TensorFlow GPU
@AWS

Sung Kim <hunkim+ml@gmail.com>
http://hunkim.github.io/ml/
Deep Network

- Takes a long time for training
  - Many forward/backward propagation and weight updates
  - Many metrics multiplications

- Very quick for testing and use in practice
  - One simple forward propagation
A graphics processing unit (GPU), also occasionally called visual processing unit (VPU), is a specialized electronic circuit designed to rapidly manipulate and alter memory to accelerate the creation of images in a frame buffer intended for output to a display.
GPU version

# Ubuntu/Linux 64-bit, GPU enabled, Python 2.7
# Requires CUDA toolkit 7.5 and CuDNN v4. For other versions, see "Install from sources" below.
$ export TF_BINARY_URL=https://storage.googleapis.com/tensorflow/linux/gpu/tensorflow-0.9.0rc0-cp27-none-linux_x86_64.whl

# Python 2
$ sudo pip install --upgrade $TF_BINARY_URL
http://solutionsreview.com/cloud-platforms/microsoft-beats-aws-google-on-cloud-storage-benchmark-test/
AWS GPU price in Oregon

<table>
<thead>
<tr>
<th>GPU Instances - Current Generation</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>g2.2xlarge</td>
<td>8</td>
<td>26</td>
<td>15</td>
<td>60</td>
<td>60 SSD</td>
<td>$0.65 per Hour</td>
</tr>
<tr>
<td>g2.8xlarge</td>
<td>32</td>
<td>104</td>
<td>60</td>
<td>2 x 120 SSD</td>
<td>$2.6 per Hour</td>
<td></td>
</tr>
</tbody>
</table>
EC2 Console: Oregon

Resources
You are using the following Amazon EC2 resources in the US West (Oregon) region:

- Running Instances: 0
- Dedicated Hosts: 0
- Volumes: 1
- Key Pairs: 2
- Placement Groups: 0
- Elastic IPs: 0
- Snapshots: 1
- Load Balancers: 0
- Security Groups: 13

Create Instance
To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

Launch Instance

Note: Your instances will launch in the US West (Oregon) region.
ubuntu,
GPU,
12G or more
key to access the server

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.

Choose an existing key pair

**Select a key pair**

hunkim-oregon

I acknowledge that I have access to the selected private key file (hunkim-oregon.pem), and that without this file, I won’t be able to log into my instance.
EC2: Create an instance
It's ready to ssh!

<table>
<thead>
<tr>
<th>Instance ID</th>
<th>Instance Type</th>
<th>Availability Zone</th>
<th>Instance State</th>
<th>Status Checks</th>
<th>Alarm Status</th>
<th>Public DNS</th>
</tr>
</thead>
<tbody>
<tr>
<td>i-068c65db</td>
<td>g2.2xlarge</td>
<td>us-west-2c</td>
<td>running</td>
<td></td>
<td>Initializing</td>
<td>ec2-54-186-153-9.us-w...</td>
</tr>
</tbody>
</table>

Instance: i-068c65db  

- **Description**
  - Instance ID: i-068c65db
  - Instance state: running
  - Instance type: g2.2xlarge
  - Private DNS: ip-172.31.16.155.us-west-2.compute.internal

- **Public DNS**: ec2-54-186-153-9.us-west-2.compute.amazonaws.com
- **Public IP**: 54.186.153.9
- **Elastic IP**: -
- **Availability zone**: us-west-2c
Requires CUDA and CuDNN

# Ubuntu/Linux 64-bit, GPU enabled, Python 2.7
# Requires **CUDA toolkit 7.5 and CuDNN v4**. For other versions, see "Install from sources" below.
$ export TF_BINARY_URL=https://storage.googleapis.com/tensorflow/linux/gpu/tensorflow-0.9.0rc0-cp27-none-linux_x86_64.whl

# Python 2
$ sudo pip install --upgrade $TF_BINARY_URL
CUDA 7.5 on AWS GPU Instance Running Ubuntu 14.04

Launch stock Ubuntu AMI

- Launch ami-d05ec75bd
- Choose a GPU instance type: g2.2xlarge or g2.8xlarge
- Increase the size of the storage (this depends on what else you plan to install, I’d suggest at least 20 GB)

SSH in

```
$ ssh ubuntu@<instance ip>
```
cuDNN

Optional: cuDNN

One can apply for the developer program here https://developer.nvidia.com/cudnn. When approved, download cuDNN for Linux (either v4 RC or v3 is fine), upload the cuDNN package from the local computer to the instance, and install cuDNN:

```
tar -zxvf cudnn-7.0-linux-x64-v4.0-rc.tgz # or cudnn-7.0-linux-x64-v3.0
cd cuda
sudo cp lib64/* /usr/local/cuda/lib64/
sudo cp include/cudnn.h /usr/local/cuda/include/
```

https://no2147483647.wordpress.com/2016/01/16/setup-amazon-aws-gpu-instance-with-mxnet/
wget http://developer.download.nvidia.com/…/cuda-repo-ubuntu1404…
sudo dpkg -i cuda-repo-ubuntu1404_7.5-18_amd64.deb
sudo apt-get update
sudo apt-get upgrade -y
sudo apt-get install -y opencl-headers build-essential protobuf-compiler libprotoc-dev libboost-all-dev libleveldb-dev hdf5-tools libhdf5-serial-dev libopencv-core-dev libopencv-highgui-dev libsnappy-dev libsnappy1 libatlas-base-dev cmake libstdc++6-4.8-dbg libgoogle-glog0 libgoogle-glog-dev libgflags-dev liblmdb-dev git python-pip gfortran
sudo apt-get clean
sudo apt-get install -y linux-image-extra-`uname -r` linux-headers-`uname -r` linux-image-`uname -r`
sudo apt-get install -y cuda
nvidia-smi
sudo apt-get install python-pip python-dev
sudo pip install --upgrade https://storage.googleapis.com/…/tensorflow-0.8.0rc0-cp27-n…
git clone https://github.com/nlintz/TensorFlow-Tutorials
cd TensorFlow-Tutorials/
vi ~/.profile # add PATH, LD PATH
source ~/.profile
python 06_autoencoder.py
Add Path

- export PATH=/usr/local/cuda/bin:$PATH
- export LD_LIBRARY_PATH=/usr/local/cuda/lib64:$LD_LIBRARY_PATH
Requires CUDA and CuDNN

# Ubuntu/Linux 64-bit, GPU enabled, Python 2.7
# Requires **CUDA toolkit 7.5 and CuDNN v4**. For other versions, see "Install from sources" below.
$ export TF_BINARY_URL=https://storage.googleapis.com/tensorflow/linux/gpu/tensorflow-0.9.0rc0-cp27-none-linux_x86_64.whl

# Python 2
$ sudo pip install --upgrade $TF_BINARY_URL
Reuse ami-9e39dcf3 (N. Virginia) 
ami-38f60658 (oregon)

Perhaps, it will be unavailable when later CUDA versions are out.
Creating TensorFlow device (/gpu:0)

05] successfully opened CUDA library libcusolver.so locally
9] Couldn't open CUDA library libcudnn.so. LD_LIBRARY_PATH: /usr/local/cuda/lib64:
0c:1562] Unable to load cuDNN DSO
05] successfully opened CUDA library libcufft.so locally
05] successfully opened CUDA library libcudade.so.1 locally
05] successfully opened CUDA library libcurand.so locally
-ubyte.gz')
cationWarning: converting an array with ndim > 0 to an index will result in an error in the
ej
.py:42: VisibleDeprecationWarning: converting an array with ndim > 0 to an index will result
, 1)
-ubyte.gz')
ubyte.gz'
ubyte.gz')
executor.cc:900] successful NUMA node read from SysFS had negative value (-1), but there must
t.cc:102] Found device 0 with properties:

7

t.cc:126] DMA: 0
t.cc:136] 0: Y
ice.cc:755] Creating TensorFlow device (/gpu:0) -> (device: 0, name: GRID K520, pci bus id:
locator.cc:244] PoolAllocator: After 1704 get requests, put_count=1321 evicted_count=1000 ev
locator.cc:256] Raising pool_size_limit_ from 100 to 110
locator.cc:244] PoolAllocator: After 1704 get requests, put_count=1812 evicted_count=1000 ev
locator.cc:256] Raising pool_size_limit_ from 256 to 281
model saved to ./save/model.ckpt

1/6750 (epoch 0), train_loss = 11.127, time/batch = 4.687
2/6750 (epoch 0), train_loss = 11.176, time/batch = 4.430
3/6750 (epoch 0), train_loss = 10.946, time/batch = 4.604
4/6750 (epoch 0), train_loss = 10.616, time/batch = 4.483
5/6750 (epoch 0), train_loss = 10.287, time/batch = 5.334
6/6750 (epoch 0), train_loss = 9.723, time/batch = 5.674
7/6750 (epoch 0), train_loss = 9.449, time/batch = 5.972
8/6750 (epoch 0), train_loss = 9.489, time/batch = 5.785
9/6750 (epoch 0), train_loss = 8.861, time/batch = 5.518
10/6750 (epoch 0), train_loss = 8.797, time/batch = 5.856
11/6750 (epoch 0), train_loss = 8.688, time/batch = 5.795
12/6750 (epoch 0), train_loss = 8.814, time/batch = 5.803
13/6750 (epoch 0), train_loss = 8.728, time/batch = 5.595
14/6750 (epoch 0), train_loss = 8.928, time/batch = 5.586
15/6750 (epoch 0), train_loss = 8.657, time/batch = 5.529
16/6750 (epoch 0), train_loss = 8.736, time/batch = 5.536
17/6750 (epoch 0), train_loss = 8.643, time/batch = 5.690
18/6750 (epoch 0), train_loss = 8.985, time/batch = 5.828
19/6750 (epoch 0), train_loss = 8.792, time/batch = 5.584
20/6750 (epoch 0), train_loss = 8.882, time/batch = 5.555
21/6750 (epoch 0), train_loss = 8.798, time/batch = 5.437
22/6750 (epoch 0), train_loss = 8.872, time/batch = 5.597

OS X El Capitan
Version 10.11.3
MacBook Pro (Retina, 13-inch, Early 2015)
Processor 3.1 GHz Intel Core i7
Memory 16 GB 1867 MHz DDR3
AWS GPU price in Oregon

<table>
<thead>
<tr>
<th>GPU Instances - Current Generation</th>
</tr>
</thead>
<tbody>
<tr>
<td>g2.2xlarge</td>
</tr>
<tr>
<td>g2.8xlarge</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Amazon Elastic Compute Cloud running Linux/UNIX</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0.650 per On Demand Linux g2.2xlarge Instance Hour</td>
</tr>
<tr>
<td>$2.6 per On Demand Linux g2.8xlarge Instance Hour</td>
</tr>
<tr>
<td>Total:</td>
</tr>
</tbody>
</table>

\[
2.6 \times 24 \times 30 = 1,872 \text{ USD}
\]
Spot instances

You do not have any Spot instance requests in this region.

Click on the "Request Spot Instances" button to request your first Spot instance.
Price bidding

### Step 3: Configure Instance Details
Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

<table>
<thead>
<tr>
<th>Number of instances</th>
<th>1</th>
<th>Launch into Auto Scaling Group</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Purchasing option</strong></td>
<td>[ ] Request Spot instances</td>
<td></td>
</tr>
<tr>
<td><strong>Current price</strong></td>
<td>us-east-1a 0.2428</td>
<td></td>
</tr>
<tr>
<td></td>
<td>us-east-1c 0.1442</td>
<td></td>
</tr>
<tr>
<td></td>
<td>us-east-1d 0.153</td>
<td></td>
</tr>
<tr>
<td></td>
<td>us-east-1e 6.500</td>
<td></td>
</tr>
<tr>
<td><strong>Maximum price</strong></td>
<td>$ (e.g. 0.045 = 4.5 cents/hour)</td>
<td></td>
</tr>
<tr>
<td><strong>Launch group</strong></td>
<td>(Optional)</td>
<td></td>
</tr>
</tbody>
</table>
bill, bill, bill!
Check, stop, and terminate
Cloud Watch

http://docs.aws.amazon.com/AmazonCloudWatch/latest/DeveloperGuide/UsingAlarmActions.html#AddingStopActions
Stop when CPU utilization <= 0.3
Shutdown after training

$ screen

$ sudo -i

# python train.py; shutdown -h now
Deep learning for Everyone
Season 2 coming soon!

Sung Kim <hunkim+ml@gmail.com>
http://hunkim.github.io/ml/