Chapter 01: Getting Started with TensorFlow

```
ubuntu@ubuntu-PC:~$ sudo apt-get install python-pip python-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
python-dev is already the newest version (2.7.11-1).
python-pip is already the newest version (8.1.1-2ubuntu0.4).
0 upgraded, 0 newly installed, 0 to remove and 222 not upgraded.
ubuntu@ubuntu-PC:~$
```

```
    100% |[redacted]| 890kB 924kB/s
Collecting werkzeug>=0.11.10 (from tensorflow-tensorboard<0.2.0,>=0.1.0->tensor
  Downloading Werkzeug-0.12.2-py2.py3-none-any.whl (312kB)
    100% |[redacted]| 317kB 965kB/s
Collecting setuptools (from protobuf>=3.3.0->tensorflow)
  Downloading setuptools-36.5.0-py2.py3-none-any.whl (478kB)
    100% |[redacted]| 481kB 852kB/s
Installing collected packages: six, funcsigs, pbr, mock, numpy, backports.weakr
5lib, bleach, markdown, setuptools, protobuf, werkzeug, tensorflow-tensorboard,
  Found existing installation: six 1.10.0
    Not uninstalling six at /usr/lib/python2.7/dist-packages, outside environme
  Found existing installation: wheel 0.29.0
    Not uninstalling wheel at /usr/lib/python2.7/dist-packages, outside environ
  Running setup.py install for html5lib ... done
  Running setup.py install for markdown ... done
  Found existing installation: setuptools 20.7.0
    Not uninstalling setuptools at /usr/lib/python2.7/dist-packages, outside en
Successfully installed backports.weakref-1.0.post1 bleach-1.5.0 funcsigs-1.0.2
999 markdown-2.6.9 mock-2.0.0 numpy-1.13.1 pbr-3.1.1 protobuf-3.4.0 setuptools-
.0 tensorflow-1.3.0 tensorflow-tensorboard-0.1.6 werkzeug-0.12.2 wheel-0.30.0
You are using pip version 8.1.1, however version 9.0.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
ubuntu@ubuntu-PC:~$
```

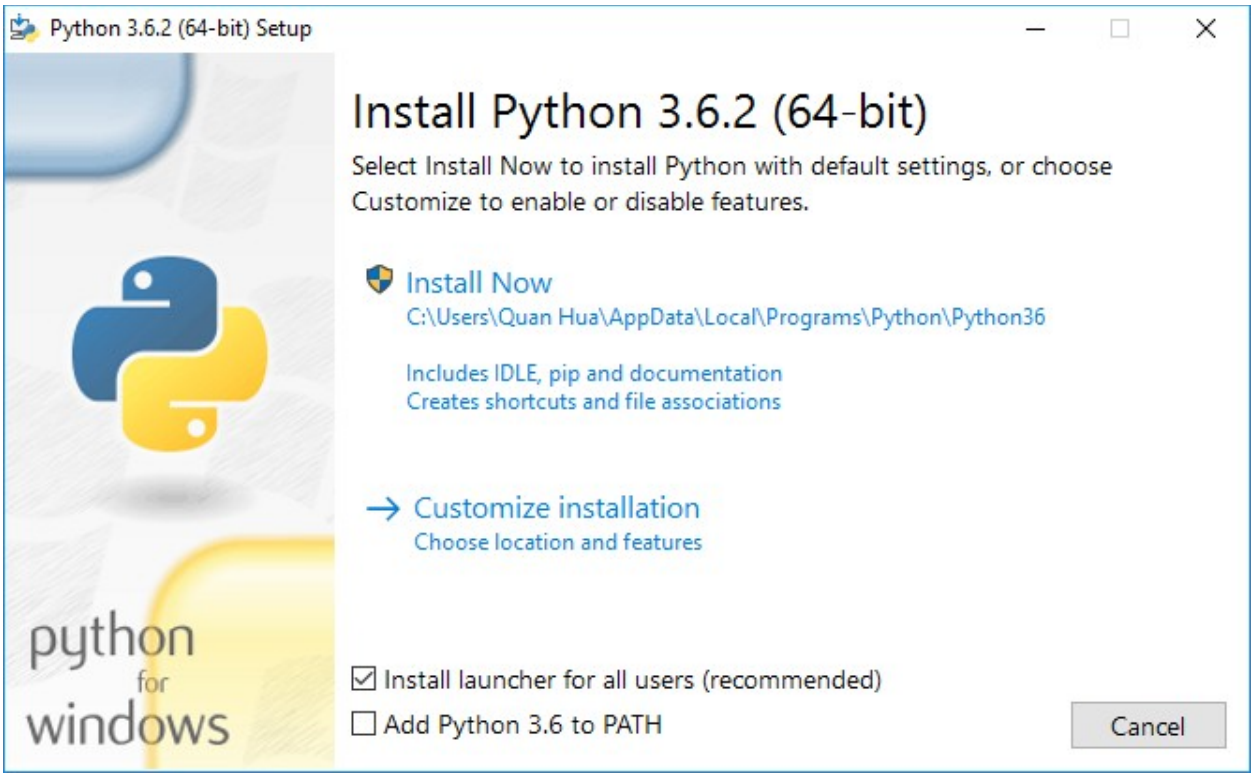
saif — -bash — 80×24

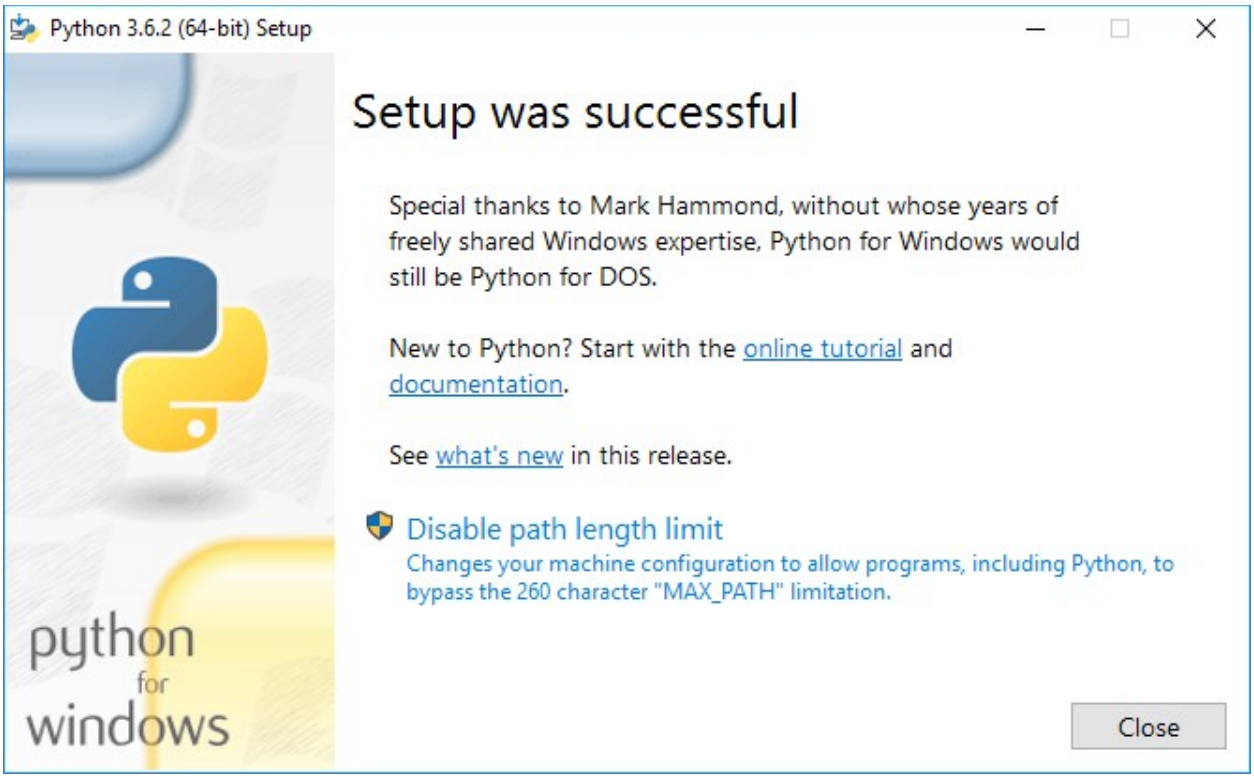
```
Last login: Thu Feb 18 12:52:14 on ttys001
~ @ alpha-al-ghaib (saif) ::  sudo easy_install pip
Password:
Searching for pip
Best match: pip 8.0.2
Adding pip 8.0.2 to easy-install.pth file
Installing pip script to /Users/saif/anaconda/bin
Installing pip2.7 script to /Users/saif/anaconda/bin
Installing pip2 script to /Users/saif/anaconda/bin

Using /Users/saif/anaconda/lib/python2.7/site-packages
Processing dependencies for pip
Finished processing dependencies for pip
```

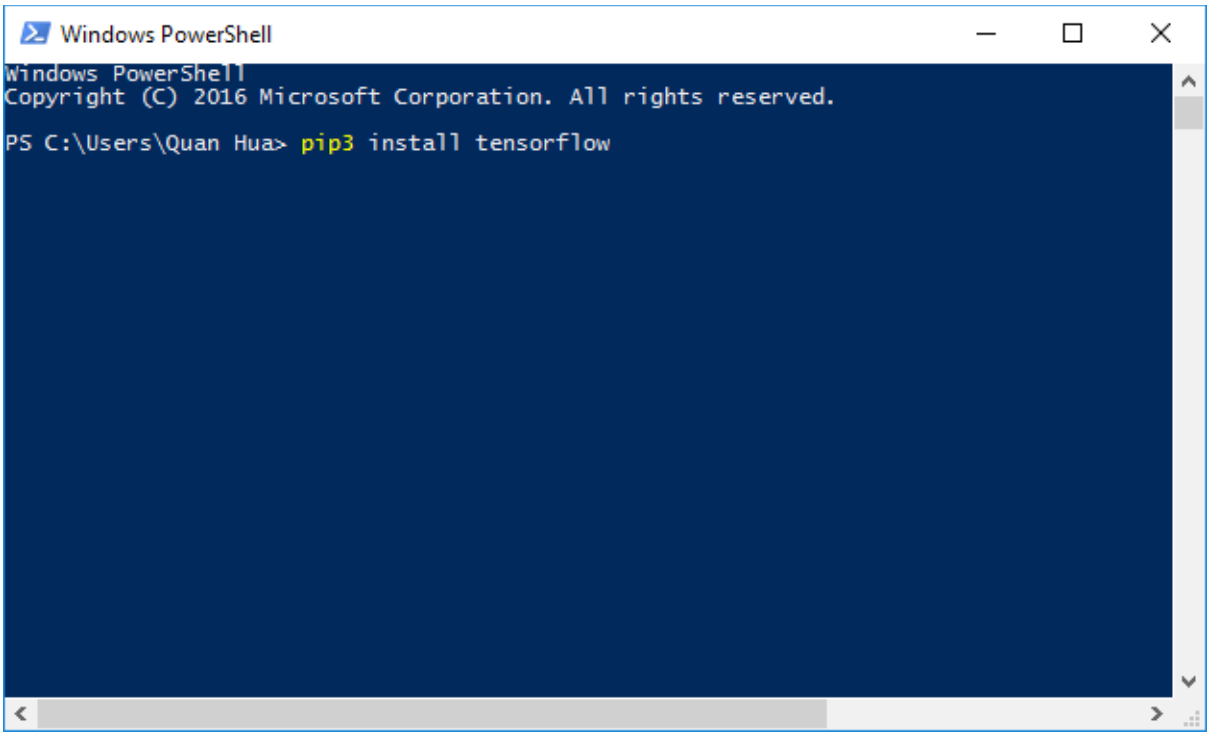
saif — -bash — 80×24

```
~ @ alpha-al-ghaib (saif) ::  sudo easy_install --upgrade six
Password:
Searching for six
Reading https://pypi.python.org/simple/six/
Best match: six 1.10.0
Processing six-1.10.0-py2.7.egg
six 1.10.0 is already the active version in easy-install.pth

Using /Users/saif/anaconda/lib/python2.7/site-packages/six-1.10.0-py2.7.egg
Processing dependencies for six
Finished processing dependencies for six
```

```
Requirement already satisfied: funcsigs>=1; python_version < "3.3" in /usr/local/lib/python2.7/site-packages (from mock>=2.0.0->tensorflow)
Requirement already satisfied: pbr>=0.11 in /usr/local/lib/python2.7/site-packages (from mock>=2.0.0->tensorflow)
Collecting html5lib==0.9999999 (from tensorflow-tensorboard<0.2.0,>=0.1.0->tensorflow)
  Downloading html5lib-0.9999999.tar.gz (889kB)
    100% |████████████████████████████████| 890kB 1.2MB/s
Collecting werkzeug>=0.11.10 (from tensorflow-tensorboard<0.2.0,>=0.1.0->tensorflow)
  Downloading Werkzeug-0.12.2-py2.py3-none-any.whl (312kB)
    100% |████████████████████████████████| 317kB 2.2MB/s
Collecting markdown>=2.6.8 (from tensorflow-tensorboard<0.2.0,>=0.1.0->tensorflow)
  Downloading Markdown-2.6.9.tar.gz (271kB)
    100% |████████████████████████████████| 276kB 3.0MB/s
Collecting bleach==1.5.0 (from tensorflow-tensorboard<0.2.0,>=0.1.0->tensorflow)
  Downloading bleach-1.5.0-py2.py3-none-any.whl
Installing collected packages: backports.weakref, protobuf, html5lib, werkzeug, markdown, bleach, tensorflow-tensorboard, tensorflow
  Found existing installation: html5lib 0.999999999
    Uninstalling html5lib-0.999999999:
      Successfully uninstalled html5lib-0.999999999
  Running setup.py install for html5lib ... done
  Running setup.py install for markdown ... done
  Found existing installation: bleach 2.0.0
    Uninstalling bleach-2.0.0:
      Successfully uninstalled bleach-2.0.0
Successfully installed backports.weakref-1.0.post1 bleach-1.5.0 html5lib-0.9999999 markdown-2.6.9 protobuf-3.4.0 tensorflow-1.3.0 tensorflow-tensorboard-0.1.7
werkzeug-0.12.2
➜ ~
```

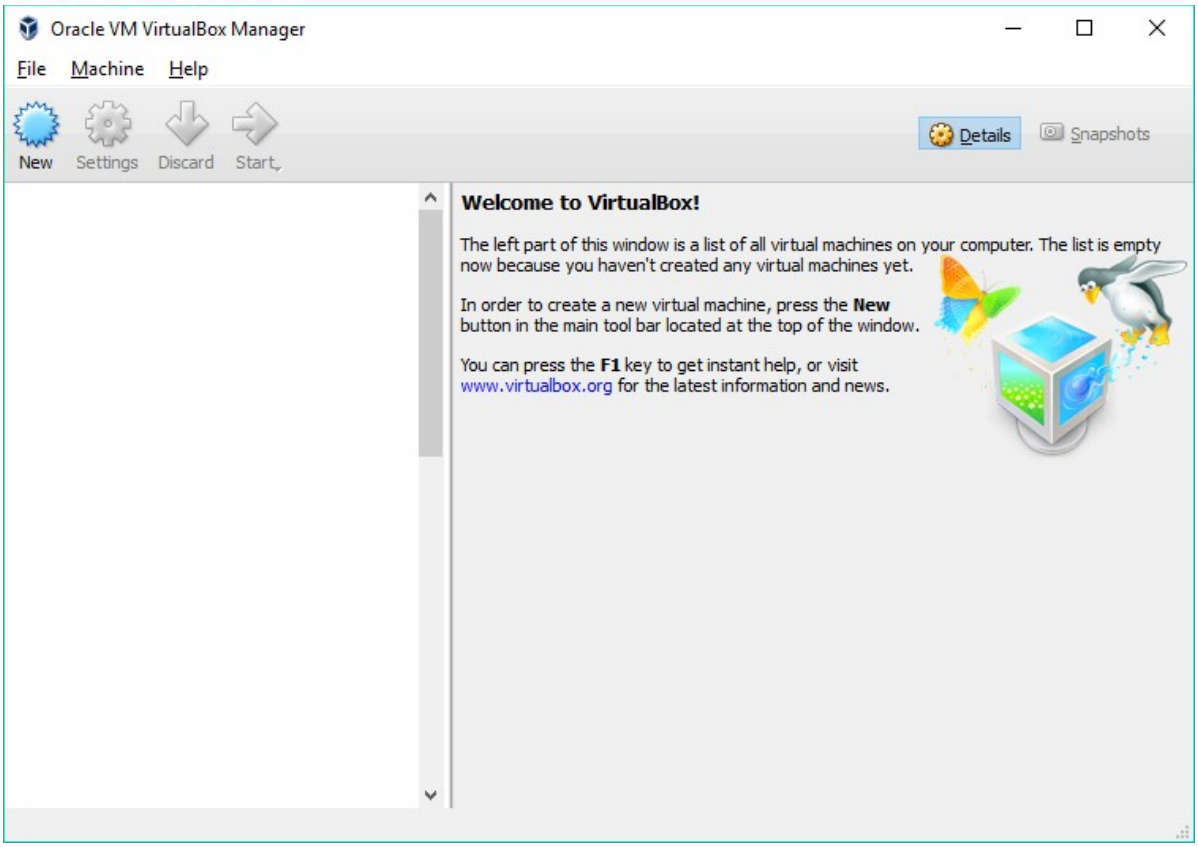

Python 3.6.2 (64-bit) Setup
Install Python 3.6.2 (64-bit)
Select Install Now to install Python with default settings, or choose Customize to enable or disable features.
Install Now
C:\Users\Quan Hua\AppData\Local\Programs\Python\Python36
Includes IDLE, pip and documentation
Creates shortcuts and file associations
Customize installation
Choose location and features
python
for
windows
Install launcher for all users (recommended)
Add Python 3.6 to PATH
Cancel

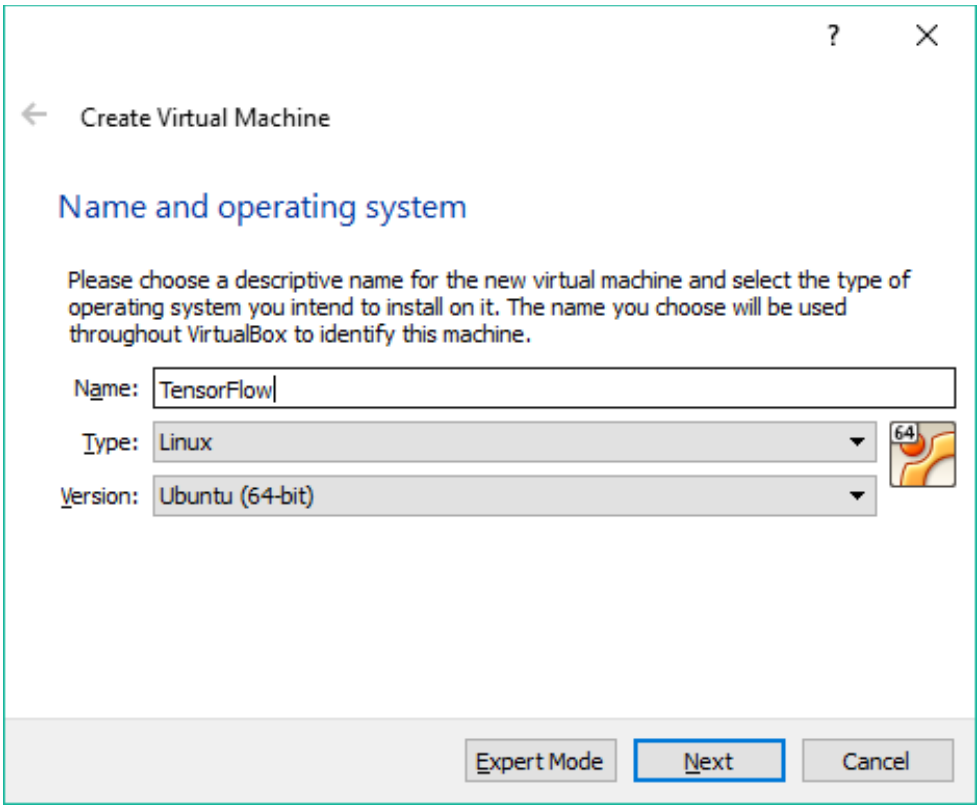

Python 3.6.2 (64-bit) Setup
Setup was successful
Special thanks to Mark Hammond, without whose years of freely shared Windows expertise, Python for Windows would still be Python for DOS.
New to Python? Start with the online tutorial and documentation.
See what's new in this release.
Disable path length limit
Changes your machine configuration to allow programs, including Python, to bypass the 260 character "MAX_PATH" limitation.
python
for
windows
Close

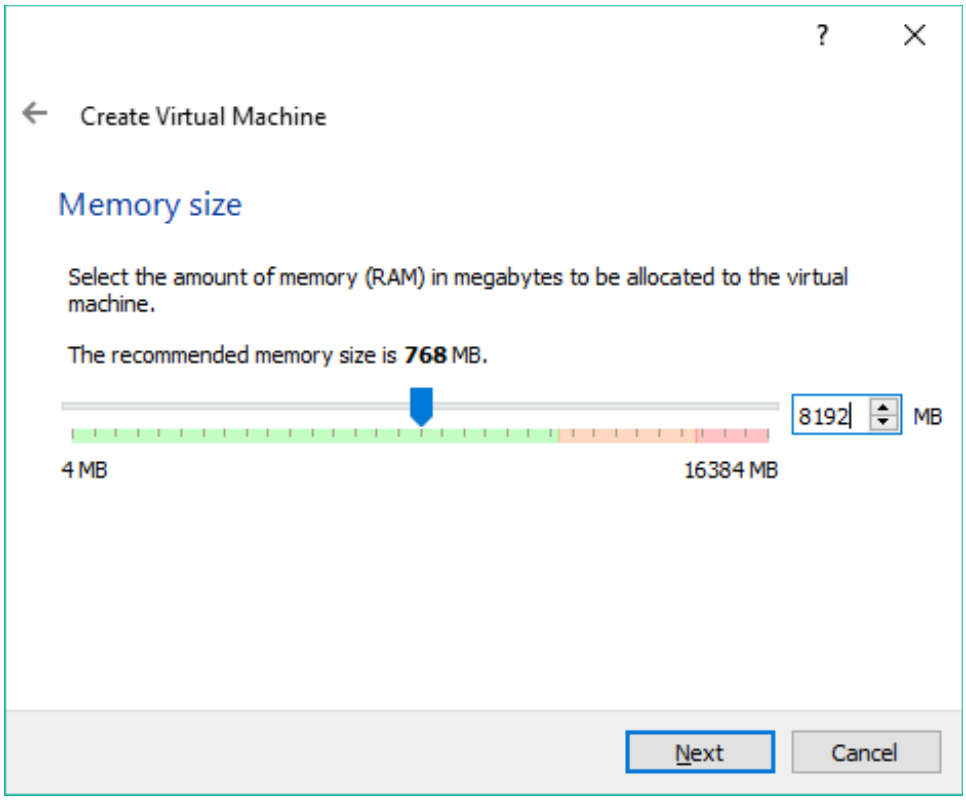

Windows PowerShell
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.
PS C:\Users\Quan Hua> pip3 install tensorflow

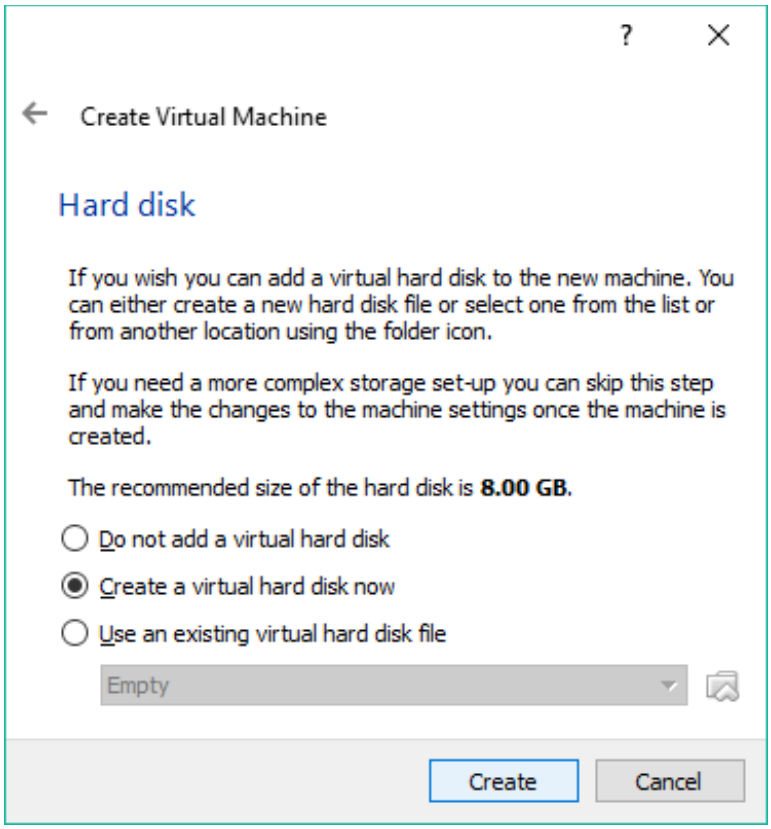
Windows PowerShell

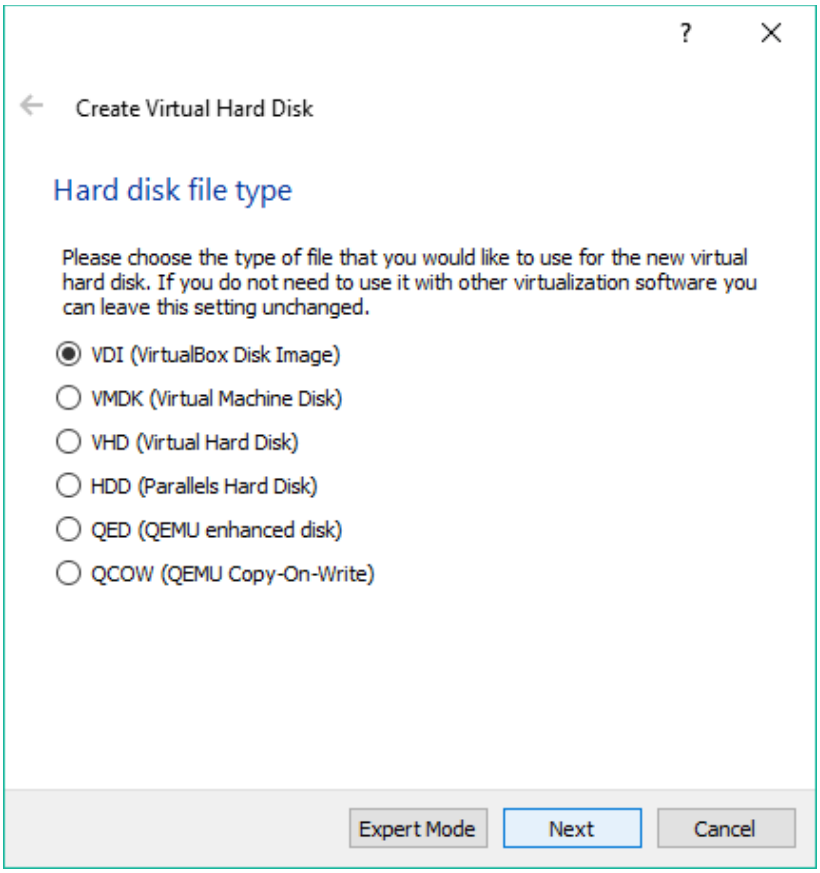
```
    100% |                                | 2.2MB 640kB/s
Collecting numpy>=1.11.0 (from tensorflow)
  Downloading numpy-1.13.1-cp36-none-win_amd64.whl (7.8MB)
    100% |                                | 7.8MB 209kB/s
Collecting wheel>=0.26 (from tensorflow)
  Downloading wheel-0.30.0-py2.py3-none-any.whl (49kB)
    100% |                                | 51kB 2.3MB/s
Collecting protobuf>=3.3.0 (from tensorflow)
  Downloading protobuf-3.4.0-py2.py3-none-any.whl (375kB)
    100% |                                | 378kB 586kB/s
Collecting six>=1.10.0 (from tensorflow)
  Downloading six-1.11.0-py2.py3-none-any.whl
Collecting bleach==1.5.0 (from tensorflow-tensorboard<0.2.0,>=0.1.0->tensorflow)
  Downloading bleach-1.5.0-py2.py3-none-any.whl
Collecting html5lib==0.9999999 (from tensorflow-tensorboard<0.2.0,>=0.1.0->tensorflow)
  Downloading html5lib-0.9999999.tar.gz (889kB)
    100% |                                | 890kB 880kB/s
Collecting werkzeug>=0.11.10 (from tensorflow-tensorboard<0.2.0,>=0.1.0->tensorflow)
  Downloading Werkzeug-0.12.2-py2.py3-none-any.whl (312kB)
    100% |                                | 317kB 969kB/s
Collecting markdown>=2.6.8 (from tensorflow-tensorboard<0.2.0,>=0.1.0->tensorflow)
  Downloading Markdown-2.6.9.tar.gz (271kB)
    100% |                                | 276kB 1.5MB/s
Requirement already satisfied: setuptools in c:\users\quan hua\appdata\local\programs\pytho
(from protobuf>=3.3.0->tensorflow)
Installing collected packages: six, protobuf, numpy, html5lib, bleach, wheel, werkzeug, mar
, tensorflow
  Running setup.py install for html5lib ... done
  Running setup.py install for markdown ... done
Successfully installed bleach-1.5.0 html5lib-0.9999999 markdown-2.6.9 numpy-1.13.1 protobuf
1.3.0 tensorflow-tensorboard-0.1.6 werkzeug-0.12.2 wheel-0.30.0
PS C:\Users\Quan Hua>
```

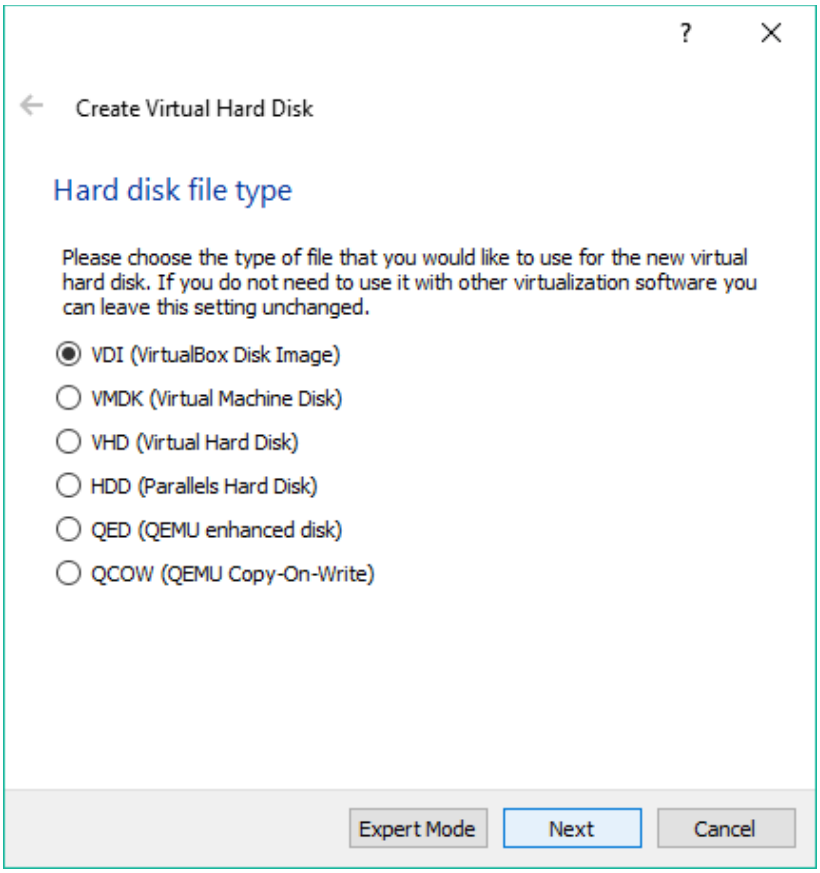

Oracle VM VirtualBox Manager
File
Machine
Help
New
Settings
Discard
Start
Details
Snapshots
Welcome to VirtualBox!
The left part of this window is a list of all virtual machines on your computer. The list is empty now because you haven't created any virtual machines yet.
In order to create a new virtual machine, press the New button in the main tool bar located at the top of the window.
You can press the F1 key to get instant help, or visit www.virtualbox.org for the latest information and news.

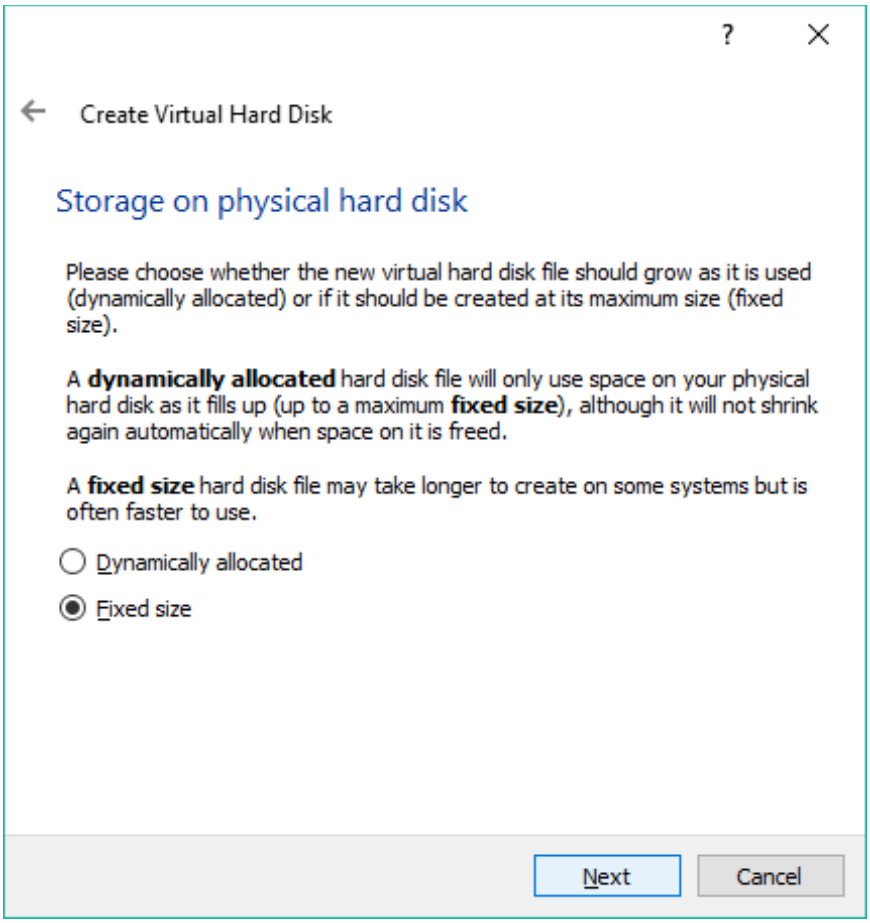

Create Virtual Machine
Name and operating system
Please choose a descriptive name for the new virtual machine and select the type of operating system you intend to install on it. The name you choose will be used throughout VirtualBox to identify this machine.
Name:
TensorFlow
Type:
Linux
Version:
Ubuntu (64-bit)
64
Expert Mode
Next
Cancel

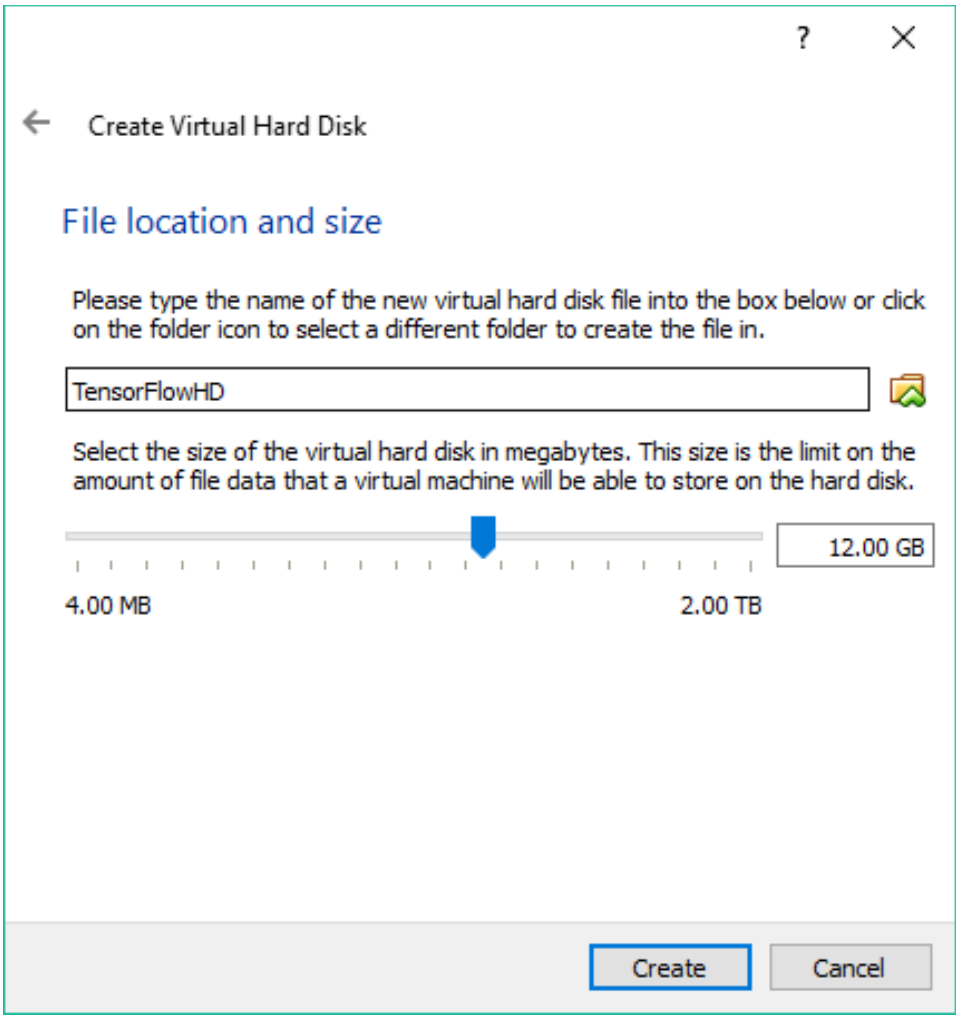

Create Virtual Machine
Memory size
Select the amount of memory (RAM) in megabytes to be allocated to the virtual machine.
The recommended memory size is 768 MB.
8192
MB
4 MB
16384 MB
Next
Cancel

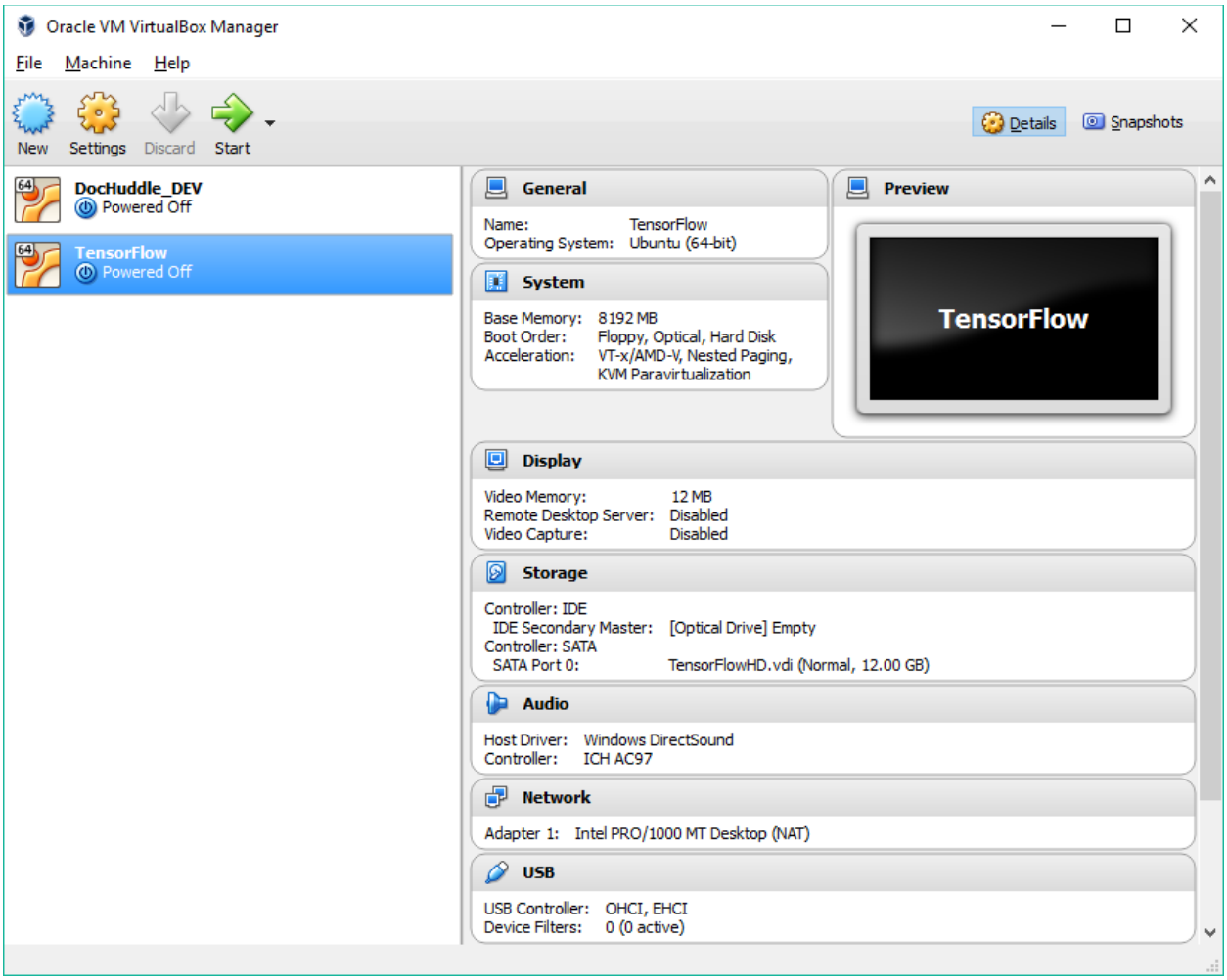

Create Virtual Machine
Hard disk
If you wish you can add a virtual hard disk to the new machine. You can either create a new hard disk file or select one from the list or from another location using the folder icon.
If you need a more complex storage set-up you can skip this step and make the changes to the machine settings once the machine is created.
The recommended size of the hard disk is 8.00 GB.
Do not add a virtual hard disk
Create a virtual hard disk now
Use an existing virtual hard disk file
Empty
Create
Cancel

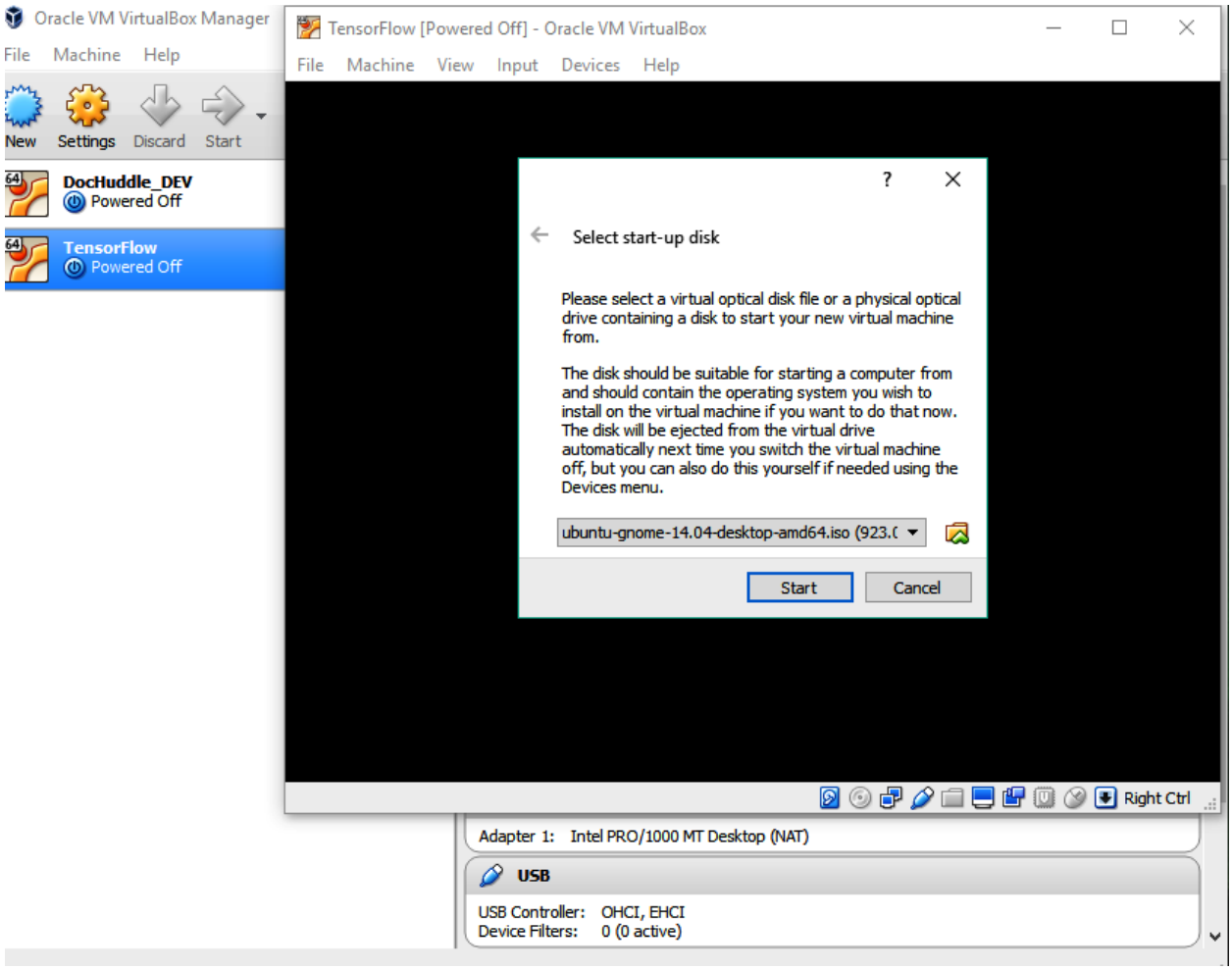

Create Virtual Hard Disk
Hard disk file type
Please choose the type of file that you would like to use for the new virtual hard disk. If you do not need to use it with other virtualization software you can leave this setting unchanged.
VDI (VirtualBox Disk Image)
VMDK (Virtual Machine Disk)
VHD (Virtual Hard Disk)
HDD (Parallels Hard Disk)
QED (QEMU enhanced disk)
QCOW (QEMU Copy-On-Write)
Expert Mode
Next
Cancel


Create Virtual Hard Disk
Hard disk file type
Please choose the type of file that you would like to use for the new virtual hard disk. If you do not need to use it with other virtualization software you can leave this setting unchanged.
VDI (VirtualBox Disk Image)
VMDK (Virtual Machine Disk)
VHD (Virtual Hard Disk)
HDD (Parallels Hard Disk)
QED (QEMU enhanced disk)
QCOW (QEMU Copy-On-Write)
Expert Mode
Next
Cancel


Create Virtual Hard Disk
Storage on physical hard disk
Please choose whether the new virtual hard disk file should grow as it is used (dynamically allocated) or if it should be created at its maximum size (fixed size).
A dynamically allocated hard disk file will only use space on your physical hard disk as it fills up (up to a maximum fixed size), although it will not shrink again automatically when space on it is freed.
A fixed size hard disk file may take longer to create on some systems but is often faster to use.
Dynamically allocated
Fixed size
Next
Cancel


Create Virtual Hard Disk
File location and size
Please type the name of the new virtual hard disk file into the box below or click on the folder icon to select a different folder to create the file in.
TensorFlowHD
Select the size of the virtual hard disk in megabytes. This size is the limit on the amount of file data that a virtual machine will be able to store on the hard disk.
12.00 GB
4.00 MB
2.00 TB
Create
Cancel


Oracle VM VirtualBox Manager
File
Machine
Help
New
Settings
Discard
Start
Details
Snapshots
DocHuddle_DEV
Powered Off
TensorFlow
Powered Off
General
Name: TensorFlow
Operating System: Ubuntu (64-bit)
System
Base Memory: 8192 MB
Boot Order: Floppy, Optical, Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging, KVM Paravirtualization
Preview
TensorFlow
Display
Video Memory: 12 MB
Remote Desktop Server: Disabled
Video Capture: Disabled
Storage
Controller: IDE
IDE Secondary Master: [Optical Drive] Empty
Controller: SATA
SATA Port 0: TensorFlowHD.vdi (Normal, 12.00 GB)
Audio
Host Driver: Windows DirectSound
Controller: ICH AC97
Network
Adapter 1: Intel PRO/1000 MT Desktop (NAT)
USB
USB Controller: OHCI, EHCI
Device Filters: 0 (0 active)

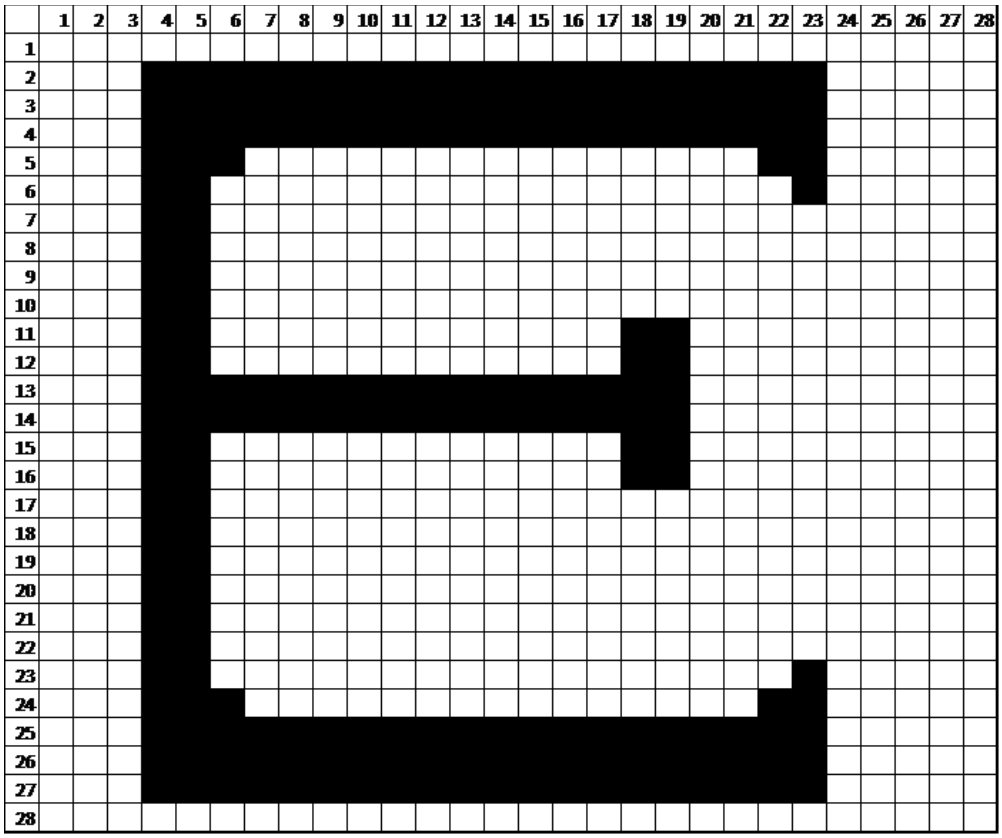

Oracle VM VirtualBox Manager
File Machine Help
New Settings Discard Start
DocHuddle_DEV
Powered Off
TensorFlow
Powered Off
TensorFlow [Powered Off] - Oracle VM VirtualBox
File Machine View Input Devices Help
Select start-up disk
Please select a virtual optical disk file or a physical optical drive containing a disk to start your new virtual machine from.
The disk should be suitable for starting a computer from and should contain the operating system you wish to install on the virtual machine if you want to do that now. The disk will be ejected from the virtual drive automatically next time you switch the virtual machine off, but you can also do this yourself if needed using the Devices menu.
ubuntu-gnome-14.04-desktop-amd64.iso (923.(
Start
Cancel
Right Ctrl
Adapter 1: Intel PRO/1000 MT Desktop (NAT)
USB
USB Controller: OHCI, EHCI
Device Filters: 0 (0 active)

Windows PowerShell

```
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Users\Quan Hua> python
Python 3.6.2 (v3.6.2:5fd33b5, Jul  8 2017, 04:57:36) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import tensorflow as tf
>>> a = tf.constant(1.0)
>>> b = tf.constant(2.0)
>>> c = a + b
>>> sess = tf.Session()
2017-09-27 01:35:50.710620: W C:\tf_jenkins\home\workspace\rel-win\M\windows\PY\36\tensorflow\core\
guard.cc:45] The TensorFlow library wasn't compiled to use AVX instructions, but these are availabl
 could speed up CPU computations.
2017-09-27 01:35:50.710761: W C:\tf_jenkins\home\workspace\rel-win\M\windows\PY\36\tensorflow\core\
guard.cc:45] The TensorFlow library wasn't compiled to use AVX2 instructions, but these are availab
d could speed up CPU computations.
>>> print(sess.run(c))
3.0
>>>
```

Chapter 02: Your First Classifier

```
/notMNIST_small/J
Started loading images from: /home/ubuntu/github/mlwithtf/datasets/notMNIST/test
/notMNIST_small/J
Finished loading data from: /home/ubuntu/github/mlwithtf/datasets/notMNIST/test/
notMNIST_small/J
        Started pickling: J
Finished pickling: J
Finished loading testing data
Started pickling final dataset
Merging train, valid data
Merging test data
('Training set', (200000, 28, 28), (200000,))
('Validation set', (10000, 28, 28), (10000,))
('Test set', (10000, 28, 28), (10000,))
('Compressed pickle size:', 690800514)
Finished pickling final dataset
Finished preparing notMNIST dataset
After reformat:
('Training set', (200000, 784), (200000, 10))
('Validation set', (10000, 784), (10000, 10))
('Test set', (10000, 784), (10000, 10))
(work2) ubuntu@ubuntu-PC:~/github/mlwithtf/chapter_02$
```



Anaconda Prompt

```
  0.19  0.01 -0.14 -0.5 ]
[-0.5  -0.5  -0.5  -0.5  -0.5  -0.5  -0.48 -0.48 -0.43 -0.35 -0.5  -0.49
 -0.49 -0.5  -0.42 -0.49  0.2   0.05  0.46  0.5   0.5   0.5   0.49  0.4
  0.15 -0.29 -0.12 -0.5 ]
[-0.5  -0.5  -0.5  -0.45 -0.42 -0.46 -0.05 -0.21 -0.35 -0.44 -0.5  -0.5
 -0.45 -0.48 -0.49 -0.36  0.12  0.25  0.5   0.5   0.5   0.5   0.48  0.36
  0.28 -0.08 -0.48 -0.5 ]
[-0.49 -0.5  -0.5  -0.48 -0.15  0.02  0.45  0.47  0.39  0.21 -0.07 -0.06
 -0.22 -0.43 -0.38  0.24  0.24  0.48  0.5   0.5   0.5   0.5   0.5   0.43
  0.1  -0.39 -0.47 -0.43]
[-0.48 -0.44 -0.36 -0.02  0.34  0.45  0.45  0.5   0.5   0.5   0.43  0.29
  0.03 -0.08  0.15  0.48  0.5   0.5   0.5   0.5   0.5   0.48  0.5   0.06
  0.03 -0.27 -0.33 -0.46]
[-0.43 -0.5  -0.2   0.48  0.45  0.45  0.49  0.5   0.49  0.48  0.47  0.5
  0.46  0.49  0.5   0.5   0.49  0.5   0.5   0.49  0.5   0.32  0.42  0.06
 -0.1  -0.3  -0.46 -0.5 ]
[-0.5  -0.41 -0.43 -0.21 -0.23  0.13  0.46  0.48  0.5   0.5   0.5   0.49
  0.5   0.48  0.47  0.49  0.5   0.48  0.48  0.5   0.32  0.2   0.35 -0.37
 -0.44 -0.33 -0.5  -0.49]
[-0.5  -0.5  -0.5  -0.39 -0.36 -0.1   0.16  0.14  0.44  0.42  0.47  0.48
  0.5   0.5   0.5   0.48  0.46  0.5   0.5   0.41  0.07  0.04 -0.17 -0.45
 -0.47 -0.5  -0.5  -0.5 ]
[-0.5  -0.5  -0.49 -0.45 -0.47 -0.43 -0.37 -0.35 -0.17 -0.02  0.17  0.23
  0.01  0.02  0.15  0.3   0.18  0.01 -0.06 -0.14 -0.15 -0.41 -0.44 -0.5
 -0.5  -0.49 -0.5  -0.5 ]
[-0.5  -0.5  -0.5  -0.5  -0.5  -0.5  -0.5  -0.5  -0.5  -0.5  -0.5  -0.45
 -0.42 -0.37 -0.26 -0.23 -0.32 -0.48 -0.5  -0.47 -0.37 -0.49 -0.5  -0.5
 -0.5  -0.5  -0.5  -0.5 ]]

[anaconda] S:\__WORKSPACE\PyCharm\book>
```

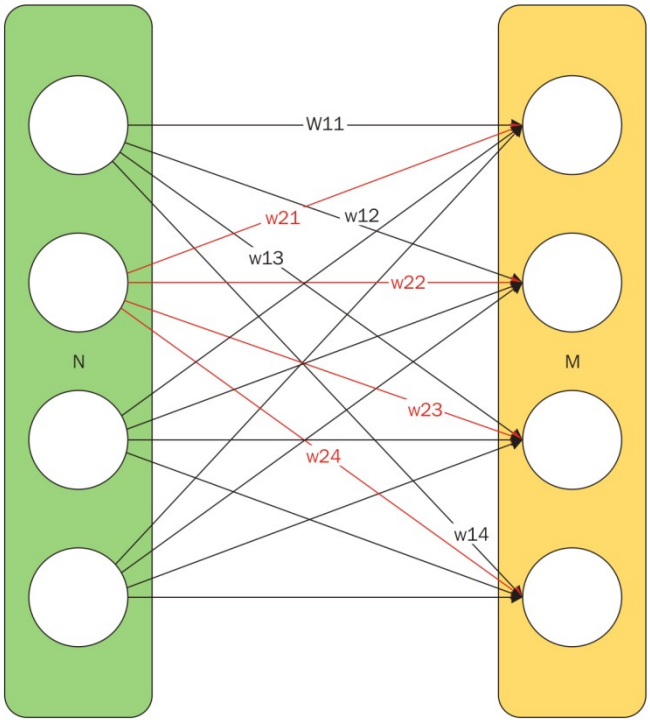
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
1																												
2				255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255					
3				255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255					
4				255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255					
5				255	255	253																251	253					
6				255	255																		251					
7				255	255																							
8				255	255																							
9				255	255																							
10				255	255																							
11				255	255													251	253									
12				255	255													255	255									
13				255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255									
14				255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255									
15				255	255													255	255									
16				255	255													251	253									
17				255	255																							
18				255	255																							
19				255	255																							
20				255	255																							
21				255	255																							
22				255	255																							
23				255	255																		251					
24				255	255	253																251	253					
25				255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255					
26				255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255					
27				255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255					
28																												

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
1	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50
2	-0.50	-0.50	-0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	-0.50	-0.50	-0.50	-0.50	-0.50
3	-0.50	-0.50	-0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	-0.50	-0.50	-0.50	-0.50	-0.50
4	-0.50	-0.50	-0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	-0.50	-0.50	-0.50	-0.50	-0.50
5	-0.50	-0.50	-0.50	0.50	0.50	0.49	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	0.48	0.49	-0.50	-0.50	-0.50	-0.50	-0.50
6	-0.50	-0.50	-0.50	0.50	0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	0.48	-0.50	-0.50	-0.50	-0.50	-0.50
7	-0.50	-0.50	-0.50	0.50	0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50
8	-0.50	-0.50	-0.50	0.50	0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50
9	-0.50	-0.50	-0.50	0.50	0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50
10	-0.50	-0.50	-0.50	0.50	0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50
11	-0.50	-0.50	-0.50	0.50	0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	0.48	0.49	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50
12	-0.50	-0.50	-0.50	0.50	0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	0.50	0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50
13	-0.50	-0.50	-0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50
14	-0.50	-0.50	-0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50
15	-0.50	-0.50	-0.50	0.50	0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	0.50	0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50
16	-0.50	-0.50	-0.50	0.50	0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	0.48	0.49	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50
17	-0.50	-0.50	-0.50	0.50	0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50
18	-0.50	-0.50	-0.50	0.50	0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50
19	-0.50	-0.50	-0.50	0.50	0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50
20	-0.50	-0.50	-0.50	0.50	0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50
21	-0.50	-0.50	-0.50	0.50	0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50
22	-0.50	-0.50	-0.50	0.50	0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50
23	-0.50	-0.50	-0.50	0.50	0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	0.48	-0.50	-0.50	-0.50	-0.50	-0.50
24	-0.50	-0.50	-0.50	0.50	0.50	0.49	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	0.48	0.49	-0.50	-0.50	-0.50	-0.50	-0.50
25	-0.50	-0.50	-0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	-0.50	-0.50	-0.50	-0.50	-0.50
26	-0.50	-0.50	-0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	-0.50	-0.50	-0.50	-0.50	-0.50
27	-0.50	-0.50	-0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	-0.50	-0.50	-0.50	-0.50	-0.50
28	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50

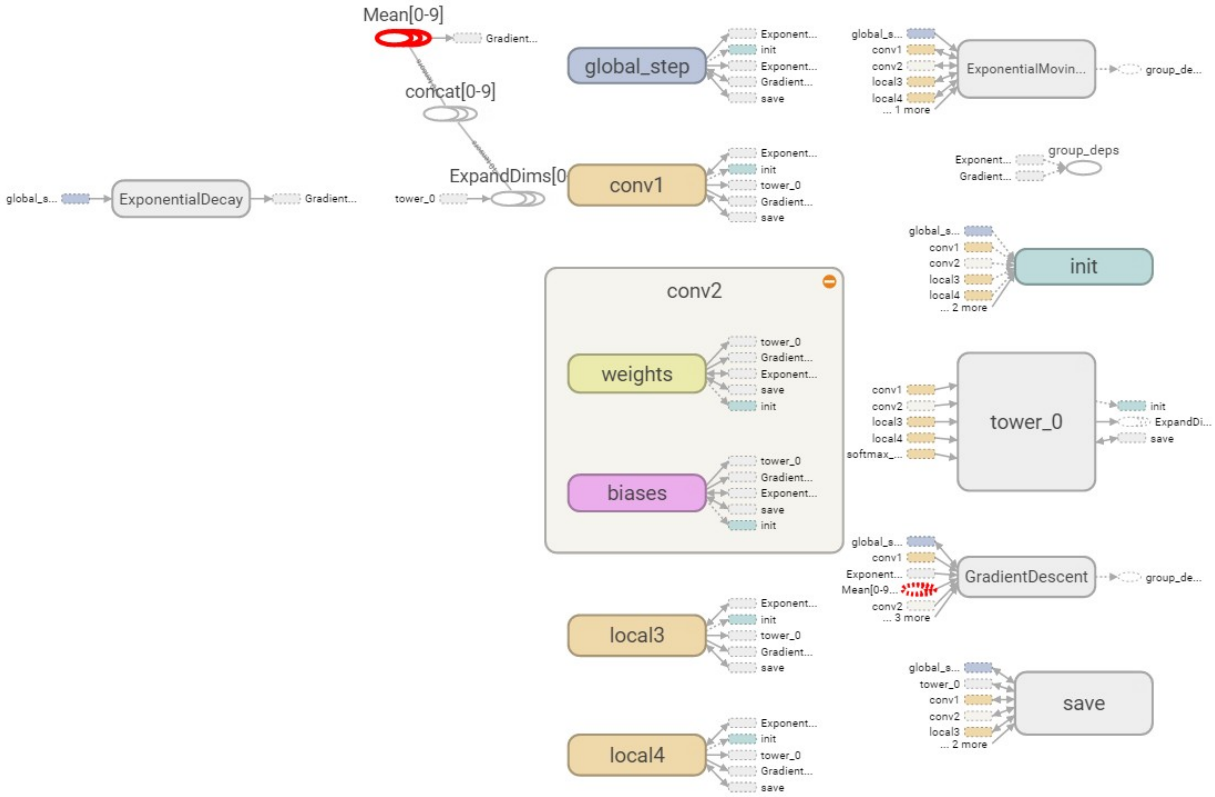
	1	2	3	4	5	6	7	...	780	781	782	783	784
1	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	-0.50	...	-0.50	-0.50	-0.50	-0.50	-0.50
2	-0.50	-0.50	-0.49	-0.50	-0.50	-0.50	0.50	...	0.50	0.50	0.50	0.50	0.50
3	-0.50	-0.50	-0.49	-0.50	-0.50	-0.50	0.50	...	0.50	0.50	0.50	0.50	0.50
...	...	...	...	...	...	...	...	...	...	...	...	...	...
200000	-0.50	-0.50	-0.40	-0.50	-0.50	-0.50	0.50	...	0.50	0.50	0.50	0.50	0.50
200001	-0.50	-0.50	-0.40	-0.50	-0.50	-0.50	0.50	...	0.50	0.50	0.50	0.50	0.50
200002	-0.50	-0.50	-0.40	-0.50	-0.50	-0.50	0.50	...	0.50	0.50	0.50	0.50	0.50
200003	-0.50	-0.50	-0.39	-0.50	-0.50	-0.50	-0.50	...	-0.50	-0.50	-0.50	-0.50	-0.50

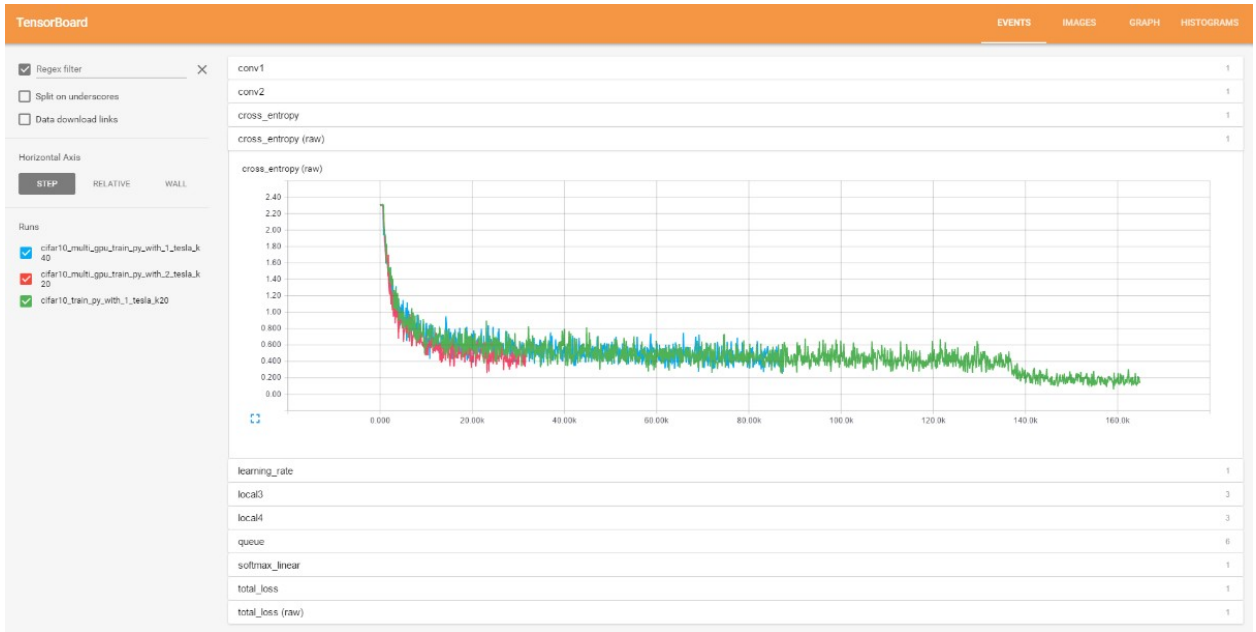
```
with tf.Session(graph=graph) as session:
    session.run(tf.global_variables_initializer())
    print("Initialized")
    for step in range(num_steps + 1):
        sys.stdout.write('Training on batch %d of %d\r' % (step + 1, num_steps))
        sys.stdout.flush()
        # Pick an offset within the training data, which has been randomized.
        # Note: we could use better randomization across epochs.
        offset = (step * batch_size) % (dataset.train_labels.shape[0] - batch_size)
        # Generate a minibatch.
        batch_data = dataset.train_dataset[offset:(offset + batch_size), :]
        batch_labels = dataset.train_labels[offset:(offset + batch_size), :]
        # Prepare a dictionary telling the session where to feed the minibatch.
        # The key of the dictionary is the placeholder node of the graph to be fed,
        # and the value is the numpy array to feed to it.
        feed_dict = {tf_train_dataset: batch_data, tf_train_labels: batch_labels}
        _, l, predictions = session.run([optimizer, loss, train_prediction], feed_dict=feed_dict)
        if step % data_showing_step == 0:
            acc_minibatch = accuracy(predictions, batch_labels)
            acc_val = accuracy(valid_prediction.eval(), dataset.valid_labels)
            logmanager.logger.info('# %03d  Acc Train: %03.2f%%  Acc Val: %03.2f%% Loss %f' % (
                step, acc_minibatch, acc_val, l))
    logmanager.logger.info("Test accuracy: %.1f%%" % accuracy(test_prediction.eval(), dataset.test_labels))
```

```
Initialized
2017-09-29 03:33:28,900 - MLwithTF - INFO - # 000  Acc Train: 11.72%  Acc Val: 18.66% Loss 33.751575
2017-09-29 03:33:29,295 - MLwithTF - INFO - # 500  Acc Train: 68.75%  Acc Val: 69.35% Loss 1.133016
2017-09-29 03:33:29,683 - MLwithTF - INFO - # 1000  Acc Train: 66.41%  Acc Val: 68.23% Loss 0.997880
2017-09-29 03:33:30,101 - MLwithTF - INFO - # 1500  Acc Train: 77.34%  Acc Val: 75.75% Loss 0.744131
2017-09-29 03:33:30,491 - MLwithTF - INFO - # 2000  Acc Train: 69.53%  Acc Val: 76.69% Loss 1.102222
2017-09-29 03:33:30,897 - MLwithTF - INFO - # 2500  Acc Train: 69.53%  Acc Val: 77.12% Loss 0.942900
2017-09-29 03:33:31,290 - MLwithTF - INFO - # 3000  Acc Train: 80.47%  Acc Val: 77.96% Loss 0.652119
2017-09-29 03:33:31,689 - MLwithTF - INFO - # 3500  Acc Train: 77.34%  Acc Val: 78.18% Loss 0.738713
2017-09-29 03:33:32,088 - MLwithTF - INFO - # 4000  Acc Train: 75.78%  Acc Val: 75.89% Loss 0.811817
2017-09-29 03:33:32,486 - MLwithTF - INFO - # 4500  Acc Train: 76.56%  Acc Val: 71.15% Loss 0.767980
2017-09-29 03:33:32,890 - MLwithTF - INFO - # 5000  Acc Train: 78.91%  Acc Val: 79.68% Loss 0.725805
2017-09-29 03:33:33,291 - MLwithTF - INFO - # 5500  Acc Train: 78.91%  Acc Val: 79.76% Loss 0.720853
2017-09-29 03:33:33,688 - MLwithTF - INFO - # 6000  Acc Train: 81.25%  Acc Val: 79.60% Loss 0.630755
2017-09-29 03:33:34,094 - MLwithTF - INFO - # 6500  Acc Train: 81.25%  Acc Val: 79.94% Loss 0.717576
2017-09-29 03:33:34,483 - MLwithTF - INFO - # 7000  Acc Train: 82.03%  Acc Val: 79.53% Loss 0.730982
2017-09-29 03:33:34,870 - MLwithTF - INFO - # 7500  Acc Train: 82.81%  Acc Val: 80.48% Loss 0.606189
2017-09-29 03:33:35,257 - MLwithTF - INFO - # 8000  Acc Train: 78.91%  Acc Val: 79.64% Loss 0.683653
2017-09-29 03:33:35,647 - MLwithTF - INFO - # 8500  Acc Train: 82.81%  Acc Val: 80.42% Loss 0.643934
2017-09-29 03:33:36,023 - MLwithTF - INFO - # 9000  Acc Train: 83.59%  Acc Val: 80.34% Loss 0.608325
2017-09-29 03:33:36,420 - MLwithTF - INFO - # 9500  Acc Train: 82.81%  Acc Val: 80.38% Loss 0.595343
2017-09-29 03:33:36,820 - MLwithTF - INFO - # 10000  Acc Train: 74.22%  Acc Val: 80.62% Loss 0.943106
2017-09-29 03:33:36,840 - MLwithTF - INFO - Test accuracy: 87.4%
(work2) ubuntu@ubuntu-PC:~/github/mlwithtf/chapter_02$
```

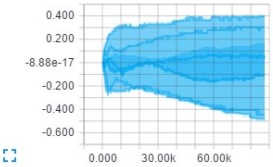


Chapter 03: The TensorFlow Toolbox

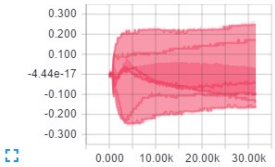



TensorBoard
EVENTS
IMAGES
GRAPH
HISTOGRAMS
Regex filter
Split on underscores
Data download links
Horizontal Axis
STEP
RELATIVE
WALL
Runs
cifar10_multi_gpu_train_py_with_1_tesla_k40
cifar10_multi_gpu_train_py_with_2_tesla_k20
cifar10_train_py_with_1_tesla_k20
conv1
conv2
cross_entropy
cross_entropy (raw)
cross_entropy (raw)
2.40
2.20
2.00
1.80
1.60
1.40
1.20
1.00
0.800
0.600
0.400
0.200
0.00
0.000
20.00k
40.00k
60.00k
80.00k
100.0k
120.0k
140.0k
160.0k
learning_rate
local3
local4
queue
softmax_linear
total_loss
total_loss (raw)

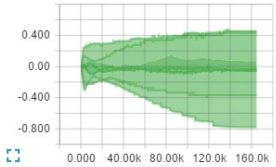
softmax_linear/biases



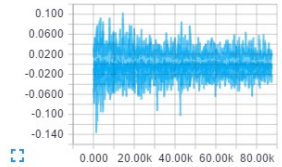
softmax_linear/biases



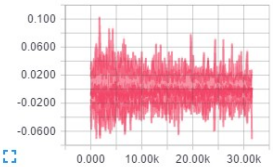
softmax_linear/biases



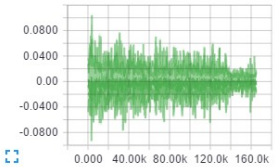
softmax_linear/biases/gradients



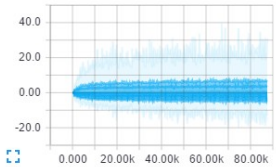
softmax_linear/biases/gradients



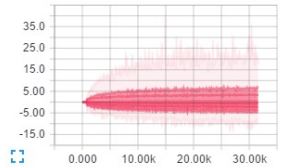
softmax_linear/biases/gradients



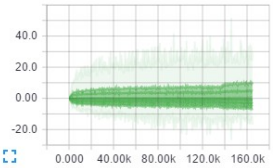
softmax_linear/softmax_linear/activations



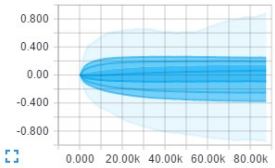
softmax_linear/softmax_linear/activations



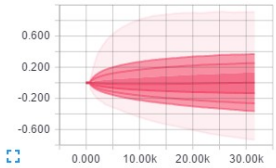
softmax_linear/softmax_linear/activations



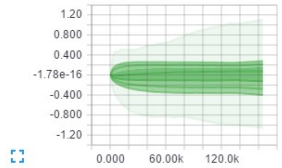
softmax_linear/weights



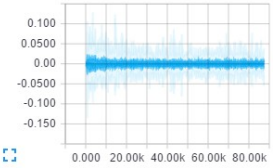
softmax_linear/weights



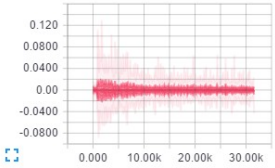
softmax_linear/weights



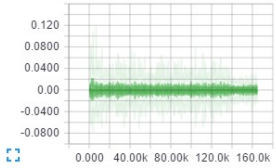
softmax_linear/weights/gradients



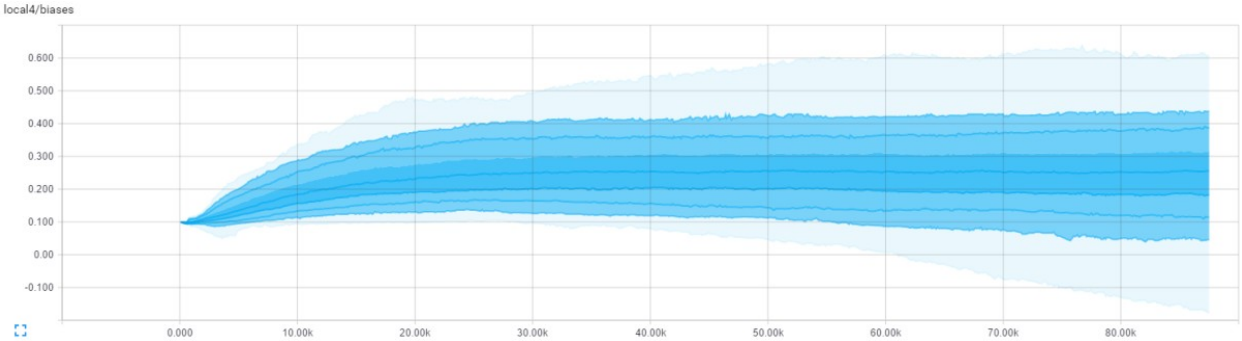
softmax_linear/weights/gradients



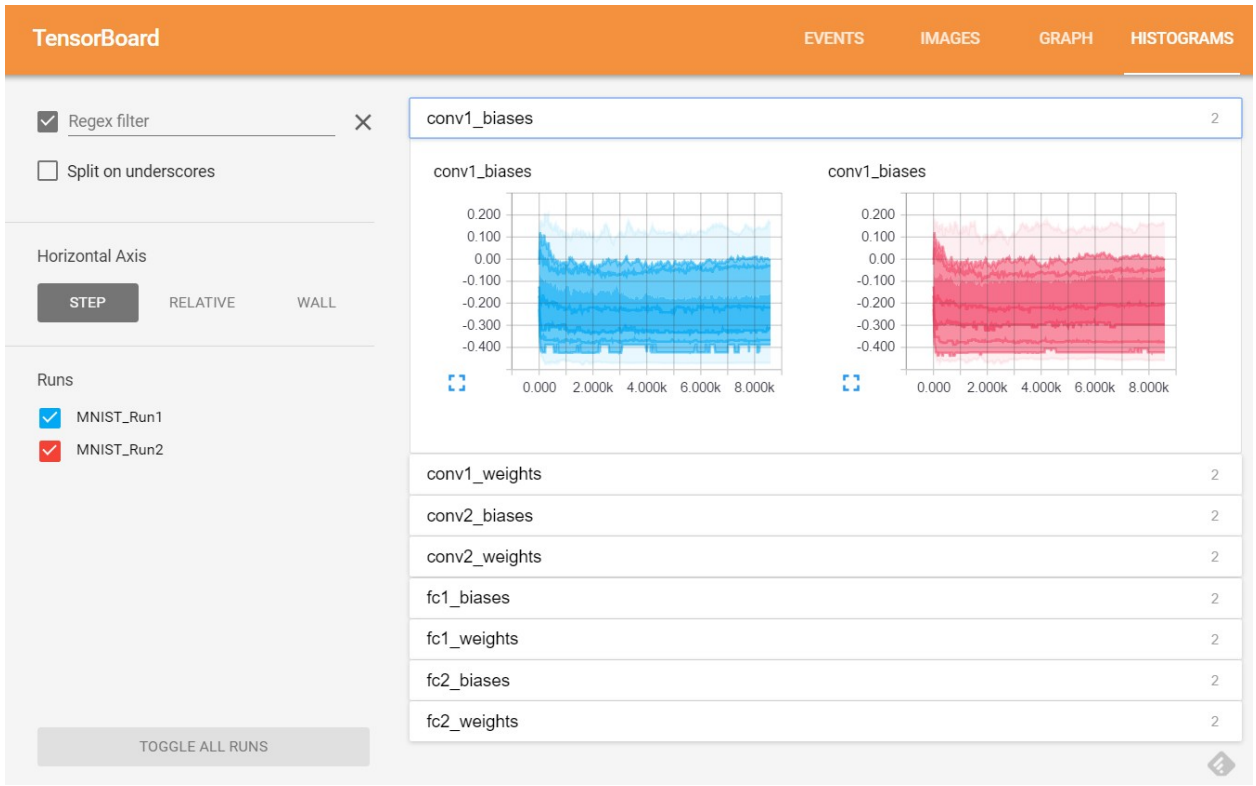
softmax_linear/weights/gradients

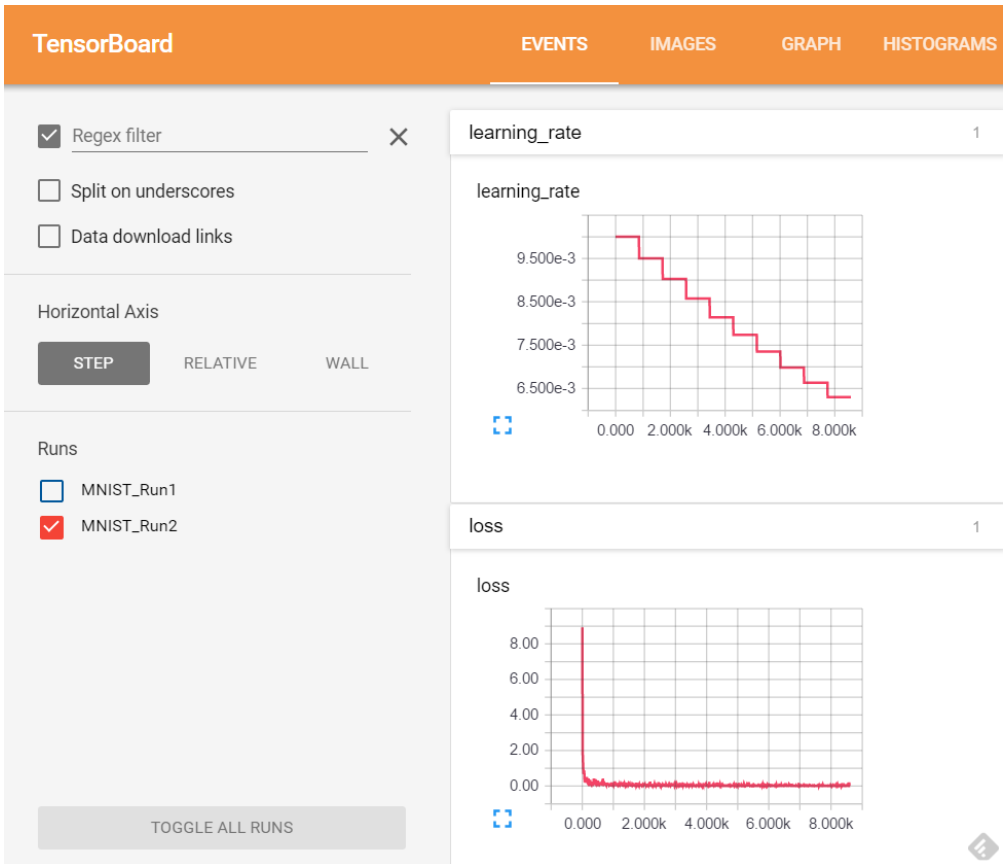


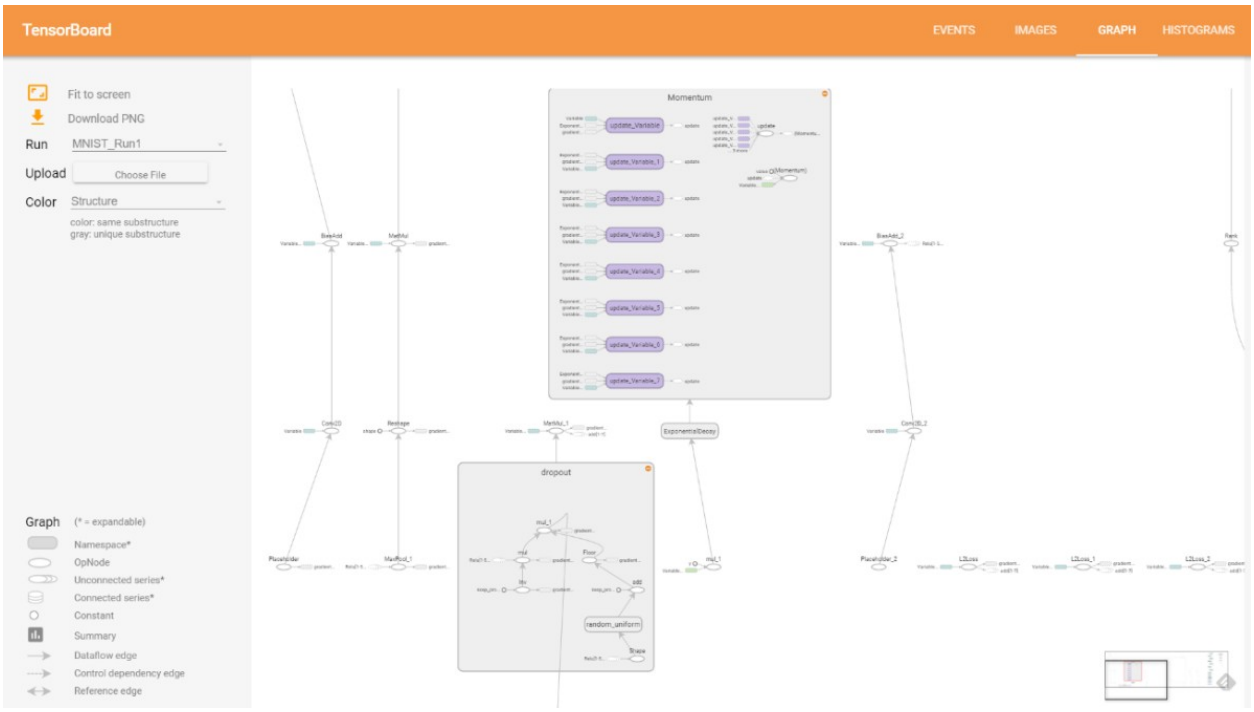
local4/biases

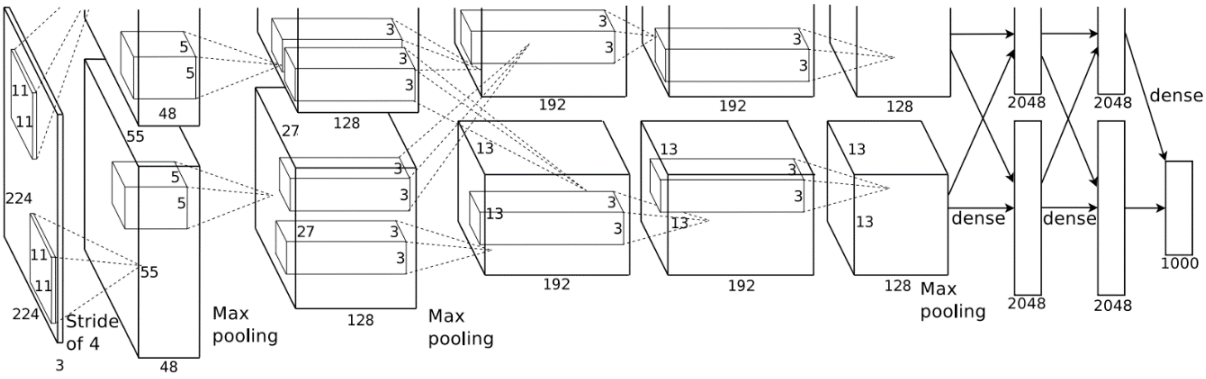


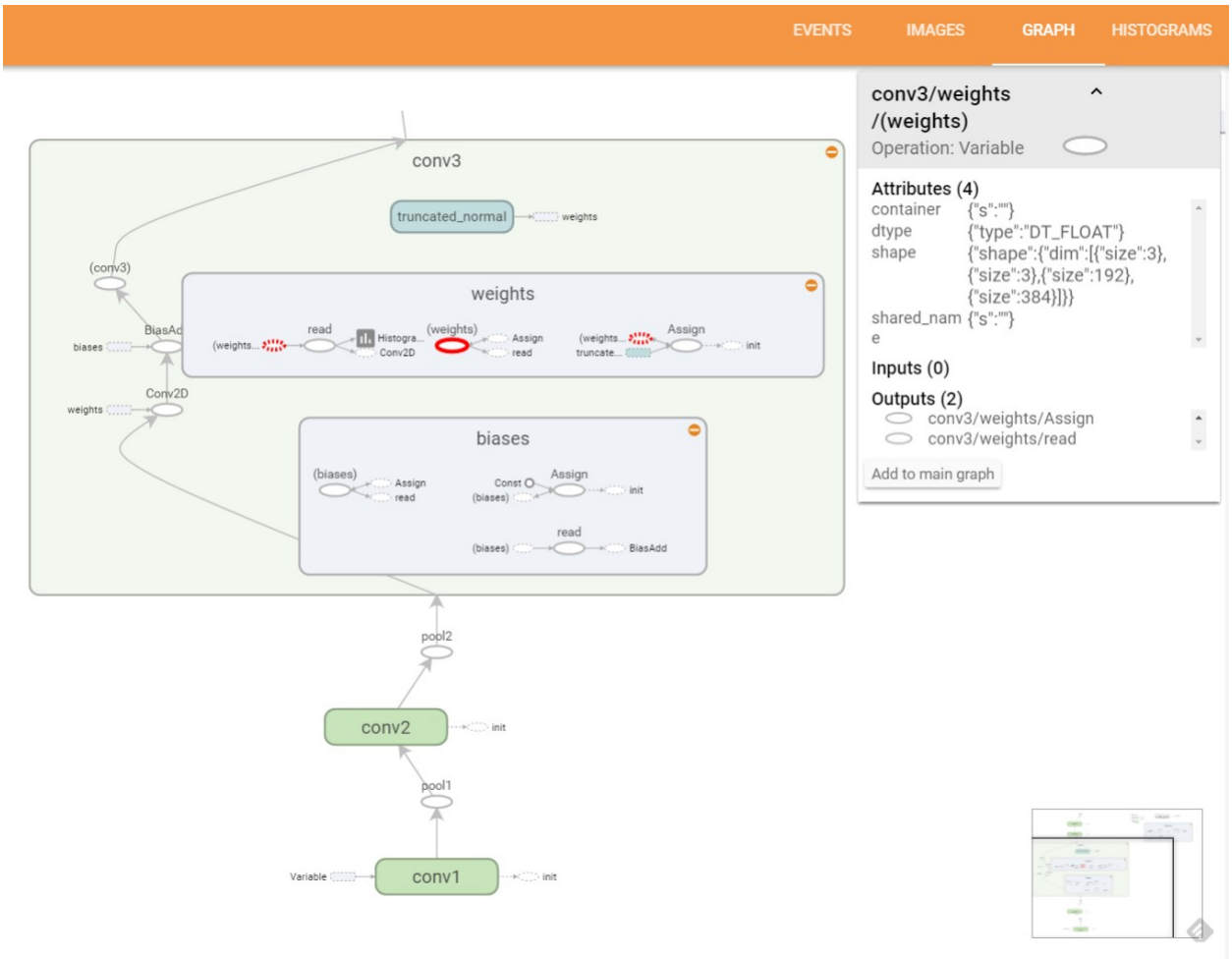

TensorBoard
EVENTS
IMAGES
GRAPH
HISTOGRAMS
Regex filter
Split on underscores
Data download links
Horizontal Axis
STEP
RELATIVE
WALL
Runs
MNIST_Run1
MNIST_Run2
TOGGLE ALL RUNS
learning_rate
0.800
0.400
0.00
-0.400
-0.800
0.000
100.0
200.0
300.0
400.0
loss
1
loss
9.00
8.00
7.00
6.00
5.00
4.00
3.00
2.00
1.00
0.00
0.000
50.00
100.0
150.0
200.0
250.0
300.0
350.0
400.0
450.0

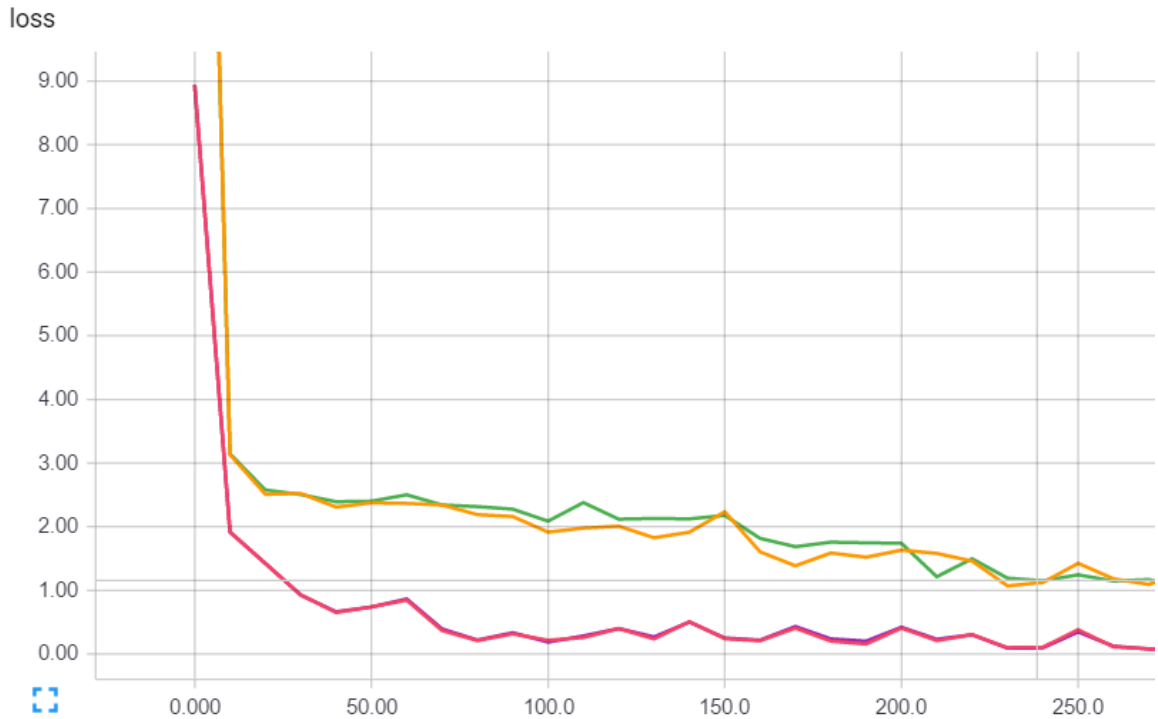

TensorBoard
EVENTS
IMAGES
GRAPH
HISTOGRAMS
Regex filter
Split on underscores
Horizontal Axis
STEP
RELATIVE
WALL
Runs
MNIST_Run1
MNIST_Run2
TOGGLE ALL RUNS
conv1_biases 2
conv1_biases
conv1_biases
0.200
0.100
0.00
-0.100
-0.200
-0.300
-0.400
0.000
2.000k
4.000k
6.000k
8.000k
conv1_weights 2
conv2_biases 2
conv2_weights 2
fc1_biases 2
fc1_weights 2
fc2_biases 2
fc2_weights 2


TensorBoard
EVENTS
IMAGES
GRAPH
HISTOGRAMS
Regex filter
Split on underscores
Data download links
Horizontal Axis
STEP
RELATIVE
WALL
Runs
MNIST_Run1
MNIST_Run2
TOGGLE ALL RUNS
learning_rate
1
learning_rate
9.500e-3
8.500e-3
7.500e-3
6.500e-3
0.000
2.000k
4.000k
6.000k
8.000k
loss
1
loss
8.00
6.00
4.00
2.00
0.00
0.000
2.000k
4.000k
6.000k
8.000k

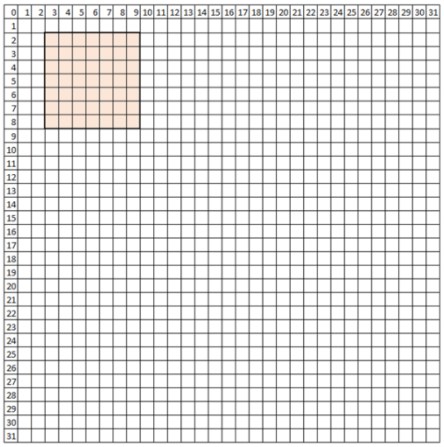

TensorBoard
EVENTS
IMAGES
GRAPH
HISTOGRAMS
Fit to screen
Download PNG
Run
MNIST_Run1
Upload
Choose File
Color
Structure
color: same substructure
gray: unique substructure
Graph
(* = expandable)
Namespace*
OpNode
Unconnected series*
Connected series*
Constant
Summary
Dataflow edge
Control dependency edge
Reference edge
Momentum
dropout
ExponentialDecay

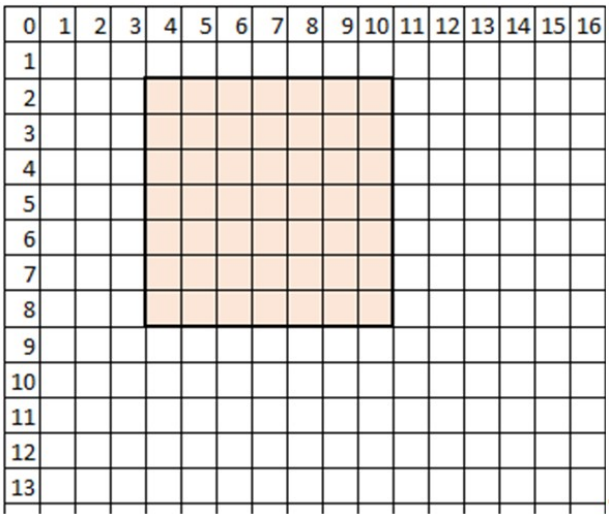

224
224
3
11
11
Stride
of 4
55
55
48
48
5
5
Max
pooling
27
27
128
128
3
3
Max
pooling
13
13
192
192
128
128
Max
pooling
2048
2048
2048
2048
dense
dense
dense
1000

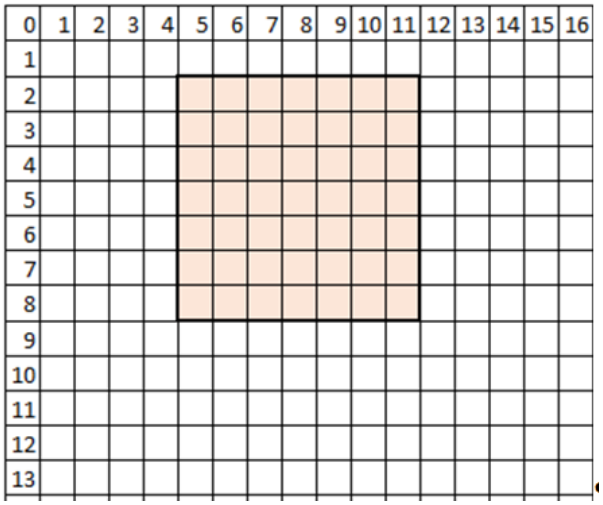

EVENTS
IMAGES
GRAPH
HISTOGRAMS
conv3
truncated_normal
weights
(conv3)
weights
BiasAd
biases
(weights...
read
Histogra...
Conv2D
(weights)
Assign
read
(weights...
truncate...
Assign
init
Conv2D
weights
biases
(biases)
Assign
read
Const
(biases)
Assign
init
read
(biases)
BiasAdd
pool2
conv2
init
pool1
Variable
conv1
init
conv3/weights
/(weights)
Operation: Variable
Attributes (4)
container {"s":""}
dtype {"type":"DT_FLOAT"}
shape {"shape":{"dim":[{"size":3},
{"size":3},{"size":192},
{"size":384}]}}
shared_nam {"s":""}
e
Inputs (0)
Outputs (2)
conv3/weights/Assign
conv3/weights/read
Add to main graph

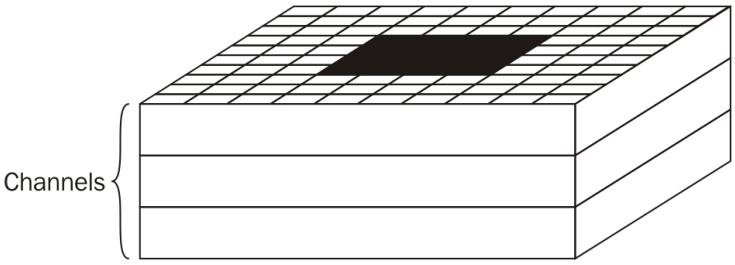


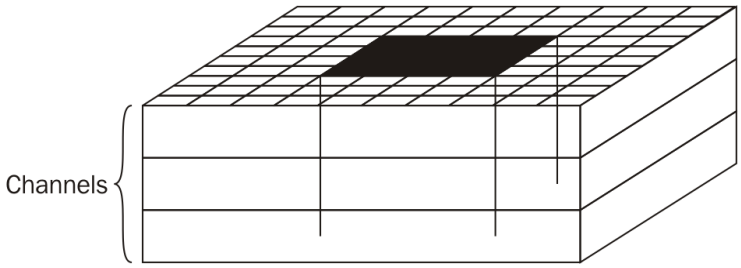
Chapter 04: Cats and Dogs

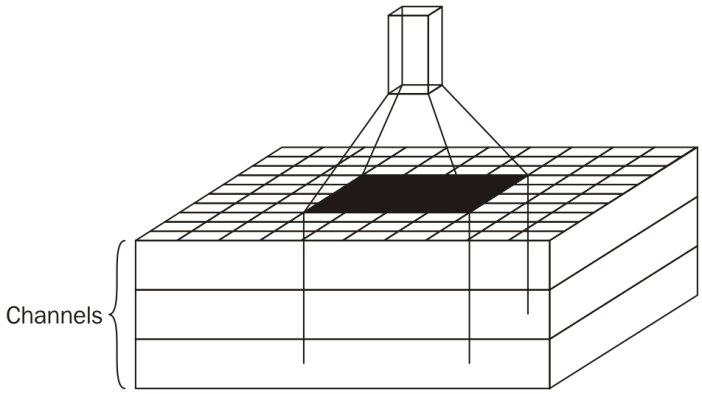


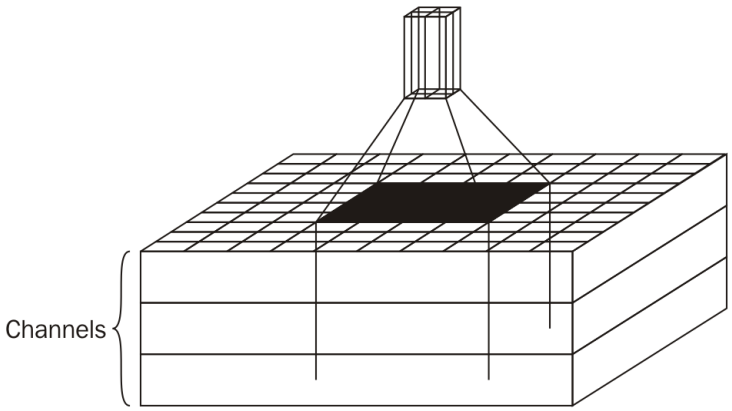

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
1
2
3
4
5
6
7
8
9
10
11
12
13


0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
1
2
3
4
5
6
7
8
9
10
11
12
13


Channels


Channels


Channels


Channels

	0	1	2	3	4	5	6	7
0	0.5	0.63	0.35	0.46	0.63	0.24	0.7	0.95
1	0.56	0.3	0.32	0.52	0.29	0.95	0.76	0.1
2	0.33	0.15	0.42	0.01	0.61	0.33	0.66	0.74
3	0.9	0.73	0.22	0.16	0.81	0.74	0.21	0.67
4	0.92	0.83	0.02	0.67	0.97	0.32	0.6	0.11
5	0.86	0.81	0.3	0.83	0.78	0.97	0.86	0.35
6	0.8	0.92	0.65	0.39	0.16	0.45	0.66	0.89
7	0.39	0.96	0.12	0.02	0.26	0.73	0.4	0.53

0.63	0.63	0.95

```python
def nn_model(data, weights, biases):
    layer_fc1 = tf.matmul(data, weights['fc1']) + biases['fc1']
    relu_layer = tf.nn.relu(layer_fc1)
    for relu in range(2, relu_layers + 1):
        relu_layer = tf.nn.relu(relu_layer)
    return tf.matmul(relu_layer, weights['fc2']) + biases['fc2']
```

S:\users\saif\appdata\local\temp\sublime-sftp-browse-146102303\1\DeepX_LOCO\home\saif\Documents\WORKSPACE\MLwithTF\book_code\chapter_04\ann_benchmark.py - Sublime Text

File Edit Selection Find View Goto Tools Project Preferences Help

ann_benchmark.py | data_utils.py | logmanager.py | nn_benchmark_with_tensorboard.py | untitled | nn_benchmark.py

```python
        output = math.ceil(output/2.0)
    return int(output)

def nn_model(data, weights, biases, TRAIN=False):
    with tf.name_scope('Layer_1') as scope:
        conv = tf.nn.conv2d(data, weights['conv1'], strides=[1, 1, 1, 1], padding='SAME', name='conv1')
        bias_add = tf.nn.bias_add(conv, biases['conv1'], name='bias_add_1')
        relu = tf.nn.relu(bias_add, name='relu_1')
        max_pool = tf.nn.max_pool(relu, ksize=[1, 2, 2, 1], strides=[1, 2, 2, 1], padding='SAME', name=scope)

    with tf.name_scope('Layer_2') as scope:
        conv = tf.nn.conv2d(max_pool, weights['conv2'], strides=[1, 1, 1, 1], padding='SAME', name='conv2')
        bias_add = tf.nn.bias_add(conv, biases['conv2'], name='bias_add_2')
        relu = tf.nn.relu(bias_add, name='relu_2')
        max_pool = tf.nn.max_pool(relu, ksize=[1, 2, 2, 1], strides=[1, 2, 2, 1], padding='SAME', name=scope)

    with tf.name_scope('Layer_3') as scope:
        conv = tf.nn.conv2d(max_pool, weights['conv3'], strides=[1, 1, 1, 1], padding='SAME', name='conv3')
        bias_add = tf.nn.bias_add(conv, biases['conv3'], name='bias_add_3')
        relu = tf.nn.relu(bias_add, name='relu_3')
        max_pool = tf.nn.max_pool(relu, ksize=[1, 2, 2, 1], strides=[1, 2, 2, 1], padding='SAME', name=scope)
        if TRAIN:
            max_pool = tf.nn.dropout(max_pool, dropout_prob, seed=SEED, name='dropout')

    shape = max_pool.get_shape().as_list()
    reshape = tf.reshape(max_pool, [shape[0], shape[1] * shape[2] * shape[3]])

    with tf.name_scope('FC_Layer_1') as scope:
        matmul = tf.matmul(reshape, weights['fc1'], name='fc1_matmul')
        bias_add = tf.nn.bias_add(matmul, biases['fc1'], name='fc1_bias_add')
        relu = tf.nn.relu(bias_add, name=scope)

    with tf.name_scope('FC_Layer_2') as scope:
        matmul = tf.matmul(relu, weights['fc2'], name='fc2_matmul')
        layer_fc2 = tf.nn.bias_add(matmul, biases['fc2'], name=scope)

    return layer_fc2
```

4 characters selected | Spaces: 4 | Python

```python
weights = {
    'conv1': tf.Variable(tf.truncated_normal(shape=[patch_size, patch_size, num_channels, depth_inc], dtype=tf.float32,
                                             stddev=stddev, seed=SEED), name='weights_conv1'),
    'conv2': tf.Variable(tf.truncated_normal([patch_size, patch_size, depth_inc, depth_inc], dtype=tf.float32,
                                             stddev=stddev, seed=SEED), name='weights_conv2'),
    'conv3': tf.Variable(tf.truncated_normal([patch_size, patch_size, depth_inc, depth_inc], dtype=tf.float32,
                                             stddev=stddev, seed=SEED), name='weights_conv3'),
    'fc1': tf.Variable(tf.truncated_normal([(fc_first_layer_dimen(image_size, conv_layers) ** 2) * depth_inc,
                                            num_hidden_inc], dtype=tf.float32,
                                           stddev=stddev, seed=SEED), name='weights_fc1'),
    'fc2': tf.Variable(tf.truncated_normal([num_hidden_inc, num_of_classes], dtype=tf.float32,
                                           stddev=stddev, seed=SEED), name='weights_fc2')
}
biases = {
    'conv1': tf.Variable(tf.zeros(shape=[depth_inc], dtype=tf.float32), name='biases_conv1'),
    'conv2': tf.Variable(tf.zeros(shape=[depth_inc], dtype=tf.float32), name='biases_conv2'),
    'conv3': tf.Variable(tf.zeros(shape=[depth_inc], dtype=tf.float32), name='biases_conv3'),
    'fc1': tf.Variable(tf.zeros(shape=[num_hidden_inc], dtype=tf.float32), name='biases_fc1'),
    'fc2': tf.Variable(tf.zeros(shape=[num_of_classes], dtype=tf.float32), name='biases_fc2'),
}
```

```python
with tf.Session(graph=graph) as session:
    writer = tf.summary.FileWriter(log_location, session.graph)
    merged = tf.summary.merge_all()
    tf.global_variables_initializer().run()
    print("Initialized")
    for step in range(num_steps + 1):
        sys.stdout.write('Training on batch %d of %d\r' % (step + 1, num_steps))
        sys.stdout.flush()
        # Pick an offset within the training data, which has been randomized.
        # Note: we could use better randomization across epochs.
        offset = (step * batch_size) % (dataset.train_labels.shape[0] - batch_size)
        # Generate a minibatch.
        batch_data = dataset.train_dataset[offset:(offset + batch_size), :]
        batch_labels = dataset.train_labels[offset:(offset + batch_size), :]
        # Prepare a dictionary telling the session where to feed the minibatch.
        # The key of the dictionary is the placeholder node of the graph to be fed,
        # and the value is the numpy array to feed to it.
        feed_dict = {tf_train_dataset: batch_data, tf_train_labels: batch_labels}
        summary_result, _, l, predictions = session.run([merged, optimizer, loss, train_prediction], feed_dict=feed_dict)
        writer.add_summary(summary_result, step)

        if step % data_showing_step == 0:
            acc_minibatch = accuracy(predictions, batch_labels)
            acc_val = accuracy(valid_prediction.eval(), dataset.valid_labels)
            logmanager.logger.info('# %03d  Acc Train: %03.2f%%  Acc Val: %03.2f%% Loss %f' % (
                step, acc_minibatch, acc_val, l))

    logmanager.logger.info("Test accuracy: %.1f%%" % accuracy(test_prediction.eval(), dataset.test_labels))
```

```
2017-09-29 03:32:05,861 - MLwithTF - INFO - # 19500  Acc Train: 90.62%  Acc Val: 87.45% Loss 0.319119
2017-09-29 03:32:06,640 - MLwithTF - INFO - # 20000  Acc Train: 93.75%  Acc Val: 87.41% Loss 0.305260
2017-09-29 03:32:07,419 - MLwithTF - INFO - # 20500  Acc Train: 90.62%  Acc Val: 86.87% Loss 0.331279
2017-09-29 03:32:08,204 - MLwithTF - INFO - # 21000  Acc Train: 84.38%  Acc Val: 87.38% Loss 0.532096
2017-09-29 03:32:08,986 - MLwithTF - INFO - # 21500  Acc Train: 87.50%  Acc Val: 86.88% Loss 0.557634
2017-09-29 03:32:09,766 - MLwithTF - INFO - # 22000  Acc Train: 84.38%  Acc Val: 87.15% Loss 0.726978
2017-09-29 03:32:10,549 - MLwithTF - INFO - # 22500  Acc Train: 81.25%  Acc Val: 86.67% Loss 0.871303
2017-09-29 03:32:11,323 - MLwithTF - INFO - # 23000  Acc Train: 81.25%  Acc Val: 87.54% Loss 0.698311
2017-09-29 03:32:12,104 - MLwithTF - INFO - # 23500  Acc Train: 81.25%  Acc Val: 87.44% Loss 0.543187
2017-09-29 03:32:12,883 - MLwithTF - INFO - # 24000  Acc Train: 87.50%  Acc Val: 87.81% Loss 0.501370
2017-09-29 03:32:13,661 - MLwithTF - INFO - # 24500  Acc Train: 93.75%  Acc Val: 87.22% Loss 0.329258
2017-09-29 03:32:14,441 - MLwithTF - INFO - # 25000  Acc Train: 90.62%  Acc Val: 87.72% Loss 0.281238
2017-09-29 03:32:15,223 - MLwithTF - INFO - # 25500  Acc Train: 78.12%  Acc Val: 87.40% Loss 0.863225
2017-09-29 03:32:16,003 - MLwithTF - INFO - # 26000  Acc Train: 90.62%  Acc Val: 87.01% Loss 0.585005
2017-09-29 03:32:16,782 - MLwithTF - INFO - # 26500  Acc Train: 93.75%  Acc Val: 87.41% Loss 0.243985
2017-09-29 03:32:17,563 - MLwithTF - INFO - # 27000  Acc Train: 96.88%  Acc Val: 87.24% Loss 0.258554
2017-09-29 03:32:18,340 - MLwithTF - INFO - # 27500  Acc Train: 84.38%  Acc Val: 87.59% Loss 0.757773
2017-09-29 03:32:19,118 - MLwithTF - INFO - # 28000  Acc Train: 84.38%  Acc Val: 87.34% Loss 0.543425
2017-09-29 03:32:19,903 - MLwithTF - INFO - # 28500  Acc Train: 93.75%  Acc Val: 87.57% Loss 0.428805
2017-09-29 03:32:20,690 - MLwithTF - INFO - # 29000  Acc Train: 87.50%  Acc Val: 87.22% Loss 0.393010
2017-09-29 03:32:21,479 - MLwithTF - INFO - # 29500  Acc Train: 84.38%  Acc Val: 87.82% Loss 0.402495
2017-09-29 03:32:22,248 - MLwithTF - INFO - # 30000  Acc Train: 90.62%  Acc Val: 87.75% Loss 0.379222
2017-09-29 03:32:22,288 - MLwithTF - INFO - Test accuracy: 93.5%
(work2) ubuntu@ubuntu-PC:~/github/mlwithtf/chapter_04$
```

S:\users\saif\appdata\local\temp\sublime-sftp-browse-146112878?\DeepX_LOCO\home\saif\Documents\WORKSPACE\MLwithTF\book_code\data_utils.py • - Sublime Text

File Edit Selection Find View Goto Tools Project Preferences Help

untitled | ann_benchmark.py | data_utils.py

```python


def load_cifar_10_pickle(pickle_file, image_depth):
    fo = open(pickle_file, 'rb')
    dict = pickle.load(fo)
    fo.close()
    return ((dict['data'].astype(float) - image_depth / 2) / (image_depth)), dict['labels']


def load_cifar_10_from_pickles(train_pickle_files, test_pickle_files, pickle_batch_size, image_size, image_depth,
                               num_of_channels):
    all_train_data = np.ndarray(shape=(pickle_batch_size * len(train_pickle_files),
                                       image_size * image_size * num_of_channels),
                                dtype=np.float32)

    all_train_labels = np.ndarray(shape=pickle_batch_size * len(train_pickle_files), dtype=object)

    all_test_data = np.ndarray(shape=(pickle_batch_size * len(test_pickle_files),
                                      image_size * image_size * num_of_channels),
                               dtype=np.float32)
    all_test_labels = np.ndarray(shape=pickle_batch_size * len(test_pickle_files), dtype=object)

    print('Started loading training data')
    for index, train_pickle_file in enumerate(train_pickle_files):
        all_train_data[index * pickle_batch_size: (index + 1) * pickle_batch_size, :], \
        all_train_labels[index * pickle_batch_size: (index + 1) * pickle_batch_size] = \
            load_cifar_10_pickle(train_pickle_file, image_depth)
    print('Finished loading training data\n')

    print('Started loading testing data')
    for index, test_pickle_file in enumerate(test_pickle_files):
        all_test_data[index * pickle_batch_size: (index + 1) * pickle_batch_size, :], \
        all_test_labels[index * pickle_batch_size: (index + 1) * pickle_batch_size] = \
            load_cifar_10_pickle(test_pickle_file, image_depth)
    print('Finished loading testing data')

    return all_train_data, all_train_labels, all_test_data, all_test_labels

```

load_cifar_10_pickle

Find | Find Prev | Find All

Line 244, Column 64 | Spaces: 4 | Python

```python
def pickle_cifar_10(all_train_data, all_train_labels, all_test_data, all_test_labels,
                    train_size, valid_size, test_size, output_file_path, FORCE=False):

    if os.path.isfile(output_file_path) and not FORCE:
        print('Pickle file: %s already exist' % output_file_path)

        with open(output_file_path, 'rb') as f:
            save = pickle.load(f)
            train_dataset = save['train_dataset']
            train_labels = save['train_labels']
            valid_dataset = save['valid_dataset']
            valid_labels = save['valid_labels']
            test_dataset = save['test_dataset']
            test_labels = save['test_labels']
            del save  # hint to help gc free up memory
            print('Training set', train_dataset.shape, train_labels.shape)
            print('Validation set', valid_dataset.shape, valid_labels.shape)
            print('Test set', test_dataset.shape, test_labels.shape)

        return train_dataset, train_labels, valid_dataset, valid_labels, test_dataset, test_labels
    else:
        train_dataset = all_train_data[0:train_size]
        train_labels = all_train_labels[0:train_size]
        valid_dataset = all_train_data[train_size:train_size + valid_size]
        valid_labels = all_train_labels[train_size:train_size + valid_size]
        test_dataset = all_test_data[0:test_size]
        test_labels = all_test_labels[0:test_size]

        try:
            f = open(output_file_path, 'wb')
            save = {
                'train_dataset': train_dataset,
                'train_labels': train_labels,
                'valid_dataset': valid_dataset,
                'valid_labels': valid_labels,
                'test_dataset': test_dataset,
                'test_labels': test_labels,
            }
            pickle.dump(save, f, pickle.HIGHEST_PROTOCOL)
            f.close()
        except Exception as e:
            print('Unable to save data to', output_file_path, ':', e)
            raise

        statinfo = os.stat(output_file_path)
        print('Compressed pickle size:', statinfo.st_size)

    return train_dataset, train_labels, valid_dataset, valid_labels, test_dataset, test_labels
```

```
if (evaluateFile is not None):
    image = (ndimage.imread(evaluateFile).astype(float) - 255 / 2) / 255
    image = image.reshape((image_size, image_size, num_channels)).astype(np.float32)
    random_data = np.ndarray((1, image_size, image_size, num_channels), dtype=np.float32)
    random_data[0, :, :, :] = image

    feed_dict = {tf_random_dataset: random_data}
    output = session.run(
        [random_prediction], feed_dict=feed_dict)

    for i, smx in enumerate(output):
        prediction = smx[0].argmax(axis=0)
        print 'The prediction is: %d' % (prediction)
```

Chapter 05: Sequence to Sequence Models-Parlez-vous?

```
def prepare_wmt_dataset(tokenizer=None):
    vocab_size = 40000

    # URLs for WMT data.
    _WMT_ENFR_TRAIN_URL = "http://www.statmt.org/wmt10/training-giga-fren.tar"
    _WMT_ENFR_DEV_URL = "http://www.statmt.org/wmt15/dev-v2.tgz"

    # Expected number of bytes for the above two file downloads
    _WMT_ENFR_TRAIN_SIZE = 2595102720
    _WMT_ENFR_DEV_SIZE = 21393583

    train_file_path = download_file(_WMT_ENFR_TRAIN_URL,
                                    os.path.realpath('../datasets/WMT'), _WMT_ENFR_TRAIN_SIZE)

    dev_file_path = download_file(_WMT_ENFR_DEV_URL,
                                  os.path.realpath('../datasets/WMT'), _WMT_ENFR_DEV_SIZE)

    train_extracted_folder = extract_file(train_file_path, os.path.realpath('../datasets/WMT/train'), IS_SUB=False)
    dev_extracted_folder = extract_file(dev_file_path, os.path.realpath('../datasets/WMT'), IS_SUB=False,
                                        FORCE=True)

    train_sub_gzip_files = ['giga-fren.release2.fixed.fr.gz', 'giga-fren.release2.fixed.en.gz']
    train_sub_gzip_files = [train_extracted_folder + '/' + x for x in train_sub_gzip_files]

    if not os.path.exists(train_extracted_folder + '/data'):
        os.makedirs(train_extracted_folder + '/data')

    train_sub_extracted_files = [None] * 2
    for index, train_sub_gzip_file in enumerate(train_sub_gzip_files):
        train_sub_extracted_files[index] = extract_file(train_sub_gzip_file,
                                                        train_extracted_folder + '/data/' +
                                                        train_sub_gzip_file.split('/')[-1].split('.gz')[0],
                                                        TYPE='gz',
                                                        IS_SUB=False)
```

```python
vocab_paths = [None] * 2
token_paths = [None] * 2

# Create vocabularies of the appropriate sizes and tokenizing the vocabulary
for index, train_sub_extracted_file in enumerate(train_sub_extracted_files):
    type = train_sub_extracted_file.split('.')[-1]
    vocab_paths[index] = "%s%s.%s" % (train_extracted_folder + '/data/', 'vocab%d' % vocab_size, type)
    token_paths[index] = "%s%s.%s" % (train_extracted_folder + '/data/', 'token%d' % vocab_size, type)
    create_vocabulary(vocab_paths[index], train_sub_extracted_files[index], vocab_size, tokenizer)
    data_to_token_ids(train_sub_extracted_files[index], token_paths[index], vocab_paths[index], tokenizer)

dev_sub_required_files = ['dev/newstest2013.fr', 'dev/newstest2013.en']
dev_sub_required_files = [dev_extracted_folder + '/' + x for x in dev_sub_required_files]

if not os.path.exists(dev_extracted_folder + '/dev/data'):
    os.makedirs(dev_extracted_folder + '/dev/data')

dev_token_paths = [None] * 2
for index, dev_sub_required_file in enumerate(dev_sub_required_files):
    type = dev_sub_required_file.split('.')[-1]
    dev_token_paths[index] = "%s%s.%s" % (dev_extracted_folder + '/dev/data/', 'token%d' % vocab_size, type)
    data_to_token_ids(dev_sub_required_files[index], dev_token_paths[index], vocab_paths[index], tokenizer)

def wmt(): pass
```

```python
def create_vocabulary(vocabulary_path, data_path, max_vocabulary_size,
                      tokenizer=None, normalize_digits=True):
    if not gfile.Exists(vocabulary_path):
        print("Creating vocabulary %s from data %s" % (vocabulary_path, data_path))
        vocab = {}
        with gfile.GFile(data_path, mode="rb") as f:
            counter = 0
            for line in f:
                counter += 1
                if counter % 100000 == 0:
                    print("  processing line %d" % counter)
                tokens = tokenizer(line) if tokenizer else basic_tokenizer(line)
                for w in tokens:
                    word = re.sub(_DIGIT_RE, b"0", w) if normalize_digits else w
                    if word in vocab:
                        vocab[word] += 1
                    else:
                        vocab[word] = 1
            vocab_list = _START_VOCAB + sorted(vocab, key=vocab.get, reverse=True)
            if len(vocab_list) > max_vocabulary_size:
                vocab_list = vocab_list[:max_vocabulary_size]
            with gfile.GFile(vocabulary_path, mode="wb") as vocab_file:
                for w in vocab_list:
                    vocab_file.write(w + b"\n")
```

```python
def sentence_to_token_ids(sentence, vocabulary,
                          tokenizer=None, normalize_digits=True):
    if tokenizer:
        words = tokenizer(sentence)
    else:
        words = basic_tokenizer(sentence)
    if not normalize_digits:
        return [vocabulary.get(w, UNK_ID) for w in words]
    # Normalize digits by 0 before looking words up in the vocabulary.
    return [vocabulary.get(re.sub(_DIGIT_RE, b"0", w), UNK_ID) for w in words]
```

```python
def data_to_token_ids(data_path, target_path, vocabulary_path,
                      tokenizer=None, normalize_digits=True):

    if not gfile.Exists(target_path):
        print("Tokenizing data in %s" % data_path)
        vocab, _ = initialize_vocabulary(vocabulary_path)
        with gfile.GFile(data_path, mode="rb") as data_file:
            with gfile.GFile(target_path, mode="w") as tokens_file:
                counter = 0
                for line in data_file:
                    counter += 1
                    if counter % 100000 == 0:
                        print("  tokenizing line %d" % counter)
                    token_ids = sentence_to_token_ids(line, vocab, tokenizer,
                                                      normalize_digits)
                    tokens_file.write(" ".join([str(tok) for tok in token_ids]) + "\n")
```

```python
def train():
  wmt = data_utils.prepare_wmt_dataset()
  #en_train, fr_train, en_dev, fr_dev, _, _ = data_utils.prepare_wmt_dataset()

  with tf.Session() as sess:
    # Create model.
    print("Creating %d layers of %d units." % (FLAGS.num_layers, FLAGS.size))
    model = create_model(sess, False)

    # Read data into buckets and compute their sizes.
    print ("Reading development and training data (limit: %d)."
           % FLAGS.max_train_data_size)
    dev_set = read_data(wmt.en_dev_ids_path, wmt.fr_dev_ids_path)
    train_set = read_data(wmt.en_train_ids_path, wmt.fr_train_ids_path, FLAGS.max_train_data_size)
    train_bucket_sizes = [len(train_set[b]) for b in xrange(len(_buckets))]
    train_total_size = float(sum(train_bucket_sizes))

    # A bucket scale is a list of increasing numbers from 0 to 1 that we'll use
    # to select a bucket. Length of [scale[i], scale[i+1]] is proportional to
    # the size if i-th training bucket, as used later.
    train_buckets_scale = [sum(train_bucket_sizes[:i + 1]) / train_total_size
                           for i in xrange(len(train_bucket_sizes))]
```

```python
step_time, loss = 0.0, 0.0
current_step = 0
previous_losses = []
while True:
  # Choose a bucket according to data distribution. We pick a random number
  # in [0, 1] and use the corresponding interval in train_buckets_scale.
  random_number_01 = np.random.random_sample()
  bucket_id = min([i for i in xrange(len(train_buckets_scale))
                   if train_buckets_scale[i] > random_number_01])

  # Get a batch and make a step.
  start_time = time.time()
  encoder_inputs, decoder_inputs, target_weights = model.get_batch(
      train_set, bucket_id)
  _, step_loss, _ = model.step(sess, encoder_inputs, decoder_inputs,
                               target_weights, bucket_id, False)
  step_time += (time.time() - start_time) / FLAGS.steps_per_checkpoint
  loss += step_loss / FLAGS.steps_per_checkpoint
  current_step += 1

  # Once in a while, we save checkpoint, print statistics, and run evals.
  if current_step % FLAGS.steps_per_checkpoint == 0:
    # Print statistics for the previous epoch.
    perplexity = math.exp(loss) if loss < 300 else float('inf')
    print ("global step %d learning rate %.4f step-time %.2f perplexity "
           "%.2f" % (model.global_step.eval(), model.learning_rate.eval(),
                     step_time, perplexity))
    # Decrease learning rate if no improvement was seen over last 3 times.
    if len(previous_losses) > 2 and loss > max(previous_losses[-3:]):
      sess.run(model.learning_rate_decay_op)
    previous_losses.append(loss)
    # Save checkpoint and zero timer and loss.
    checkpoint_path = os.path.join(FLAGS.train_dir, "translate.ckpt")
    model.saver.save(sess, checkpoint_path, global_step=model.global_step)
    step_time, loss = 0.0, 0.0
```

```python
    # Run evals on development set and print their perplexity.
    for bucket_id in xrange(len(_buckets)):
      if len(dev_set[bucket_id]) == 0:
        print("  eval: empty bucket %d" % (bucket_id))
        continue
      encoder_inputs, decoder_inputs, target_weights = model.get_batch(
          dev_set, bucket_id)
      _, eval_loss, _ = model.step(sess, encoder_inputs, decoder_inputs,
                                   target_weights, bucket_id, True)
      eval_ppx = math.exp(eval_loss) if eval_loss < 300 else float('inf')
      print("  eval: bucket %d perplexity %.2f" % (bucket_id, eval_ppx))
    sys.stdout.flush()
```

```python
class Seq2SeqModel(object):
  def __init__(self, source_vocab_size, target_vocab_size, buckets, size,
               num_layers, max_gradient_norm, batch_size, learning_rate,
               learning_rate_decay_factor, use_lstm=False,
               num_samples=512, forward_only=False):

    self.source_vocab_size = source_vocab_size
    self.target_vocab_size = target_vocab_size
    self.buckets = buckets
    self.batch_size = batch_size
    self.learning_rate = tf.Variable(float(learning_rate), trainable=False)
    self.learning_rate_decay_op = self.learning_rate.assign(
        self.learning_rate * learning_rate_decay_factor)
    self.global_step = tf.Variable(0, trainable=False)

    # If we use sampled softmax, we need an output projection.
    output_projection = None
    softmax_loss_function = None
    # Sampled softmax only makes sense if we sample less than vocabulary size.
    if num_samples > 0 and num_samples < self.target_vocab_size:
      with tf.device("/cpu:0"):
        w = tf.get_variable("proj_w", [size, self.target_vocab_size])
        w_t = tf.transpose(w)
        b = tf.get_variable("proj_b", [self.target_vocab_size])
      output_projection = (w, b)

      def sampled_loss(inputs, labels):
        with tf.device("/cpu:0"):
          labels = tf.reshape(labels, [-1, 1])
          return tf.nn.sampled_softmax_loss(w_t, b, inputs, labels, num_samples,
                                            self.target_vocab_size)
      softmax_loss_function = sampled_loss

    # Create the internal multi-layer cell for our RNN.
    single_cell = tf.nn.rnn_cell.GRUCell(size)
    if use_lstm:
      single_cell = tf.nn.rnn_cell.BasicLSTMCell(size)
    cell = single_cell
    if num_layers > 1:
      cell = tf.nn.rnn_cell.MultiRNNCell([single_cell] * num_layers)

    # The seq2seq function: we use embedding for the input and attention.
    def seq2seq_f(encoder_inputs, decoder_inputs, do_decode):
      return tf.nn.seq2seq.embedding_attention_seq2seq(
          encoder_inputs, decoder_inputs, cell,
          num_encoder_symbols=source_vocab_size,
          num_decoder_symbols=target_vocab_size,
          embedding_size=size,
          output_projection=output_projection,
          feed_previous=do_decode)
```

```
# Feeds for inputs.
self.encoder_inputs = []
self.decoder_inputs = []
self.target_weights = []
for i in xrange(buckets[-1][0]):  # Last bucket is the biggest one.
  self.encoder_inputs.append(tf.placeholder(tf.int32, shape=[None],
                                            name="encoder{0}".format(i)))
for i in xrange(buckets[-1][1] + 1):
  self.decoder_inputs.append(tf.placeholder(tf.int32, shape=[None],
                                            name="decoder{0}".format(i)))
  self.target_weights.append(tf.placeholder(tf.float32, shape=[None],
                                            name="weight{0}".format(i)))

# Our targets are decoder inputs shifted by one.
targets = [self.decoder_inputs[i + 1]
           for i in xrange(len(self.decoder_inputs) - 1)]

# Training outputs and losses.
if forward_only:
  self.outputs, self.losses = tf.nn.seq2seq.model_with_buckets(
      self.encoder_inputs, self.decoder_inputs, targets,
      self.target_weights, buckets, lambda x, y: seq2seq_f(x, y, True),
      softmax_loss_function=softmax_loss_function)
  # If we use output projection, we need to project outputs for decoding.
  if output_projection is not None:
    for b in xrange(len(buckets)):
      self.outputs[b] = [
          tf.matmul(output, output_projection[0]) + output_projection[1]
          for output in self.outputs[b]
      ]
else:
  self.outputs, self.losses = tf.nn.seq2seq.model_with_buckets(
      self.encoder_inputs, self.decoder_inputs, targets,
      self.target_weights, buckets,
      lambda x, y: seq2seq_f(x, y, False),
      softmax_loss_function=softmax_loss_function)

# Gradients and SGD update operation for training the model.
params = tf.trainable_variables()
if not forward_only:
  self.gradient_norms = []
  self.updates = []
  opt = tf.train.GradientDescentOptimizer(self.learning_rate)
  for b in xrange(len(buckets)):
    gradients = tf.gradients(self.losses[b], params)
    clipped_gradients, norm = tf.clip_by_global_norm(gradients,
                                                     max_gradient_norm)
    self.gradient_norms.append(norm)
    self.updates.append(opt.apply_gradients(
        zip(clipped_gradients, params), global_step=self.global_step))

self.saver = tf.train.Saver(tf.all_variables())
```

```python
def step(self, session, encoder_inputs, decoder_inputs, target_weights,
         bucket_id, forward_only):

  encoder_size, decoder_size = self.buckets[bucket_id]
  if len(encoder_inputs) != encoder_size:
    raise ValueError("Encoder length must be equal to the one in bucket,"
                     " %d != %d." % (len(encoder_inputs), encoder_size))
  if len(decoder_inputs) != decoder_size:
    raise ValueError("Decoder length must be equal to the one in bucket,"
                     " %d != %d." % (len(decoder_inputs), decoder_size))
  if len(target_weights) != decoder_size:
    raise ValueError("Weights length must be equal to the one in bucket,"
                     " %d != %d." % (len(target_weights), decoder_size))

  # Input feed: encoder inputs, decoder inputs, target_weights, as provided.
  input_feed = {}
  for l in xrange(encoder_size):
    input_feed[self.encoder_inputs[l].name] = encoder_inputs[l]
  for l in xrange(decoder_size):
    input_feed[self.decoder_inputs[l].name] = decoder_inputs[l]
    input_feed[self.target_weights[l].name] = target_weights[l]

  # Since our targets are decoder inputs shifted by one, we need one more.
  last_target = self.decoder_inputs[decoder_size].name
  input_feed[last_target] = np.zeros([self.batch_size], dtype=np.int32)

  # Output feed: depends on whether we do a backward step or not.
  if not forward_only:
    output_feed = [self.updates[bucket_id],  # Update Op that does SGD.
                   self.gradient_norms[bucket_id],  # Gradient norm.
                   self.losses[bucket_id]]  # Loss for this batch.
  else:
    output_feed = [self.losses[bucket_id]]  # Loss for this batch.
    for l in xrange(decoder_size):  # Output logits.
      output_feed.append(self.outputs[bucket_id][l])

  outputs = session.run(output_feed, input_feed)
  if not forward_only:
    return outputs[1], outputs[2], None  # Gradient norm, loss, no outputs.
  else:
    return None, outputs[0], outputs[1:]  # No gradient norm, loss, outputs.
```

```python
def get_batch(self, data, bucket_id):

  encoder_size, decoder_size = self.buckets[bucket_id]
  encoder_inputs, decoder_inputs = [], []

  # Get a random batch of encoder and decoder inputs from data,
  # pad them if needed, reverse encoder inputs and add GO to decoder.
  for _ in xrange(self.batch_size):
    encoder_input, decoder_input = random.choice(data[bucket_id])

    # Encoder inputs are padded and then reversed.
    encoder_pad = [data_utils.PAD_ID] * (encoder_size - len(encoder_input))
    encoder_inputs.append(list(reversed(encoder_input + encoder_pad)))

    # Decoder inputs get an extra "GO" symbol, and are padded then.
    decoder_pad_size = decoder_size - len(decoder_input) - 1
    decoder_inputs.append([data_utils.GO_ID] + decoder_input +
                          [data_utils.PAD_ID] * decoder_pad_size)

  # Now we create batch-major vectors from the data selected above.
  batch_encoder_inputs, batch_decoder_inputs, batch_weights = [], [], []

  # Batch encoder inputs are just re-indexed encoder_inputs.
  for length_idx in xrange(encoder_size):
    batch_encoder_inputs.append(
        np.array([encoder_inputs[batch_idx][length_idx]
                  for batch_idx in xrange(self.batch_size)], dtype=np.int32))

  # Batch decoder inputs are re-indexed decoder_inputs, we create weights.
  for length_idx in xrange(decoder_size):
    batch_decoder_inputs.append(
        np.array([decoder_inputs[batch_idx][length_idx]
                  for batch_idx in xrange(self.batch_size)], dtype=np.int32))

    # Create target_weights to be 0 for targets that are padding.
    batch_weight = np.ones(self.batch_size, dtype=np.float32)
    for batch_idx in xrange(self.batch_size):
      # We set weight to 0 if the corresponding target is a PAD symbol.
      # The corresponding target is decoder_input shifted by 1 forward.
      if length_idx < decoder_size - 1:
        target = decoder_inputs[batch_idx][length_idx + 1]
      if length_idx == decoder_size - 1 or target == data_utils.PAD_ID:
        batch_weight[batch_idx] = 0.0
    batch_weights.append(batch_weight)
  return batch_encoder_inputs, batch_decoder_inputs, batch_weights
```

Chapter 06: Finding Meaning

```
import tensorflow as tf
import numpy as np
import os
import time
import datetime
import data_helpers
from text_cnn import TextCNN

# Parameters
# ==================================================

# Model Hyperparameters
tf.flags.DEFINE_integer("embedding_dim", 128, "Dimensionality of character embedding (default: 128)")
tf.flags.DEFINE_string("filter_sizes", "3,4,5", "Comma-separated filter sizes (default: '3,4,5')")
tf.flags.DEFINE_integer("num_filters", 128, "Number of filters per filter size (default: 128)")
tf.flags.DEFINE_float("dropout_keep_prob", 0.5, "Dropout keep probability (default: 0.5)")
tf.flags.DEFINE_float("l2_reg_lambda", 0.0, "L2 regularizaion lambda (default: 0.0)")

# Training parameters
tf.flags.DEFINE_integer("batch_size", 64, "Batch Size (default: 64)")
tf.flags.DEFINE_integer("num_epochs", 200, "Number of training epochs (default: 200)")
tf.flags.DEFINE_integer("evaluate_every", 100, "Evaluate model on dev set after this many steps (default: 100)")
tf.flags.DEFINE_integer("checkpoint_every", 100, "Save model after this many steps (default: 100)")
# Misc Parameters
tf.flags.DEFINE_boolean("allow_soft_placement", True, "Allow device soft device placement")
tf.flags.DEFINE_boolean("log_device_placement", False, "Log placement of ops on devices")

FLAGS = tf.flags.FLAGS
FLAGS._parse_flags()
print("\nParameters:")
for attr, value in sorted(FLAGS.__flags.items()):
    print("{}={}".format(attr.upper(), value))
print("")
```

```
# Data Preparation
# ==================================================

# Load data
print("Loading data...")
x_text, y = data_helpers.load_data_and_labels(FLAGS.positive_data_file, FLAGS.negative_data_file)

# Build vocabulary
max_document_length = max([len(x.split(" ")) for x in x_text])
vocab_processor = learn.preprocessing.VocabularyProcessor(max_document_length)
x = np.array(list(vocab_processor.fit_transform(x_text)))

# Randomly shuffle data
np.random.seed(10)
shuffle_indices = np.random.permutation(np.arange(len(y)))
x_shuffled = x[shuffle_indices]
y_shuffled = y[shuffle_indices]

# Split train/test set
# TODO: This is very crude, should use cross-validation
dev_sample_index = -1 * int(FLAGS.dev_sample_percentage * float(len(y)))
x_train, x_dev = x_shuffled[:dev_sample_index], x_shuffled[dev_sample_index:]
y_train, y_dev = y_shuffled[:dev_sample_index], y_shuffled[dev_sample_index:]
print("Vocabulary Size: {:d}".format(len(vocab_processor.vocabulary_)))
print("Train/Dev split: {:d}/{:d}".format(len(y_train), len(y_dev)))
```

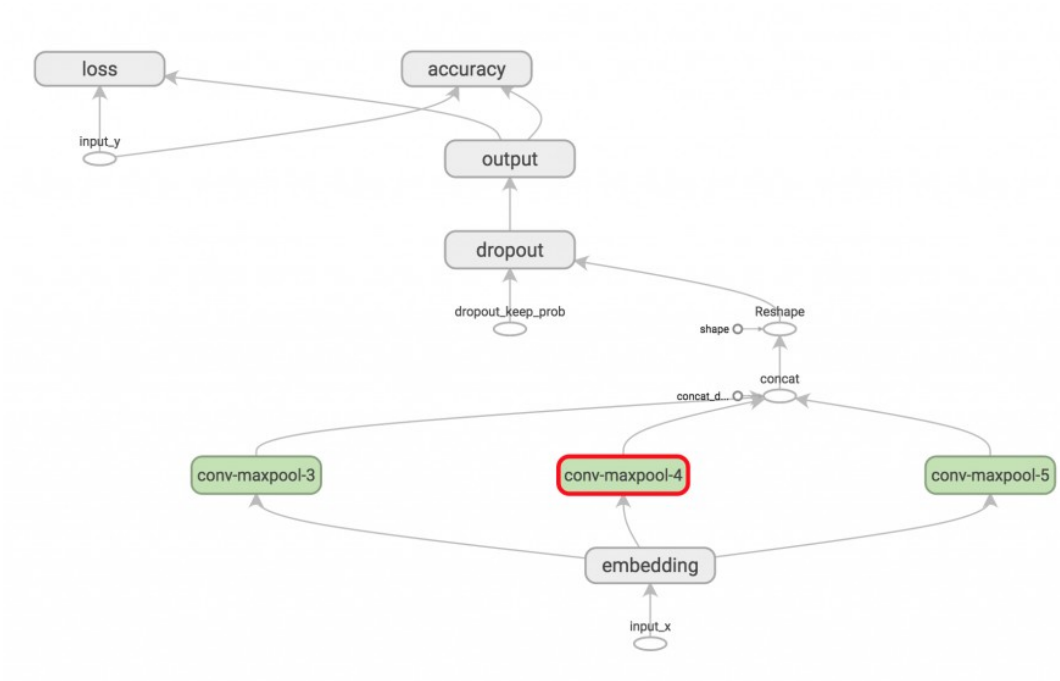
```python
# Training
# ==================================================

with tf.Graph().as_default():
    session_conf = tf.ConfigProto(
      allow_soft_placement=FLAGS.allow_soft_placement,
      log_device_placement=FLAGS.log_device_placement)
    sess = tf.Session(config=session_conf)
    with sess.as_default():
        cnn = TextCNN(
            sequence_length=x_train.shape[1],
            num_classes=2,
            vocab_size=len(vocabulary),
            embedding_size=FLAGS.embedding_dim,
            filter_sizes=list(map(int, FLAGS.filter_sizes.split(","))),
            num_filters=FLAGS.num_filters,
            l2_reg_lambda=FLAGS.l2_reg_lambda)

        # Define Training procedure
        global_step = tf.Variable(0, name="global_step", trainable=False)
        optimizer = tf.train.AdamOptimizer(1e-3)
        grads_and_vars = optimizer.compute_gradients(cnn.loss)
        train_op = optimizer.apply_gradients(grads_and_vars, global_step=global_step)

        # Keep track of gradient values and sparsity (optional)
        grad_summaries = []
        for g, v in grads_and_vars:
            if g is not None:
                grad_hist_summary = tf.histogram_summary("{}/grad/hist".format(v.name), g)
                sparsity_summary = tf.scalar_summary("{}/grad/sparsity".format(v.name), tf.nn.zero_fraction(g))
                grad_summaries.append(grad_hist_summary)
                grad_summaries.append(sparsity_summary)
        grad_summaries_merged = tf.merge_summary(grad_summaries)

        # Output directory for models and summaries
        timestamp = str(int(time.time()))
        out_dir = os.path.abspath(os.path.join(os.path.curdir, "runs", timestamp))
        print("Writing to {}\n".format(out_dir))
```

```python
# Summaries for loss and accuracy
loss_summary = tf.scalar_summary("loss", cnn.loss)
acc_summary = tf.scalar_summary("accuracy", cnn.accuracy)

# Train Summaries
train_summary_op = tf.merge_summary([loss_summary, acc_summary, grad_summaries_merged])
train_summary_dir = os.path.join(out_dir, "summaries", "train")
train_summary_writer = tf.train.SummaryWriter(train_summary_dir, sess.graph_def)

# Dev summaries
dev_summary_op = tf.merge_summary([loss_summary, acc_summary])
dev_summary_dir = os.path.join(out_dir, "summaries", "dev")
dev_summary_writer = tf.train.SummaryWriter(dev_summary_dir, sess.graph_def)

# Checkpoint directory. Tensorflow assumes this directory already exists so we need to create it
checkpoint_dir = os.path.abspath(os.path.join(out_dir, "checkpoints"))
checkpoint_prefix = os.path.join(checkpoint_dir, "model")
if not os.path.exists(checkpoint_dir):
    os.makedirs(checkpoint_dir)
saver = tf.train.Saver(tf.all_variables())
```

```python
def train_step(x_batch, y_batch):
    """
    A single training step
    """
    feed_dict = {
      cnn.input_x: x_batch,
      cnn.input_y: y_batch,
      cnn.dropout_keep_prob: FLAGS.dropout_keep_prob
    }
    _, step, summaries, loss, accuracy = sess.run(
        [train_op, global_step, train_summary_op, cnn.loss, cnn.accuracy],
        feed_dict)
    time_str = datetime.datetime.now().isoformat()
    print("{}: step {}, loss {:g}, acc {:g}".format(time_str, step, loss, accuracy))
    train_summary_writer.add_summary(summaries, step)

def dev_step(x_batch, y_batch, writer=None):
    """
    Evaluates model on a dev set
    """
    feed_dict = {
      cnn.input_x: x_batch,
      cnn.input_y: y_batch,
      cnn.dropout_keep_prob: 1.0
    }
    step, summaries, loss, accuracy = sess.run(
        [global_step, dev_summary_op, cnn.loss, cnn.accuracy],
        feed_dict)
    time_str = datetime.datetime.now().isoformat()
    print("{}: step {}, loss {:g}, acc {:g}".format(time_str, step, loss, accuracy))
    if writer:
        writer.add_summary(summaries, step)

# Generate batches
batches = data_helpers.batch_iter(
    list(zip(x_train, y_train)), FLAGS.batch_size, FLAGS.num_epochs)
# Training loop. For each batch...
for batch in batches:
    x_batch, y_batch = zip(*batch)
    train_step(x_batch, y_batch)
    current_step = tf.train.global_step(sess, global_step)
    if current_step % FLAGS.evaluate_every == 0:
        print("\nEvaluation:")
        dev_step(x_dev, y_dev, writer=dev_summary_writer)
        print("")
    if current_step % FLAGS.checkpoint_every == 0:
        path = saver.save(sess, checkpoint_prefix, global_step=current_step)
        print("Saved model checkpoint to {}\n".format(path))
```

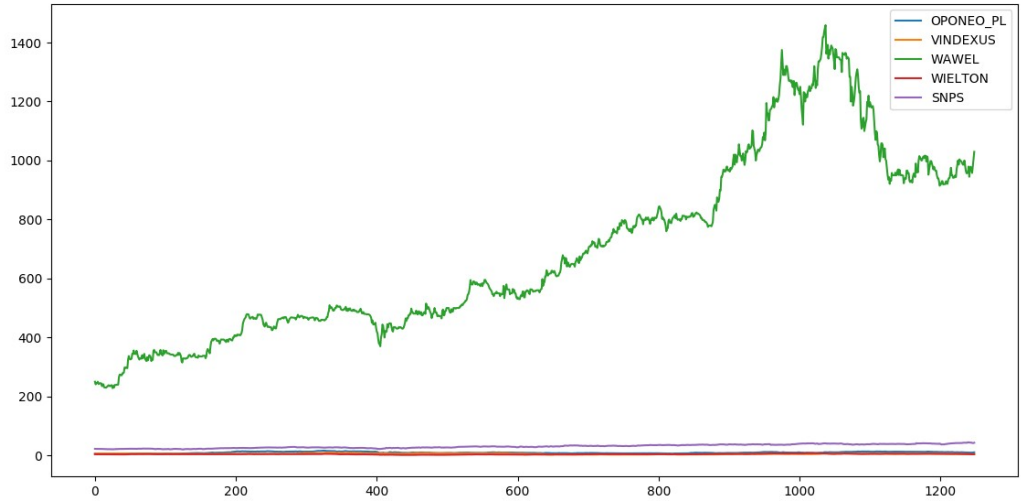
```python
#! /usr/bin/env python

import tensorflow as tf
import numpy as np
import os
import time
import datetime
import data_helpers
from text_cnn import TextCNN

# Parameters
# ==================================================

# Eval Parameters
tf.flags.DEFINE_integer("batch_size", 64, "Batch Size (default: 64)")
tf.flags.DEFINE_string("checkpoint_dir", "", "Checkpoint directory from training run")

# Misc Parameters
tf.flags.DEFINE_boolean("allow_soft_placement", True, "Allow device soft device placement")
tf.flags.DEFINE_boolean("log_device_placement", False, "Log placement of ops on devices")


FLAGS = tf.flags.FLAGS
FLAGS._parse_flags()
print("\nParameters:")
for attr, value in sorted(FLAGS.__flags.items()):
    print("{}={}".format(attr.upper(), value))
print("")

# Load data. Load your own data here
print("Loading data...")
x_test, y_test, vocabulary, vocabulary_inv = data_helpers.load_data()
y_test = np.argmax(y_test, axis=1)
print("Vocabulary size: {:d}".format(len(vocabulary)))
print("Test set size {:d}".format(len(y_test)))

print("\nEvaluating...\n")
```

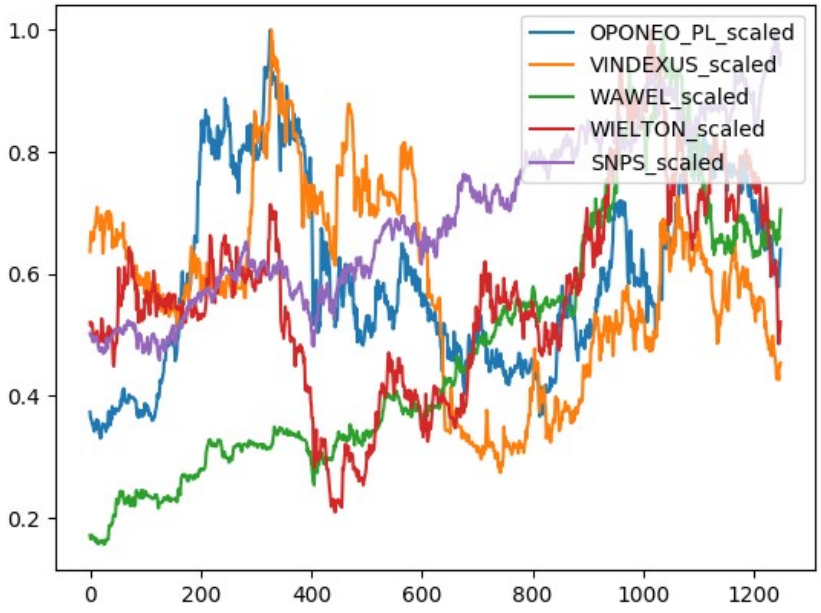
```python
print("\nEvaluating...\n")

# Evaluation
# ==================================================
checkpoint_file = tf.train.latest_checkpoint(FLAGS.checkpoint_dir)
graph = tf.Graph()
with graph.as_default():
    session_conf = tf.ConfigProto(
      allow_soft_placement=FLAGS.allow_soft_placement,
      log_device_placement=FLAGS.log_device_placement)
    sess = tf.Session(config=session_conf)
    with sess.as_default():
        # Load the saved meta graph and restore variables
        saver = tf.train.import_meta_graph("{}.meta".format(checkpoint_file))
        saver.restore(sess, checkpoint_file)

        # Get the placeholders from the graph by name
        input_x = graph.get_operation_by_name("input_x").outputs[0]
        # input_y = graph.get_operation_by_name("input_y").outputs[0]
        dropout_keep_prob = graph.get_operation_by_name("dropout_keep_prob").outputs[0]

        # Tensors we want to evaluate
        predictions = graph.get_operation_by_name("output/predictions").outputs[0]

        # Generate batches for one epoch
        batches = data_helpers.batch_iter(x_test, FLAGS.batch_size, 1, shuffle=False)

        # Collect the predictions here
        all_predictions = []

        for x_test_batch in batches:
            batch_predictions = sess.run(predictions, {input_x: x_test_batch, dropout_keep_prob: 1.0})
            all_predictions = np.concatenate([all_predictions, batch_predictions])

# Print accuracy
correct_predictions = float(sum(all_predictions == y_test))
print("Total number of test examples: {}".format(len(y_test)))
print("Accuracy: {:g}".format(correct_predictions/float(len(y_test))))
```

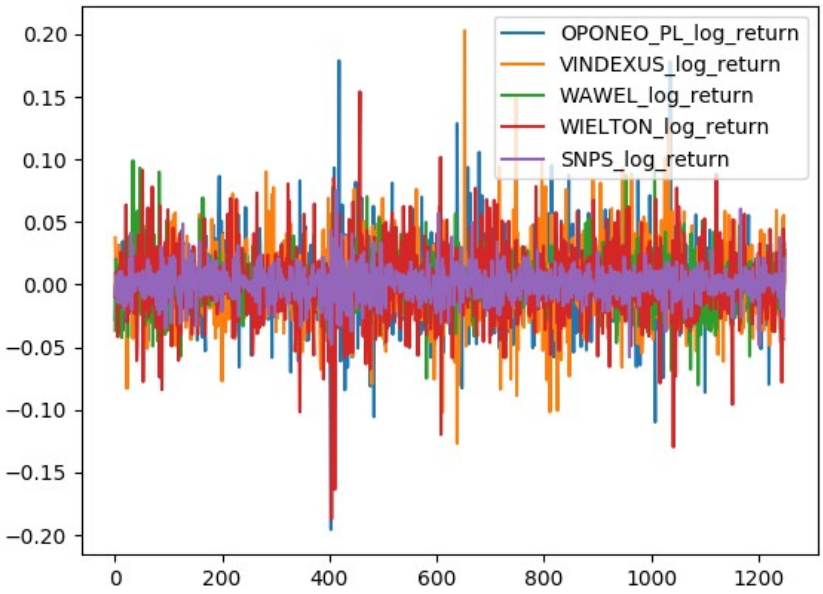

loss
accuracy
input_y
output
dropout
dropout_keep_prob
Reshape
shape
concat
concat_d...
conv-maxpool-3
conv-maxpool-4
conv-maxpool-5
embedding
input_x

Chapter 07: Making Money with Machine Learning

DATE	OPEN	HIGH	LOW	CLOSE
1999-11-18	45.5	50	40	44
1999-11-19	42.94	43	39.81	40.38
1999-11-22	41.31	44	40.06	44
1999-11-23	42.5	43.63	40.25	40.25
1999-11-24	40.13	41.94	40	41.06
1999-11-26	40.88	41.5	40.75	41.19
1999-11-29	41	42.44	40.56	42.13
1999-11-30	42	42.94	40.94	42.19
1999-12-01	42.19	43.44	41.88	42.94
1999-12-02	43.75	45	43.19	44.13
1999-12-03	44.94	45.69	44.31	44.5
1999-12-06	45.25	46.44	45.19	45.75
1999-12-07	45.75	46	44.31	45.25
1999-12-08	45.25	45.63	44.81	45.19
1999-12-09	45.25	45.94	45.25	45.81
1999-12-10	45.69	45.94	44.75	44.75
1999-12-13	45.5	46.25	44.38	45.5
1999-12-14	45.38	45.38	42.06	43
1999-12-15	42	42.31	41	41.69
1999-12-16	42	48	42	47.25
1999-12-17	46.38	47.12	45.44	45.94


OPONEO_PL
VINDEXUS
WAWEL
WIELTON
SNPS
1400
1200
1000
800
600
400
200
0
0
200
400
600
800
1000
1200


OPONEO_PL_scaled
VINDEXUS_scaled
WAWEL_scaled
WIELTON_scaled
SNPS_scaled
1.0
0.8
0.6
0.4
0.2
0
200
400
600
800
1000
1200

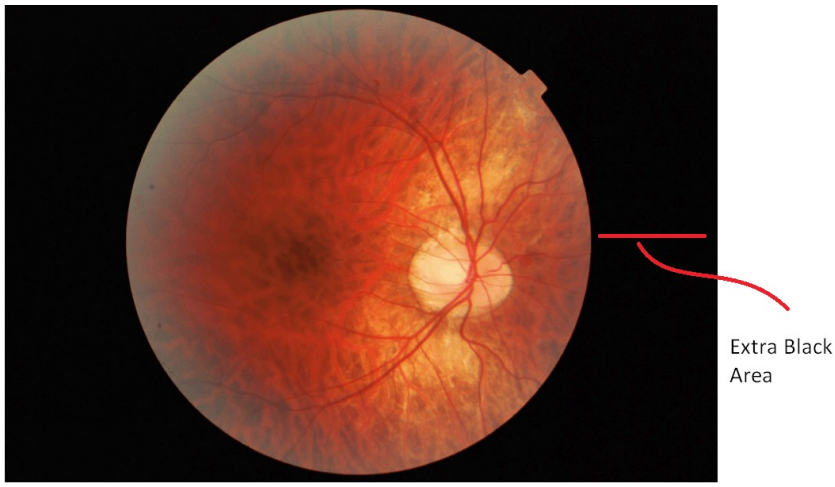

OPONEO_PL_log_return
VINDEXUS_log_return
WAWEL_log_return
WIELTON_log_return
SNPS_log_return
0.20
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
−0.20
0
200
400
600
800
1000
1200

Chapter 08: The Doctor Will See You Now

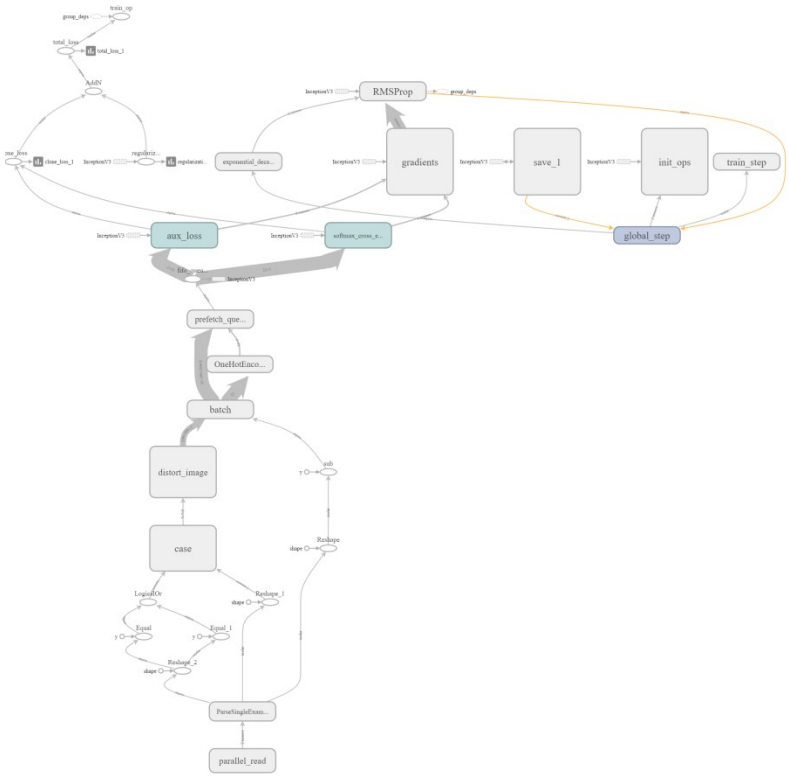
ıis PC › DATA (D:) › datasets › diabetic

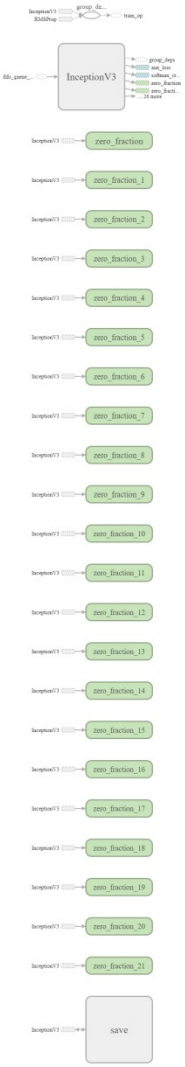
Search diabetic

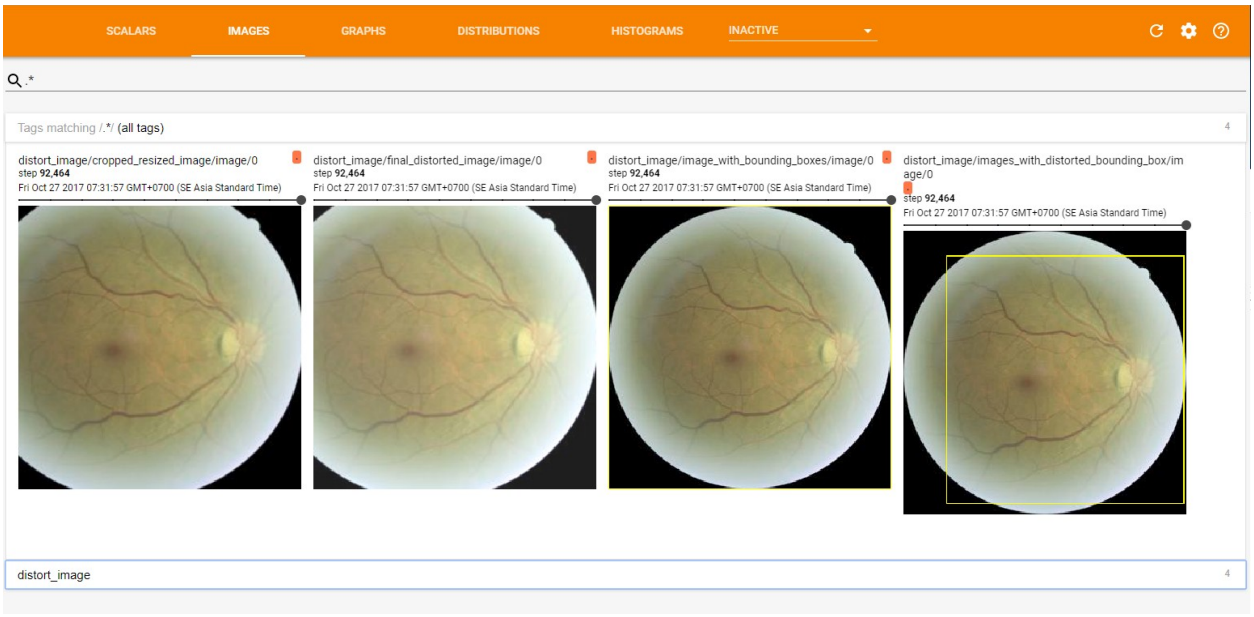
Name	Date modified	Type	Size
processed_images	10/13/2017 8:07 AM	File folder	
train	2/12/2015 9:47 AM	File folder	
diabetic_train_00000-of-00005.tfrecord	10/25/2017 11:07 ...	TFRECORD File	49,504 KB
diabetic_train_00001-of-00005.tfrecord	10/25/2017 11:07 ...	TFRECORD File	49,458 KB
diabetic_train_00002-of-00005.tfrecord	10/25/2017 11:07 ...	TFRECORD File	49,556 KB
diabetic_train_00003-of-00005.tfrecord	10/25/2017 11:08 ...	TFRECORD File	49,740 KB
diabetic_train_00004-of-00005.tfrecord	10/25/2017 11:08 ...	TFRECORD File	48,418 KB
diabetic_validation_00000-of-00005.tfrec...	10/25/2017 11:08 ...	TFRECORD File	8,546 KB
diabetic_validation_00001-of-00005.tfrec...	10/25/2017 11:08 ...	TFRECORD File	8,469 KB
diabetic_validation_00002-of-00005.tfrec...	10/25/2017 11:08 ...	TFRECORD File	8,450 KB
diabetic_validation_00003-of-00005.tfrec...	10/25/2017 11:08 ...	TFRECORD File	8,420 KB
diabetic_validation_00004-of-00005.tfrec...	10/25/2017 11:08 ...	TFRECORD File	8,233 KB
labels	10/26/2017 8:17 AM	Text Document	25 KB
train.zip	10/3/2017 2:30 PM	WinRAR archive	8,192,000 KB
train.zip.002	10/3/2017 3:21 PM	002 File	8,192,000 KB
train.zip.003	10/5/2017 12:45 AM	003 File	8,192,000 KB
train.zip.004	10/5/2017 12:59 AM	004 File	8,192,000 KB
train.zip.005	10/5/2017 12:17 AM	005 File	1,400,404 KB
trainLabels	2/6/2015 2:56 PM	Microsoft Excel C...	455 KB
trainLabels.csv	10/11/2017 3:44 PM	WinRAR ZIP archive	70 KB
unique_labels_file	10/13/2017 10:39 ...	Text Document	1 KB

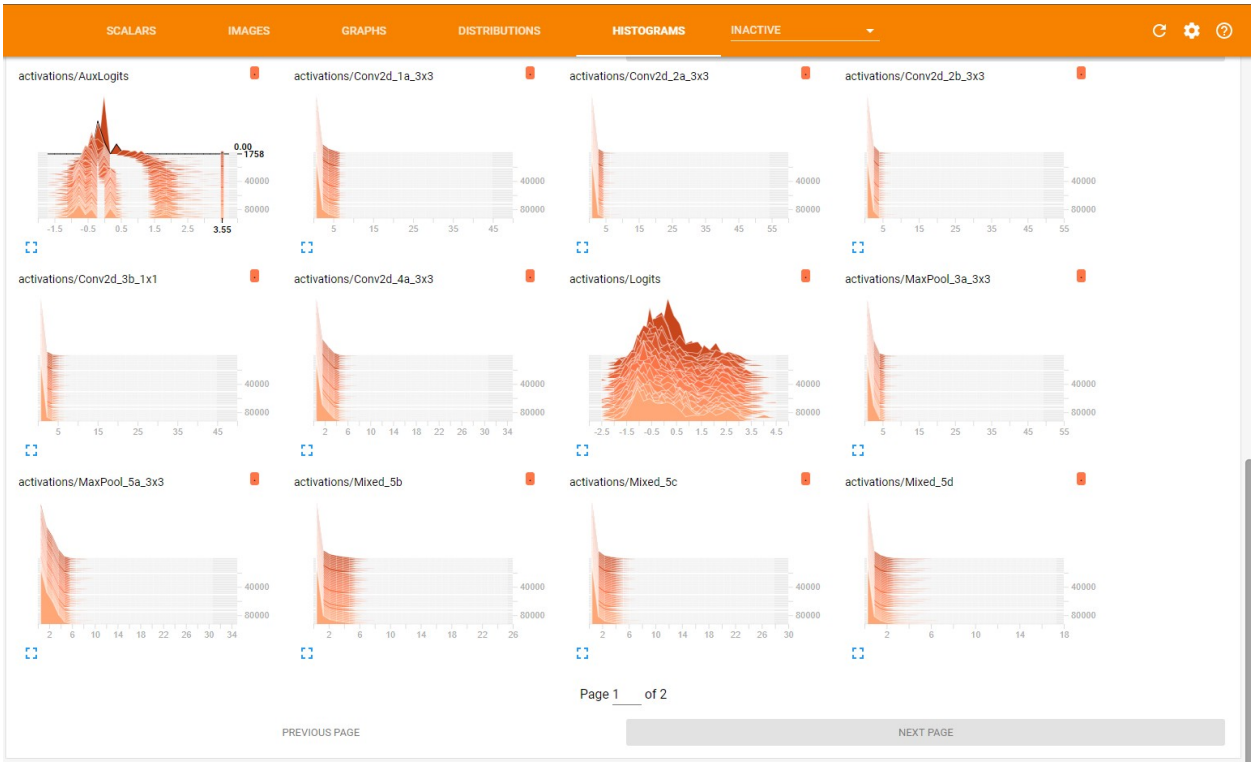


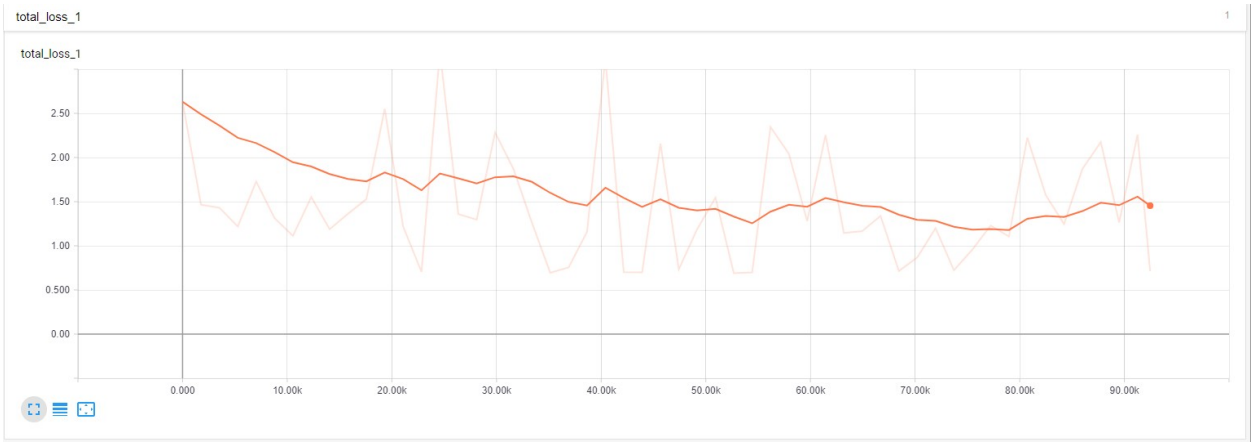

```
INFO:tensorflow:Evaluation [30/52]
INFO:tensorflow:Evaluation [31/52]
INFO:tensorflow:Evaluation [32/52]
INFO:tensorflow:Evaluation [33/52]
INFO:tensorflow:Evaluation [34/52]
INFO:tensorflow:Evaluation [35/52]
INFO:tensorflow:Evaluation [36/52]
INFO:tensorflow:Evaluation [37/52]
INFO:tensorflow:Evaluation [38/52]
INFO:tensorflow:Evaluation [39/52]
INFO:tensorflow:Evaluation [40/52]
INFO:tensorflow:Evaluation [41/52]
INFO:tensorflow:Evaluation [42/52]
INFO:tensorflow:Evaluation [43/52]
INFO:tensorflow:Evaluation [44/52]
INFO:tensorflow:Evaluation [45/52]
INFO:tensorflow:Evaluation [46/52]
INFO:tensorflow:Evaluation [47/52]
INFO:tensorflow:Evaluation [48/52]
INFO:tensorflow:Evaluation [49/52]
INFO:tensorflow:Evaluation [50/52]
INFO:tensorflow:Evaluation [51/52]
INFO:tensorflow:Evaluation [52/52]
2017-11-03 13:47:49.151625: I C:\tf_jenkins\home\workspace\rel-win\M\windows-gpu\PY\35\tensorflow\core\kernels\logging_o
ps.cc:79] eval/Accuracy[0.751153827]
2017-11-03 13:47:49.492745: I C:\tf_jenkins\home\workspace\rel-win\M\windows-gpu\PY\35\tensorflow\core\kernels\logging_o
ps.cc:79] eval/Recall_5[1]
INFO:tensorflow:Finished evaluation at 2017-11-03-06:47:49
```

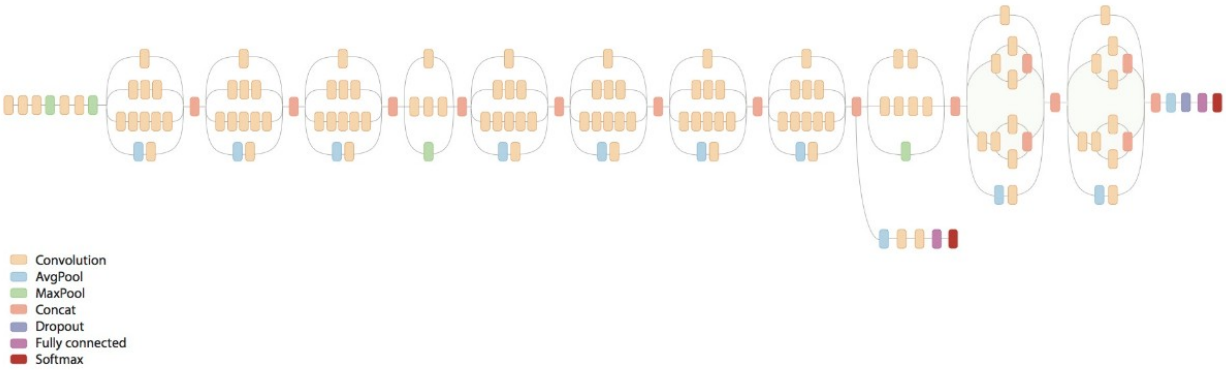

train_op
group_deps
total_loss
total_loss_1
AddN
aux_loss
clone_loss_1
regularizr...
regularizati...
exponential_deca...
RMSProp
gradients
save_1
init_ops
train_step
aux_loss
softmax_cross_e...
global_step
fifo_queue
prefetch_que...
OneHotEnco...
batch
distort_image
case
LogicalOr
Equal
Equal_1
Reshape_1
Reshape_2
Reshape
PaseSingleExam...
parallel_read


group_dc...
train_op
InceptionV3
zero_fraction
zero_fraction_1
zero_fraction_2
zero_fraction_3
zero_fraction_4
zero_fraction_5
zero_fraction_6
zero_fraction_7
zero_fraction_8
zero_fraction_9
zero_fraction_10
zero_fraction_11
zero_fraction_12
zero_fraction_13
zero_fraction_14
zero_fraction_15
zero_fraction_16
zero_fraction_17
zero_fraction_18
zero_fraction_19
zero_fraction_20
zero_fraction_21
save


SCALARS
IMAGES
GRAPHS
DISTRIBUTIONS
HISTOGRAMS
INACTIVE
.*
Tags matching /.*/ (all tags)
4
distort_image/cropped_resized_image/image/0
step 92,464
Fri Oct 27 2017 07:31:57 GMT+0700 (SE Asia Standard Time)
distort_image/final_distorted_image/image/0
step 92,464
Fri Oct 27 2017 07:31:57 GMT+0700 (SE Asia Standard Time)
distort_image/image_with_bounding_boxes/image/0
step 92,464
Fri Oct 27 2017 07:31:57 GMT+0700 (SE Asia Standard Time)
distort_image/images_with_distorted_bounding_box/image/0
step 92,464
Fri Oct 27 2017 07:31:57 GMT+0700 (SE Asia Standard Time)
distort_image
4


SCALARS
IMAGES
GRAPHS
DISTRIBUTIONS
HISTOGRAMS
INACTIVE
activations/AuxLogits
activations/Conv2d_1a_3x3
activations/Conv2d_2a_3x3
activations/Conv2d_2b_3x3
activations/Conv2d_3b_1x1
activations/Conv2d_4a_3x3
activations/Logits
activations/MaxPool_3a_3x3
activations/MaxPool_5a_3x3
activations/Mixed_5b
activations/Mixed_5c
activations/Mixed_5d
Page 1 of 2
PREVIOUS PAGE
NEXT PAGE

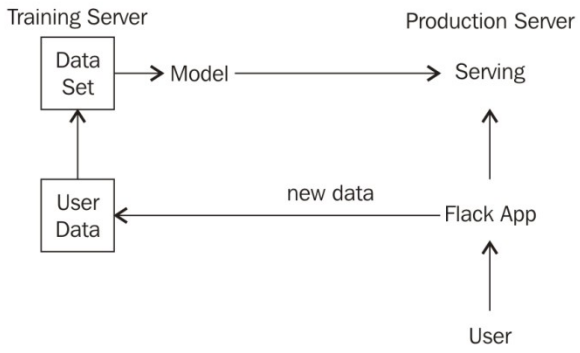

total_loss_1
1
total_loss_1
2.50
2.00
1.50
1.00
0.500
0.00
0.000
10.00k
20.00k
30.00k
40.00k
50.00k
60.00k
70.00k
80.00k
90.00k

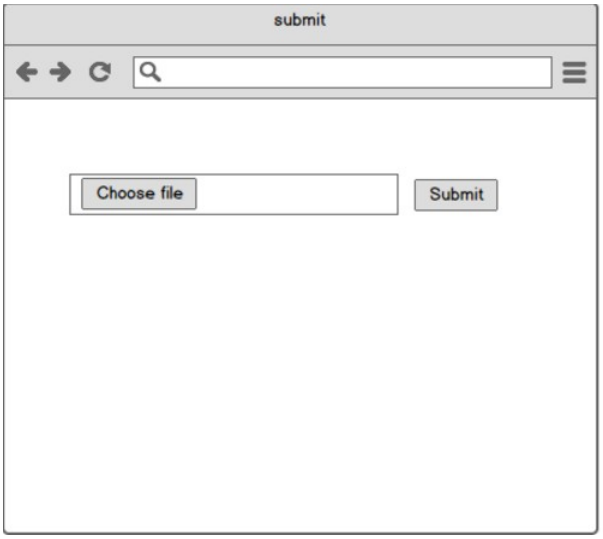

Convolution
AvgPool
MaxPool
Concat
Dropout
Fully connected
Softmax

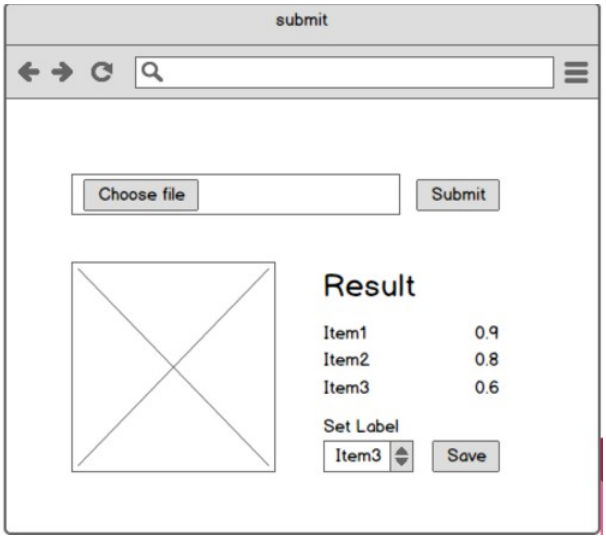
```
fileInputDICOM = r.rpop(redisQueue)
newFile = fileInputDICOM.replace(IN_DIR, OUT_DIR) + ".png"
print(num, ":", fileInputDICOM)
plan = dicomio.read_file(fileInputDICOM)
shape = plan.pixel_array.shape
wBuffer=MAX_SIZE-shape[0]
hBuffer=MAX_SIZE-shape[1]
image_2d = []
for row in plan.pixel_array:
    pixels = []
    for col in row:
        pixels.append(col)
    for h in range(hBuffer):
        pixels.append(32767)
    image_2d.append(pixels)
for w in range(wBuffer):
    image_2d.append([32767]*MAX_SIZE)

# Rescalling greyscale between 0-255
image_2d_scaled = []
for row in image_2d:
    row_scaled = []
    for col in row:
        col_scaled = int((float(col)/float(max_val))*255.0)
        col_scaled = 255.0 - col_scaled
        row_scaled.append(col_scaled)
    image_2d_scaled.append(row_scaled)

if not os.path.exists(os.path.dirname(newFile)):
    try:
        os.makedirs(os.path.dirname(newFile))
    except OSError as exc: # Guard against race condition
        if exc.errno != errno.EEXIST:
            raise

f = open(newFile, 'wb')
w = png.Writer(MAX_SIZE, MAX_SIZE, greyscale=True)
w.write(f, image_2d_scaled)
f.close()
```

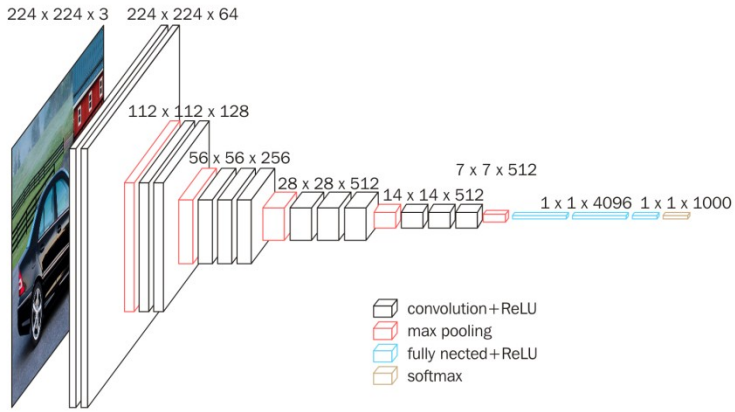
```
fileInputDICOM = r.rpop(redisQueue)
newFile = fileInputDICOM.replace(IN_DIR, OUT_DIR) + ".png"
print(num, ":", fileInputDICOM)
plan = dicomio.read_file(fileInputDICOM)
shape = plan.pixel_array.shape
wBuffer=MAX_SIZE-shape[0]
hBuffer=MAX_SIZE-shape[1]
image_2d = []
for row in plan.pixel_array:
    pixels = []
    for col in row:
        pixels.append(col)
    for h in range(hBuffer):
        pixels.append(32767)
    image_2d.append(pixels)
for w in range(wBuffer):
    image_2d.append([32767]*MAX_SIZE)

# Rescalling greyscale between 0-255
image_2d_scaled = []
for row in image_2d:
    row_scaled = []
    for col in row:
        col_scaled = int((float(col)/float(max_val))*255.0)
        col_scaled = 255.0 - col_scaled
        row_scaled.append(col_scaled)
    image_2d_scaled.append(row_scaled)

if not os.path.exists(os.path.dirname(newFile)):
    try:
        os.makedirs(os.path.dirname(newFile))
    except OSError as exc: # Guard against race condition
        if exc.errno != errno.EEXIST:
            raise

f = open(newFile, 'wb')
w = png.Writer(MAX_SIZE, MAX_SIZE, greyscale=True)
w.write(f, image_2d_scaled)
f.close()
```

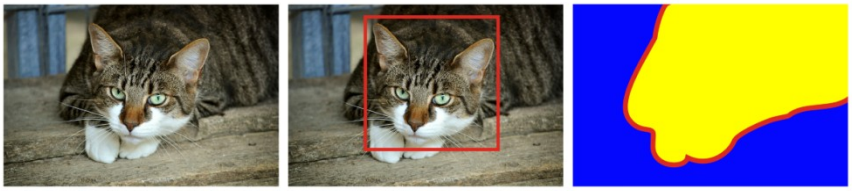
Chapter 09: Cruise Control – Automation

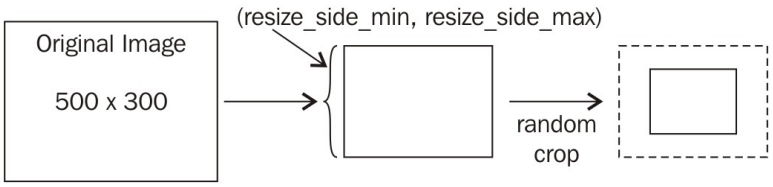












Chapter 10: Go Live and Go Big

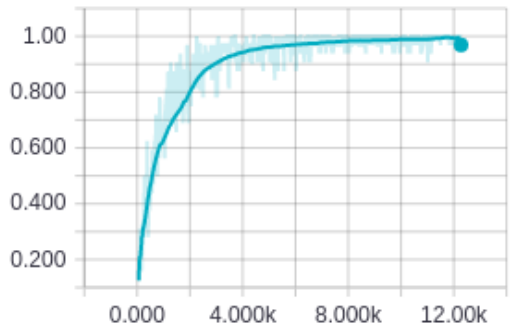
Compute: Amazon EC2 Instances:

	Description	Instances	Usage	Type	Billing Option	Monthly Cost
		1	100 % Utilized/Mo	Linux on p2.xlarge	On-Demand (No Contract	$ 658.80
	Add New Row					

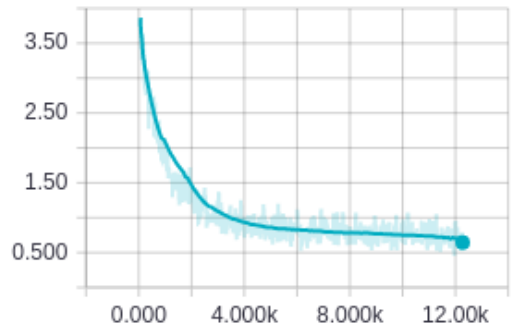
Per Instance Prices & Projected Costs (all in USD)

Select	Name	Upfront Price	Effective Hourly Cost	Effective Monthly Cost	1 Year Cost	3 Year Cost
◉	On-Demand (No Contract)	---	0.900	658.80	7905.60	23716.80
○	1 Yr No Upfront Reserved	0.00	0.614	448.22	5378.64	16135.92
○	1 Yr Partial Upfront Reserved	2562.00	0.585	427.02	5124.24	15372.72
○	1 Yr All Upfront Reserved	5022.00	0.573	418.50	5022.00	15066.00
○	3 Yr Partial Upfront Reserved	5584.00	0.425	310.24	---	11168.32
○	3 Yr All Upfront Reserved	10499.00	0.399	291.64	---	10499.00
○	3 Yr No Upfront Convertible	0.00	0.528	385.44	---	13875.84
○	3 Yr Partial Upfront Convertible	6422.00	0.488	356.51	---	12834.32
○	3 Yr All Upfront Convertible	12588.00	0.479	349.67	---	12588.00

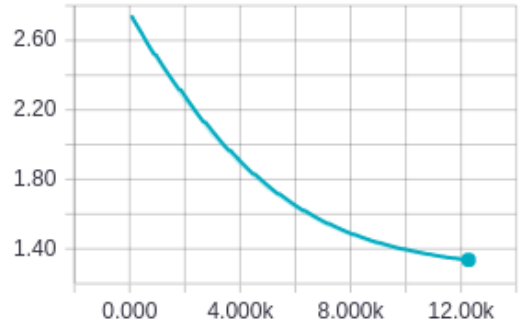
train/accuracy



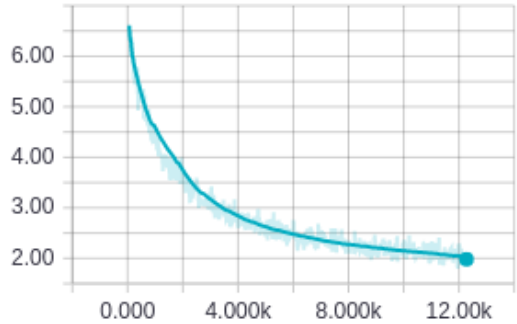
train/cross_entropy_loss



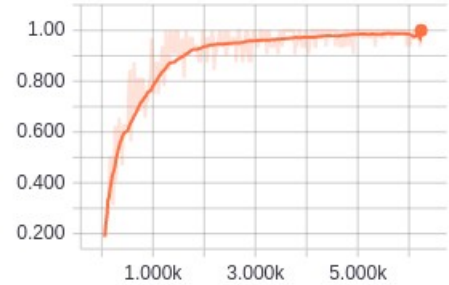
train/regularization_loss



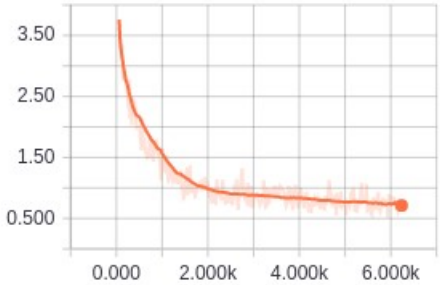
train/total_loss



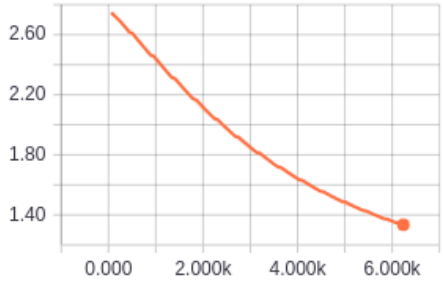
train/accuracy



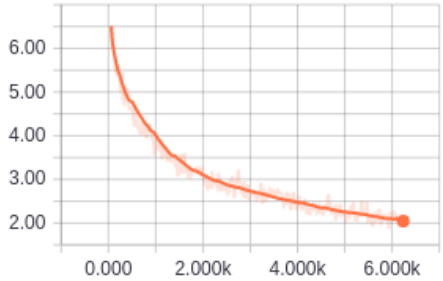
train/cross_entropy_loss



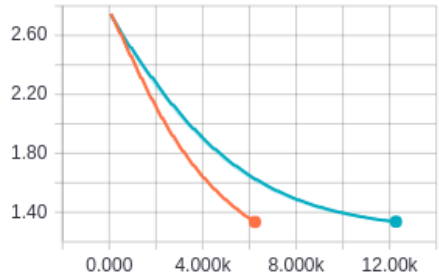
train/regularization_loss



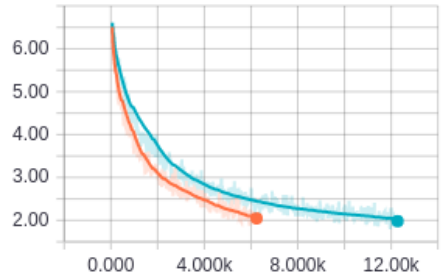
train/total_loss



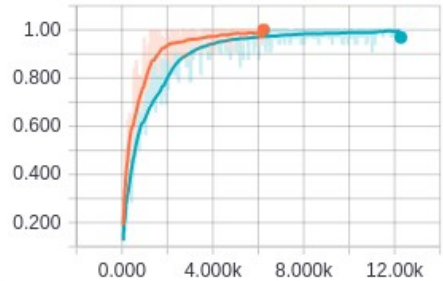
train/regularization_loss



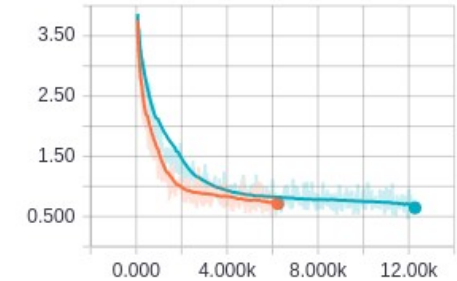
train/total_loss



train/accuracy



train/cross_entropy_loss



Discover, preview and complete HITs on the new Worker website. Try it out Today!

amazonmechanical turk
Artificial Artificial Intelligence

Your Account | HITs | Qualifications

Already have an account?
Sign in as a Worker | Requester

Introduction | Dashboard | Status | Account Settings

Mechanical Turk is a marketplace for work.

We give businesses and developers access to an on-demand, scalable workforce.
Workers select from thousands of tasks and work whenever it's convenient.

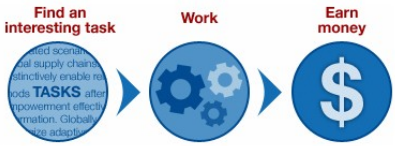
210,950 HITs available. View them now.

Make Money
by working on HITs

HITs - *Human Intelligence Tasks* - are individual tasks that you work on. Find HITs now.

As a Mechanical Turk Worker you:

- Can work from home
- Choose your own work hours
- Get paid for doing good work



Find HITs Now

or learn more about being a **Worker**

Get Results
from Mechanical Turk Workers

Ask workers to complete HITs - *Human Intelligence Tasks* - and get results using Mechanical Turk. Get Started.

As a Mechanical Turk Requester you:

- Have access to a global, on-demand, 24 x 7 workforce
- Get thousands of HITs completed in minutes
- Pay only when you're satisfied with the results

Get Started

FAQ | Contact Us | Careers at Mechanical Turk | Developers | Press | Policies | State Licensing | Blog | Service Health Dashboard

An amazon.com. company

Chapter 11: Going Further - 21 Problems

```
                # Concatenate the groups
<<<<<<< HEAD
                output = tf.concat(output_groups, 3)
            # Add the biases
=======
                output = tf.concat(3, output_groups)
            # Add the bias
>>>>>>> 1324afea14711a09ea9d689ce06ea0b0cdf84a17
```

```
                # Concatenate the groups

                output = tf.concat(3, output_groups)
            # Add the bias

            if biased:
```

```
    @layer
    def concat(self, inputs, axis, name):
<<<<<<< HEAD
        return tf.concat(values=inputs, axis=axis,  name=name)
=======
        return tf.concat(axis=axis, values=inputs, name=name)
>>>>>>> 1324afea14711a09ea9d689ce06ea0b0cdf84a17

```

```

    @layer
    def concat(self, inputs, axis, name):
        return tf.concat(axis=axis, values=inputs, name=name)

```

Appendix: Advanced Installation

```
libcuda1-375 - NVIDIA CUDA runtime library
nvidia-304 - NVIDIA legacy binary driver - version 304.135
nvidia-304-updates - Transitional package for nvidia-304
nvidia-304-updates-dev - Transitional package for nvidia-304-dev
nvidia-340 - NVIDIA binary driver - version 340.102
nvidia-361 - Transitional package for nvidia-367
nvidia-361-dev - Transitional package for nvidia-367-dev
nvidia-367 - Transitional package for nvidia-375
nvidia-367-dev - Transitional package for nvidia-375-dev
nvidia-375 - NVIDIA binary driver - version 375.66
nvidia-375-dev - NVIDIA binary Xorg driver development files
nvidia-libopencl1-367 - Transitional package for nvidia-libopencl1-375
nvidia-libopencl1-375 - NVIDIA OpenCL Driver and ICD Loader library
nvidia-opencl-icd-304-updates - Transitional package for nvidia-opencl-icd-304
nvidia-opencl-icd-361 - Transitional package for nvidia-opencl-icd-367
nvidia-opencl-icd-367 - Transitional package for nvidia-opencl-icd-375
nvidia-opencl-icd-375 - NVIDIA OpenCL ICD
nvidia-libopencl1-304-updates - Transitional package for nvidia-libopencl1-304
nvidia-libopencl1-361 - Transitional package for nvidia-libopencl1-367
nvidia-352 - Transitional package for nvidia-375
cuda-visual-tools-8-0 - CUDA visual tools
nvidia-361-updates - Transitional package for nvidia-375
cuda-drivers - CUDA Driver meta-package
nvidia-opencl-icd-352-updates - Transitional package for nvidia-opencl-icd-375
nvidia-libopencl1-361-updates - Transitional package for nvidia-libopencl1-375
nvidia-libopencl1-352 - Transitional package for nvidia-libopencl1-375
nvidia-opencl-icd-361-updates - Transitional package for nvidia-opencl-icd-375
nvidia-modprobe - Load the NVIDIA kernel driver and create device files
nvidia-libopencl1-352-updates - Transitional package for nvidia-libopencl1-375
nvidia-352-updates-dev - Transitional package for nvidia-375-dev
nvidia-352-dev - Transitional package for nvidia-375-dev
nvidia-361-updates-dev - Transitional package for nvidia-375-dev
nvidia-352-updates - Transitional package for nvidia-375
nvidia-opencl-icd-352 - Transitional package for nvidia-opencl-icd-375
ubuntu@ubuntu-TITAN:~/Downloads$
```

```
ubuntu@ubuntu-TITAN:~/Downloads$ sudo apt-get install nvidia-375
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  cuda-command-line-tools-8-0 cuda-core-8-0 cuda-cublas-8-0
  cuda-cublas-dev-8-0 cuda-cudart-8-0 cuda-cudart-dev-8-0 cuda-cufft-8-0
  cuda-cufft-dev-8-0 cuda-curand-8-0 cuda-curand-dev-8-0 cuda-cusolver-8-0
  cuda-cusolver-dev-8-0 cuda-cusparse-8-0 cuda-cusparse-dev-8-0
  cuda-documentation-8-0 cuda-driver-dev-8-0 cuda-license-8-0
  cuda-misc-headers-8-0 cuda-npp-8-0 cuda-npp-dev-8-0 cuda-nvgraph-8-0
  cuda-nvgraph-dev-8-0 cuda-nvml-dev-8-0 cuda-nvrtc-8-0 cuda-nvrtc-dev-8-0
  cuda-samples-8-0 cuda-toolkit-8-0 cuda-visual-tools-8-0 libgles1-mesa
  libxmu-dev libxmu-headers linux-headers-4.8.0-52
  linux-headers-4.8.0-52-generic linux-headers-4.8.0-54
  linux-headers-4.8.0-54-generic linux-image-4.8.0-52-generic
  linux-image-4.8.0-54-generic linux-image-extra-4.8.0-52-generic
  linux-image-extra-4.8.0-54-generic linux-signed-image-4.8.0-52-generic
  linux-signed-image-4.8.0-54-generic nvidia-modprobe screen snap-confine
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libcuda1-375 nvidia-opencl-icd-375
The following NEW packages will be installed:
  libcuda1-375 nvidia-375 nvidia-opencl-icd-375
0 upgraded, 3 newly installed, 0 to remove and 65 not upgraded.
Need to get 75,2 MB of archives.
After this operation, 333 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

```
 - Installation
   - Installing to /lib/modules/4.8.0-58-generic/updates/dkms/

nvidia_375_drm.ko:
Running module version sanity check.
 - Original module
   - No original module exists within this kernel
 - Installation
   - Installing to /lib/modules/4.8.0-58-generic/updates/dkms/

nvidia_375_uvm.ko:
Running module version sanity check.
 - Original module
   - No original module exists within this kernel
 - Installation
   - Installing to /lib/modules/4.8.0-58-generic/updates/dkms/

depmod....

DKMS: install completed.
Setting up libcuda1-375 (375.66-0ubuntu0.16.04.1) ...
Setting up nvidia-opencl-icd-375 (375.66-0ubuntu0.16.04.1) ...
Processing triggers for libc-bin (2.23-0ubuntu9) ...
/sbin/ldconfig.real: /usr/local/lib/libpocketsphinx.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/lib/nvidia-375/libEGL.so.1 is not a symbolic link

/sbin/ldconfig.real: /usr/lib32/nvidia-375/libEGL.so.1 is not a symbolic link

Processing triggers for initramfs-tools (0.122ubuntu8.8) ...
update-initramfs: Generating /boot/initrd.img-4.8.0-58-generic
Processing triggers for shim-signed (1.28~16.04.1+0.9+1474479173.6c180c6-1ubuntu
1) ...
Secure Boot not enabled on this system.
ubuntu@ubuntu-TITAN:~/Downloads$
```

NVIDIA ACCELERATED COMPUTING Downloads Training Ecosystem Forums Log

CUDA Toolkit Download

Home > ComputeWorks > CUDA Toolkit > CUDA Toolkit Download

Select Target Platform

Click on the green buttons that describe your target platform. Only supported platforms will be shown.

Operating System: Windows | Linux | Mac OSX

Refer to the Release Notes for enhancements and bug fixes in the latest version.

Get Started

```
End User License Agreement
--------------------------


Preface
-------

The following contains specific license terms and conditions
for four separate NVIDIA products. By accepting this
agreement, you agree to comply with all the terms and
conditions applicable to the specific product(s) included
herein.


NVIDIA CUDA Toolkit


Description

The NVIDIA CUDA Toolkit provides command-line and graphical
tools for building, debugging and optimizing the performance
of applications accelerated by NVIDIA GPUs, runtime and math
libraries, and documentation including programming guides,
--More--(0%)
```

```
---------------------------


Preface
-------

The following contains specific license terms and conditions
for four separate NVIDIA products. By accepting this
agreement, you agree to comply with all the terms and
conditions applicable to the specific product(s) included
herein.


NVIDIA CUDA Toolkit


Description

The NVIDIA CUDA Toolkit provides command-line and graphical
tools for building, debugging and optimizing the performance
of applications accelerated by NVIDIA GPUs, runtime and math
libraries, and documentation including programming guides,
Do you accept the previously read EULA?
accept/decline/quit:
```

```
Default Install Location of CUDA Toolkit

Windows platform:

Do you accept the previously read EULA?
accept/decline/quit: accept

Install NVIDIA Accelerated Graphics Driver for Linux-x86_64 375.26?
(y)es/(n)o/(q)uit: n

Install the CUDA 8.0 Toolkit?
(y)es/(n)o/(q)uit: y

Enter Toolkit Location
 [ default is /usr/local/cuda-8.0 ]:

Do you want to install a symbolic link at /usr/local/cuda?
(y)es/(n)o/(q)uit: y

Install the CUDA 8.0 Samples?
(y)es/(n)o/(q)uit: y

Enter CUDA Samples Location
 [ default is /home/ubuntu ]:

Installing the CUDA Toolkit in /usr/local/cuda-8.0 ...
```

```
===========
= Summary =
===========

Driver:   Not Selected
Toolkit:  Installed in /usr/local/cuda-8.0
Samples:  Installed in /home/ubuntu

Please make sure that
 -   PATH includes /usr/local/cuda-8.0/bin
 -   LD_LIBRARY_PATH includes /usr/local/cuda-8.0/lib64, or, add /usr/local/cuda
-8.0/lib64 to /etc/ld.so.conf and run ldconfig as root

To uninstall the CUDA Toolkit, run the uninstall script in /usr/local/cuda-8.0/b
in

Please see CUDA_Installation_Guide_Linux.pdf in /usr/local/cuda-8.0/doc/pdf for
detailed information on setting up CUDA.

***WARNING: Incomplete installation! This installation did not install the CUDA
Driver. A driver of version at least 361.00 is required for CUDA 8.0 functionali
ty to work.
To install the driver using this installer, run the following command, replacing
 <CudaInstaller> with the name of this run file:
    sudo <CudaInstaller>.run -silent -driver

Logfile is /tmp/cuda_install_10612.log
ubuntu@ubuntu-TITAN:~/Downloads$
```

```
ubuntu@ubuntu-TITAN:~/Downloads$ nvidia-smi
Mon Jul 17 23:52:48 2017
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 375.66                 Driver Version: 375.66                    |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  GeForce GTX TIT...  Off  | 0000:01:00.0      On |                  N/A |
| 22%   42C    P8    18W / 250W |    446MiB / 12204MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID  Type  Process name                             Usage      |
|=============================================================================|
|    0      1076    G   /usr/lib/xorg/Xorg                           175MiB |
|    0      2550    G   /usr/bin/gnome-shell                          95MiB |
|    0      3664    G   ...el-token=27266A619B706F41F299C68AACC2AAFF    95MiB |
|    0     25087    G   ...s-passed-by-fd --v8-snapshot-passed-by-fd    78MiB |
+-----------------------------------------------------------------------------+
ubuntu@ubuntu-TITAN:~/Downloads$
```

NVIDIA DEVELOPER COMPUTEWORKS GAMEWORKS JETPACK DESIGNWORKS DRIVE Member Area Hello, Quan

Home > ComputeWorks > Deep Learning > Software > cuDNN Download

cuDNN Download

NVIDIA cuDNN is a GPU-accelerated library of primitives for deep neural networks.

☑ **I Agree To the Terms of the cuDNN Software License Agreement**

Please check your framework documentation to determine the recommended version of cuDNN.

If you are using cuDNN with a Pascal (GTX 1080, GTX 1070), version 5 or later is required.

- Download cuDNN v6.0 (April 27, 2017), for CUDA 8.0
- Download cuDNN v6.0 (April 27, 2017), for CUDA 7.5
- Download cuDNN v5.1 (Jan 20, 2017), for CUDA 8.0
- Download cuDNN v5.1 (Jan 20, 2017), for CUDA 7.5
- Download cuDNN v5 (May 27, 2016), for CUDA 8.0
- Download cuDNN v5 (May 12, 2016), for CUDA 7.5
- Download cuDNN v4 (Feb 10, 2016), for CUDA 7.0 and later.

Archived cuDNN Releases

Download cuDNN v5.1 (Jan 20, 2017), for CUDA 8.0

cuDNN User Guide

cuDNN Install Guide

cuDNN v5.1 Library for Linux

```
ubuntu@ubuntu-TITAN:~/Downloads$ tar -xf cudnn-8.0-linux-x64-v5.1.tgz
ubuntu@ubuntu-TITAN:~/Downloads$ cd cuda
ubuntu@ubuntu-TITAN:~/Downloads/cuda$ sudo cp -P include/cudnn.h /usr/include/
ubuntu@ubuntu-TITAN:~/Downloads/cuda$ sudo cp -P lib64/libcudnn* /usr/lib/x86_64-linux-gnu/
ubuntu@ubuntu-TITAN:~/Downloads/cuda$ sudo chmod a+r /usr/lib/x86_64-linux-gnu/libcudnn*
ubuntu@ubuntu-TITAN:~/Downloads/cuda$
```

```
Installing collected packages: html5lib, bleach, markdown, backports.weakref, te
nsorflow-gpu
  Running setup.py install for html5lib ... done
  Running setup.py install for markdown ... done
Successfully installed backports.weakref-1.0rc1 bleach-1.5.0 html5lib-0.9999999
markdown-2.6.8 tensorflow-gpu-1.2.1
ubuntu@ubuntu-TITAN:~/Downloads$
```

```
Python 2.7.12 (default, Nov 19 2016, 06:48:10)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import tensorflow as tf
>>> tf.Session()
2017-07-18 00:14:41.406864: W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow lib
rary wasn't compiled to use SSE4.1 instructions, but these are available on your machine and could
 speed up CPU computations.
2017-07-18 00:14:41.406882: W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow lib
rary wasn't compiled to use SSE4.2 instructions, but these are available on your machine and could
 speed up CPU computations.
2017-07-18 00:14:41.406885: W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow lib
rary wasn't compiled to use AVX instructions, but these are available on your machine and could sp
eed up CPU computations.
2017-07-18 00:14:41.406888: W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow lib
rary wasn't compiled to use AVX2 instructions, but these are available on your machine and could s
peed up CPU computations.
2017-07-18 00:14:41.406891: W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow lib
rary wasn't compiled to use FMA instructions, but these are available on your machine and could sp
eed up CPU computations.
2017-07-18 00:14:41.537889: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:893] successful
 NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so r
eturning NUMA node zero
2017-07-18 00:14:41.538278: I tensorflow/core/common_runtime/gpu/gpu_device.cc:940] Found device 0
 with properties:
name: GeForce GTX TITAN X
major: 5 minor: 2 memoryClockRate (GHz) 1.076
pciBusID 0000:01:00.0
Total memory: 11.92GiB
Free memory: 11.27GiB
2017-07-18 00:14:41.538293: I tensorflow/core/common_runtime/gpu/gpu_device.cc:961] DMA: 0
2017-07-18 00:14:41.538296: I tensorflow/core/common_runtime/gpu/gpu_device.cc:971] 0:   Y
2017-07-18 00:14:41.538319: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1030] Creating Tens
orFlow device (/gpu:0) -> (device: 0, name: GeForce GTX TITAN X, pci bus id: 0000:01:00.0)
<tensorflow.python.client.session.Session object at 0x7f0a3c12a0d0>
>>>
```

```
Do you approve the license terms? [yes|no]
>>> yes

Miniconda2 will now be installed into this location:
/home/ubuntu/miniconda2

  - Press ENTER to confirm the location
  - Press CTRL-C to abort the installation
  - Or specify a different location below
```

```
installing: yaml-0.1.6-0 ...
installing: zlib-1.2.8-3 ...
installing: conda-4.3.21-py27_0 ...
installing: pip-9.0.1-py27_1 ...
installing: wheel-0.29.0-py27_0 ...
Python 2.7.13 :: Continuum Analytics, Inc.
creating default environment...
installation finished.
Do you wish the installer to prepend the Miniconda2 install location
to PATH in your /home/ubuntu/.bashrc ? [yes|no]
[no] >>> yes

Prepending PATH=/home/ubuntu/miniconda2/bin to PATH in /home/ubuntu/.bashrc
A backup will be made to: /home/ubuntu/.bashrc-miniconda2.bak


For this change to become active, you have to open a new terminal.

Thank you for installing Miniconda2!

Share your notebooks and packages on Anaconda Cloud!
Sign up for free: https://anaconda.org

ubuntu@ubuntu-TITAN:~/github/chapter11$
```

```
name: env2
channels:
    - https://conda.anaconda.org/menpo
    - conda-forge
dependencies:
    - python=2.7
    - opencv3
    - numpy
    - matplotlib
    - jupyter
    - pillow
    - scikit-learn
    - scikit-image
    - scipy
    - h5py
    - eventlet
    - flask-socketio
    - seaborn
    - pandas
    - ffmpeg
    - pyqt
    - pip:
        - moviepy
        - tensorflow-gpu
        - keras
        - prettytable
~
~
~
                                                        1,1           All
```

```
Successfully installed backports.weakref-1.0rc1 funcsigs-1.0.2 imageio-2.1.2 keras-2.0.6 markdown-
2.6.8 mock-2.0.0 moviepy-0.2.3.2 pbr-3.1.1 prettytable-0.7.2 protobuf-3.3.0 tensorflow-gpu-1.2.1 t
heano-0.9.0 tqdm-4.11.2
#
# To activate this environment, use:
# > source activate env2
#
# To deactivate this environment, use:
# > source deactivate env2
#

ubuntu@ubuntu-TITAN:~/github/chapter11$
```

```
(env2) ubuntu@ubuntu-TITAN:~/github/chapter11$ python
Python 2.7.13 | packaged by conda-forge | (default, May  2 2017, 12:48:11)
[GCC 4.8.2 20140120 (Red Hat 4.8.2-15)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import tensorflow as tf
>>> tf.Session()
2017-07-18 00:35:14.578038: W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow lib
rary wasn't compiled to use SSE4.1 instructions, but these are available on your machine and could
 speed up CPU computations.
2017-07-18 00:35:14.578051: W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow lib
rary wasn't compiled to use SSE4.2 instructions, but these are available on your machine and could
 speed up CPU computations.
2017-07-18 00:35:14.578055: W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow lib
rary wasn't compiled to use AVX instructions, but these are available on your machine and could sp
eed up CPU computations.
2017-07-18 00:35:14.578058: W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow lib
rary wasn't compiled to use AVX2 instructions, but these are available on your machine and could s
peed up CPU computations.
2017-07-18 00:35:14.578061: W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow lib
rary wasn't compiled to use FMA instructions, but these are available on your machine and could sp
eed up CPU computations.
2017-07-18 00:35:14.710152: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:893] successful
 NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so r
eturning NUMA node zero
2017-07-18 00:35:14.710550: I tensorflow/core/common_runtime/gpu/gpu_device.cc:940] Found device 0
 with properties:
name: GeForce GTX TITAN X
major: 5 minor: 2 memoryClockRate (GHz) 1.076
pciBusID 0000:01:00.0
Total memory: 11.92GiB
Free memory: 11.27GiB
2017-07-18 00:35:14.710563: I tensorflow/core/common_runtime/gpu/gpu_device.cc:961] DMA: 0
2017-07-18 00:35:14.710567: I tensorflow/core/common_runtime/gpu/gpu_device.cc:971] 0:   Y
2017-07-18 00:35:14.710586: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1030] Creating Tens
orFlow device (/gpu:0) -> (device: 0, name: GeForce GTX TITAN X, pci bus id: 0000:01:00.0)
<tensorflow.python.client.session.Session object at 0x7fd15b3e4110>
>>>
```

```yaml
name: env3
channels:
    - https://conda.anaconda.org/menpo
    - conda-forge
dependencies:
    - python=3
    - opencv
    - numpy
    - matplotlib
    - jupyter
    - pillow
    - scikit-learn
    - scikit-image
    - scipy
    - h5py
    - eventlet
    - flask-socketio
    - seaborn
    - pandas
    - ffmpeg
    - pyqt
    - pip:
        - moviepy
        - tensorflow-gpu
        - keras
        - prettytable
~
~
~
"env3.yml" 26L, 416C                                    4,1           All
```

NVIDIA ACCELERATED COMPUTING Downloads Training Ecosystem Forums

CUDA Toolkit Download

Home > ComputeWorks > CUDA Toolkit > CUDA Toolkit Download

Select Target Platform

Click on the green buttons that describe your target platform. Only supported platforms will be shown.

Operating System	Windows, Linux, Mac OSX
Architecture	x86_64, ppc64le
Distribution	Fedora, OpenSUSE, RHEL, CentOS, SLES, Ubuntu
Version	16.04, 14.04
Installer Type	runfile (local), deb (local), deb (network), cluster (local)

Download Installers for Linux Ubuntu 16.04 x86_64

The base installer is available for download below.
There is 1 patch available. This patch requires the base installer to be installed first.

Base Installer

Download (1.4 GB)

Installation Instructions:

1. Run `sudo sh cuda_8.0.61_375.26_linux.run`
2. Follow the command-line prompts

Patch 2 (Released Jun 26, 2017)

Download (95.3 MB)

cuBLAS Patch Update to CUDA 8: Includes performance enhancements and bug-fixes

The CUDA Toolkit contains Open-Source Software. The source code can be found here.
The checksums for the installer and patches can be found in Installer Checksums.
For further information, see the Installation Guide for Linux and the CUDA Quick Start Guide.