

Prof. Professor Brian Rowe
DATA 622: Machine Learning and Big Data
Department of Data Analytics, City University of New York

Week 2 Assignment

Docker Session

```
MINGW64/c/Users/dundeva
dundeva@SteveHome MINGW64 ~
$
dundeva@SteveHome MINGW64 ~
$ docker pull floydhub/dl-docker:cpu
cpu: Pulling from floydhub/dl-docker
Digest: sha256:377e9443b323ff2346d33b096f3bd4b7ae0a707823dd8430e093cccf59e021e9
Status: Image is up to date for floydhub/dl-docker:cpu
dundeva@SteveHome MINGW64 ~
$ docker build -t floydhub/dl-docker:cpu -f Dockerfile.cpu
"docker build" requires exactly 1 argument(s).
See 'docker build --help'.

Usage:  docker build [OPTIONS] PATH | URL | -

Build an image from a Dockerfile

dundeva@SteveHome MINGW64 ~
$ docker run -it -p 8888:8888 -p 6006:6006 -v /sharedfolder:/root/sharedfolder floydhub/dl-docker:cpu bash
root@33c214fad0ba:~#
root@33c214fad0ba:~# jupyter notebook
[I 22:42:16.731 NotebookApp] Copying /root/.ipython/kernels -> /root/.local/share/jupyter/kernels
[I 22:42:16.870 NotebookApp] Writing notebook server cookie secret to /root/.local/share/jupyter/runtime/notebook_cookie_secret
[W 22:42:17.077 NotebookApp] WARNING: The notebook server is listening on all IP addresses and not using encryption. This is not recommended.
[W 22:42:17.078 NotebookApp] WARNING: The notebook server is listening on all IP addresses and not using authentication. This is highly insecure and not recommended.
[I 22:42:17.150 NotebookApp] Serving notebooks from local directory: /root
[I 22:42:17.151 NotebookApp] 0 active kernels
[I 22:42:17.153 NotebookApp] The Jupyter Notebook is running at: http://[all ip addresses on your system]:8888/
[I 22:42:17.154 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[I 22:42:34.660 NotebookApp] 302 GET / (192.168.99.1) 1.15ms
```

Compute the gradient [2] of $z = 2x^2 - 3y^2 + 1$ at the point (3,2).

```
In [13]: import numpy as np

In [38]: %matplotlib inline

In [39]: import matplotlib.pyplot as plt

In [40]: import theano

In [41]: import theano.tensor as T

In [42]: from theano import pp

In [43]: x,y= T.dscalars('x','y')

In [64]: z= 2*x**2-3*y**2 + 1

In [65]: theano.pprint(z)
Out[65]: '(((TensorConstant{2} * (x ** TensorConstant{2}))) - (TensorConstant{3} * (y ** TensorConstant{2}))) + TensorConstant{1})'

In [70]: gy = T.grad(z,[x,y])

In [71]: f(3,2)
Out[71]: array(7.0)
```