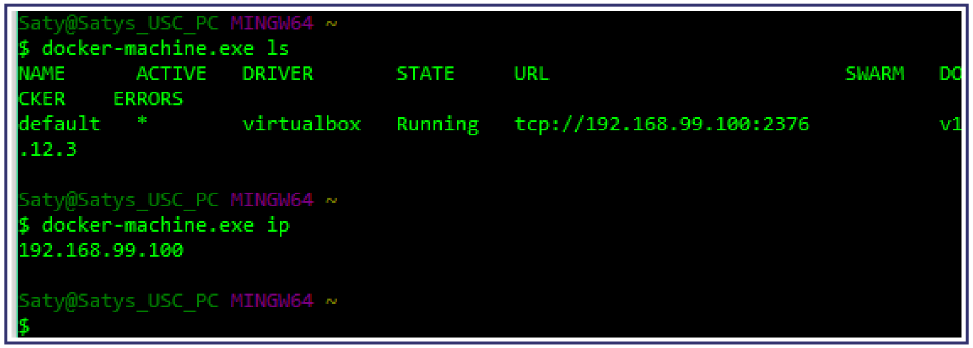
Install Docker Toolbox first:



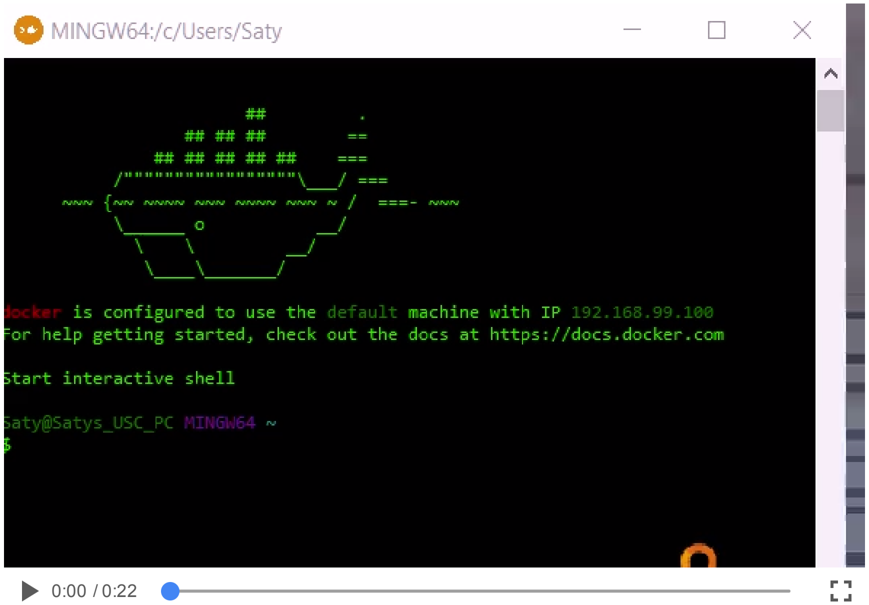
Next, launch a Docker shell (Docker Quickstart Terminal):



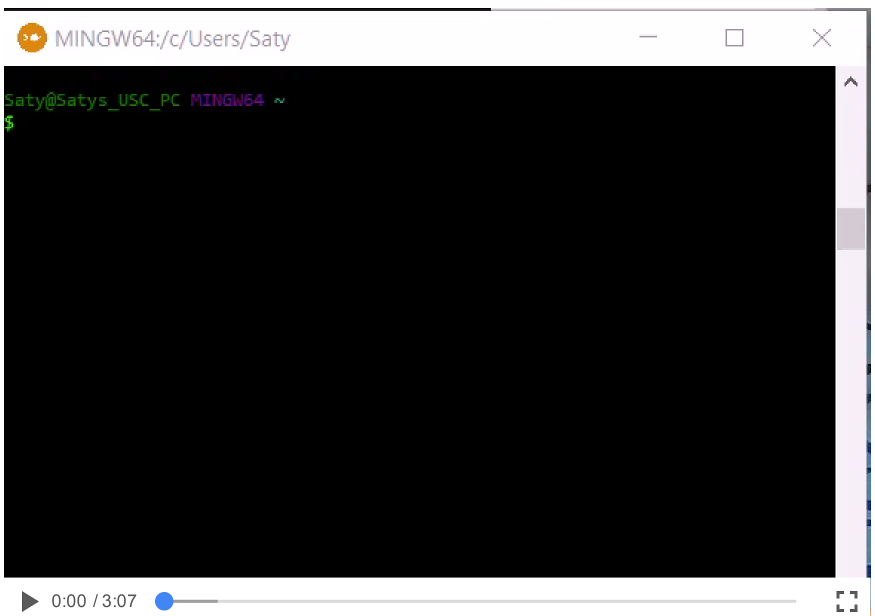
Next, verify that Docker is running properly [note - this screenshot is off a different PC compared to above!]:

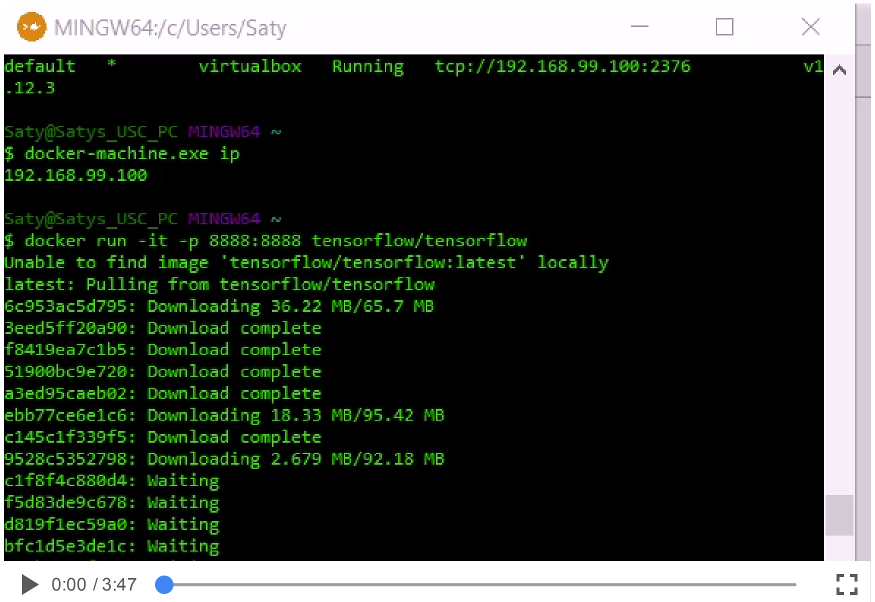


Now we can run Hello World :)

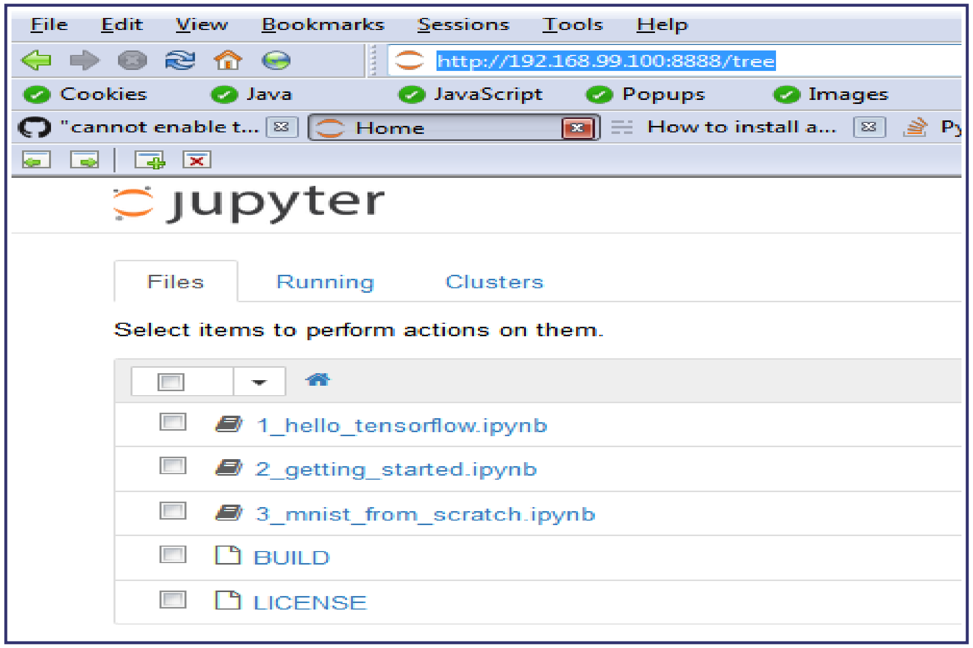


Time to install Python, followed by tensorflow:





Excellent! Now we can start playing with TensorFlow by visiting localhost:8888 (or http://192.168.99.100:8888) on our browser, and running "literate computing-style" Jupyter notebooks there.



You can even run Jupyter notebooks on the cloud (note - you can't edit using this interface). Eg. here is a pre-loaded example (GitHub can run this, too).

A quick example

Here is an example where we add two "tensors" (arrays of identical length/size) to obtain a resulting "tensor".

Another short example

Here we chain additions:

In [8]:
```
import tensorflow as tf

with tf.Session():
    input1 = tf.constant(1.0, shape=[4])
    input2 = tf.constant(2.0, shape=[4])
    input3 = tf.constant(3.0, shape=[4])
    output = tf.add(tf.add(input1, input2), input3)
    result = output.eval()
    print(result)
```

```
[ 6   6   6   6 ]
```

0:00 / 0:13

numpy ("np")

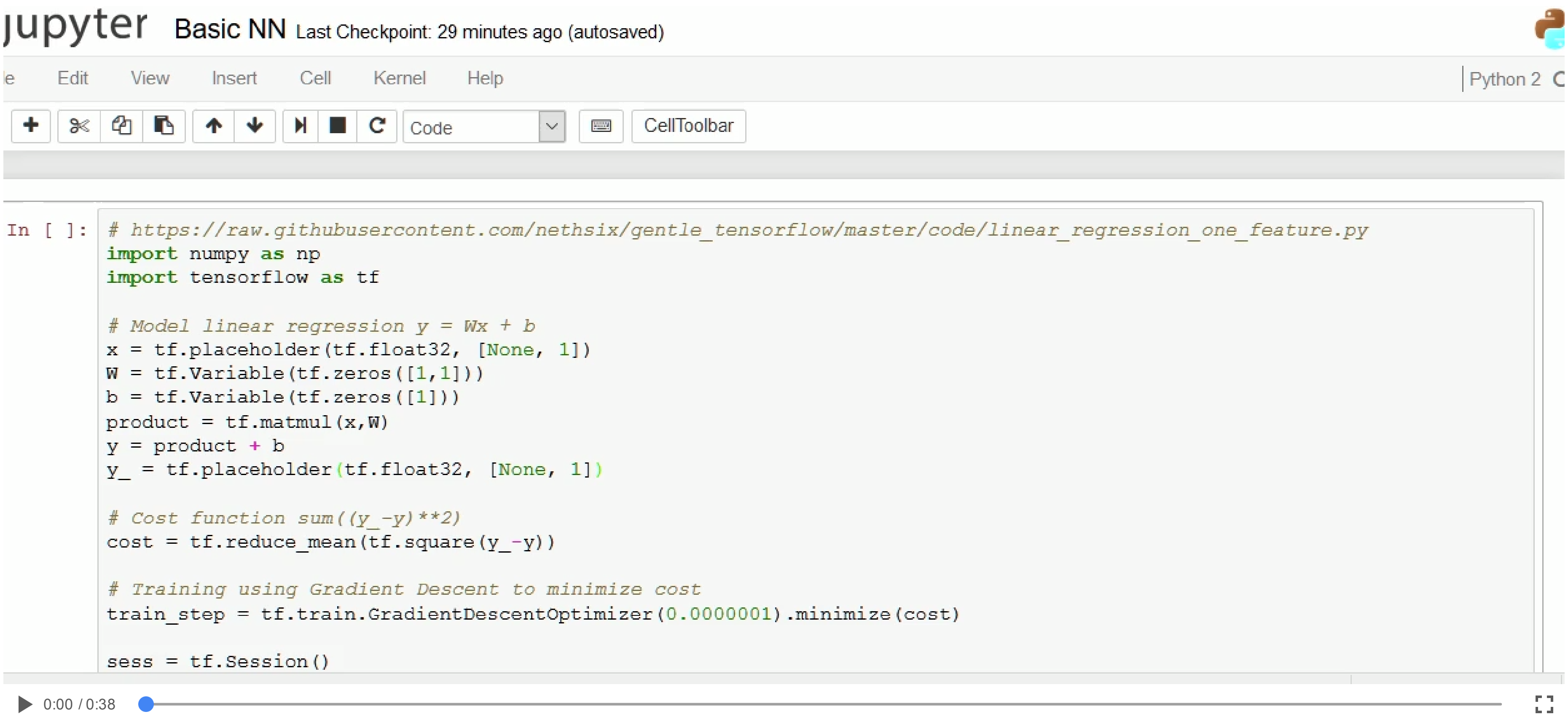
numpy is a popular, capable module that we use with TF.

Mat mult

This is how to multiply two matrices [like we'd do in a CNN!].

A linear regression learner

This short and sweet example shows we can iteratively solve for m and c for a y=mx+c line equation, given pairs of (x,y) data :)



You can use the above as a starting point for more regression experiments - multiple-linear, non-linear..

Resources

As you can imagine, we barely scratched the surface! Here are starting points for more exploring:

- these are Google's own TF tutorials
- this page from O'Reilly is a great introduction
- here are some nice examples
- tflearn - this add-on library makes it very easy to implement sophisticated NN algorithms.. Here is one way to install tflearn.
- an Udacity course

Also, Magenta (see this page, this one) is an experiment to get NNs to GENERATE art (including music)..