Accelerating Deep Learning

Rajesh K Jeyapaul,
Architect and Advocate ,
IBM

Agenda and Take Away

- **Agenda**
  - Understand the need for high computing environment
  - Maths intensive convolution
  - Developers approach to GPU based environment
  - Performance results
- **Take Away**
  - Now I know, how to train my model in GPU environment
  - Code snippet to try it out on the GPU system
  - Contributing to future IBM meetups

Deep Learning

- Image processing
  - Identifying Objects, Characters…
- Natural Language processing
  - Sentiment Analysis , Machine translation of languages ..
- Audio /Speech
  - Identifying spoken digits in audio sample
  - Speaker Identification
  - Music MIR (Music Information Retrieval)

Music Information Retrieval : Training time –> 2+hr

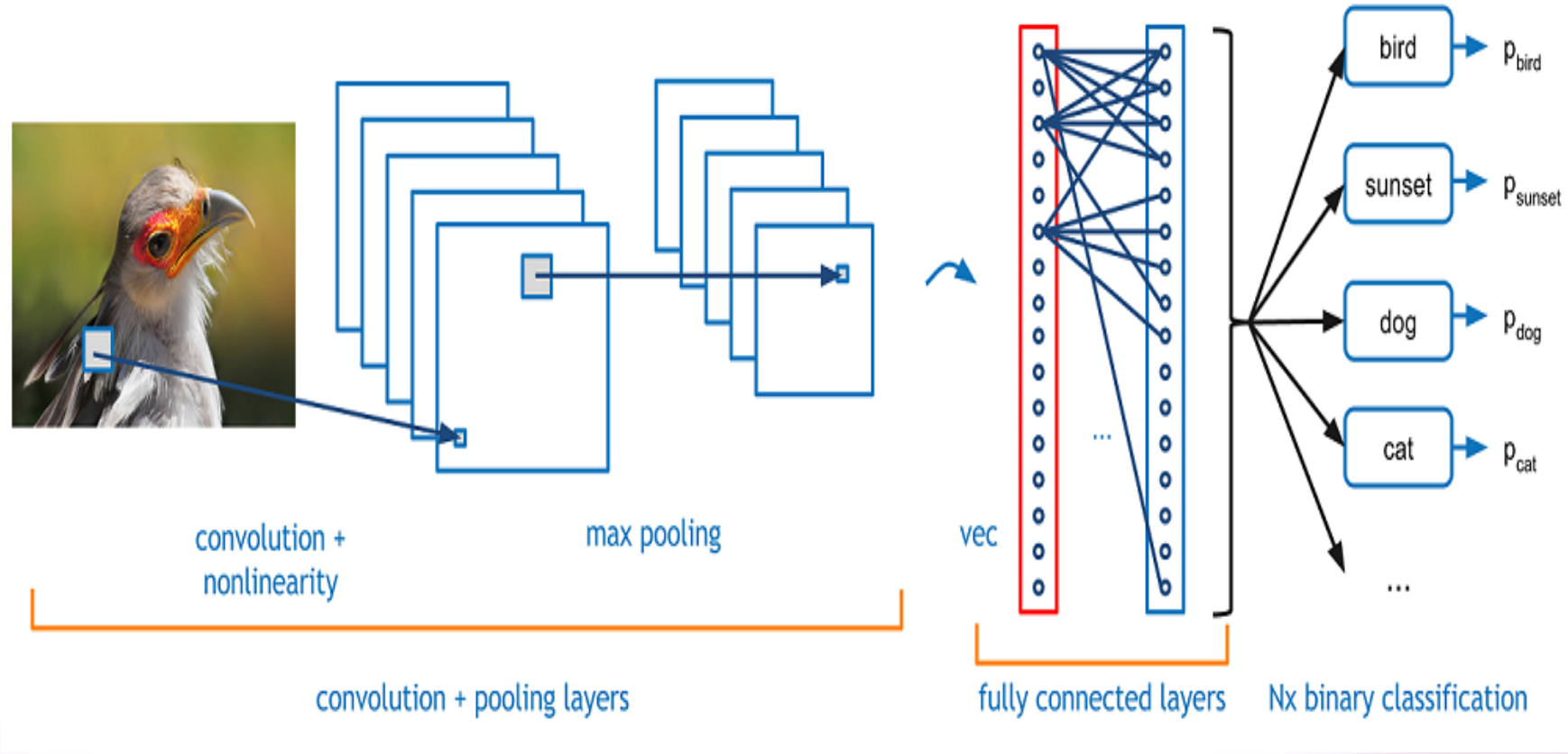
- Free Music Archive (FMA)
  - Size – 1000GB
  - Content – 100k audio tracks , 16k artists , 15k album , 161 genres
  - Challenge – Music information Retrievel (MIR)
  - Evaluation metric – Log Loss and F1 score
  - Approaches used – ConvNet on spectrograms , Deep Neural net , ExtraTrees , XGBoost
  - Classes – Blues, Classical , Country , Folk , Instrumental , Jazz , Rock , Gospel..

```
python features.py
```

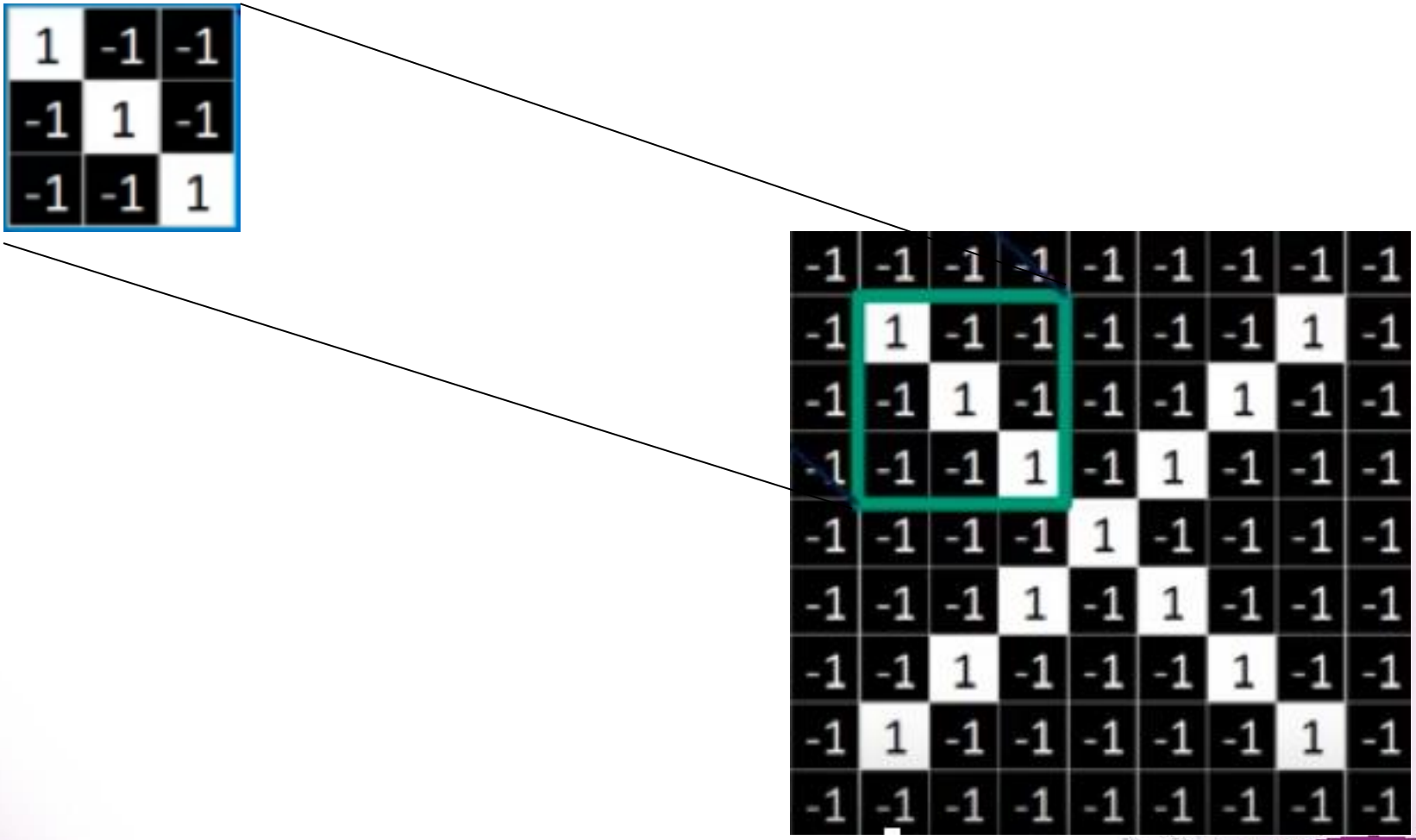
Note that this script can take many hours to complete on the whole 60k tracks. For you to play with the da
you'll find those features pre-computed on the challenge's dataset page.

  - Reference - https://arxiv.org/pdf/1803.05337.pdf
  - https://github.com/crowdAI/crowdai-musical-genre-recognition-starter-kit

Maths behind Convolutional Neural Network (CNN)

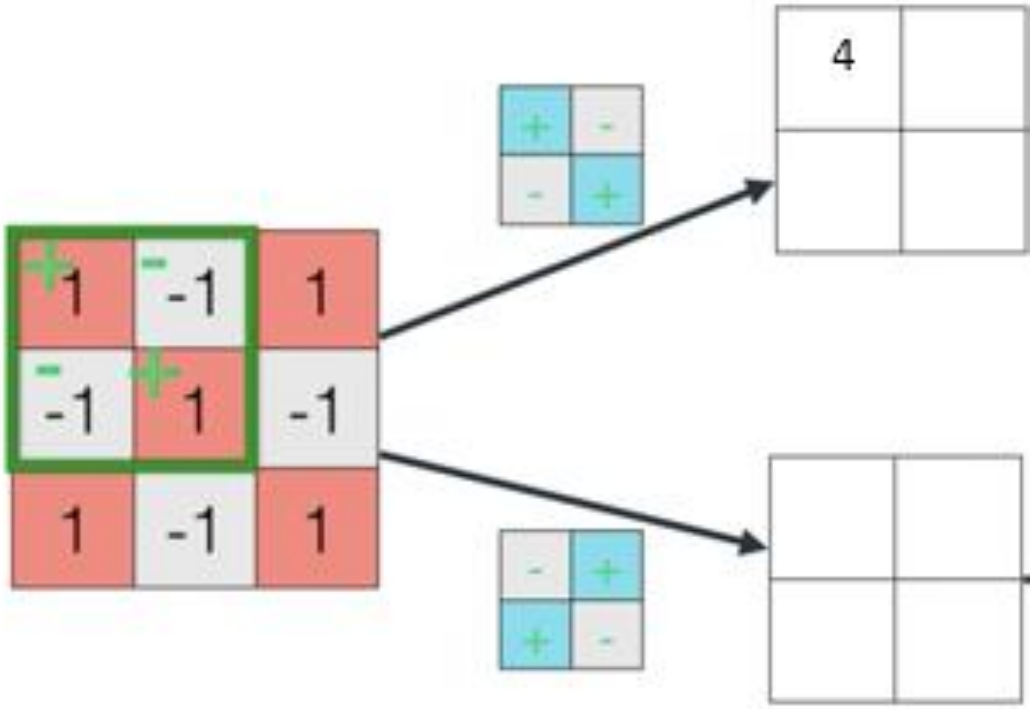


Extract the feature of “X” using CNN



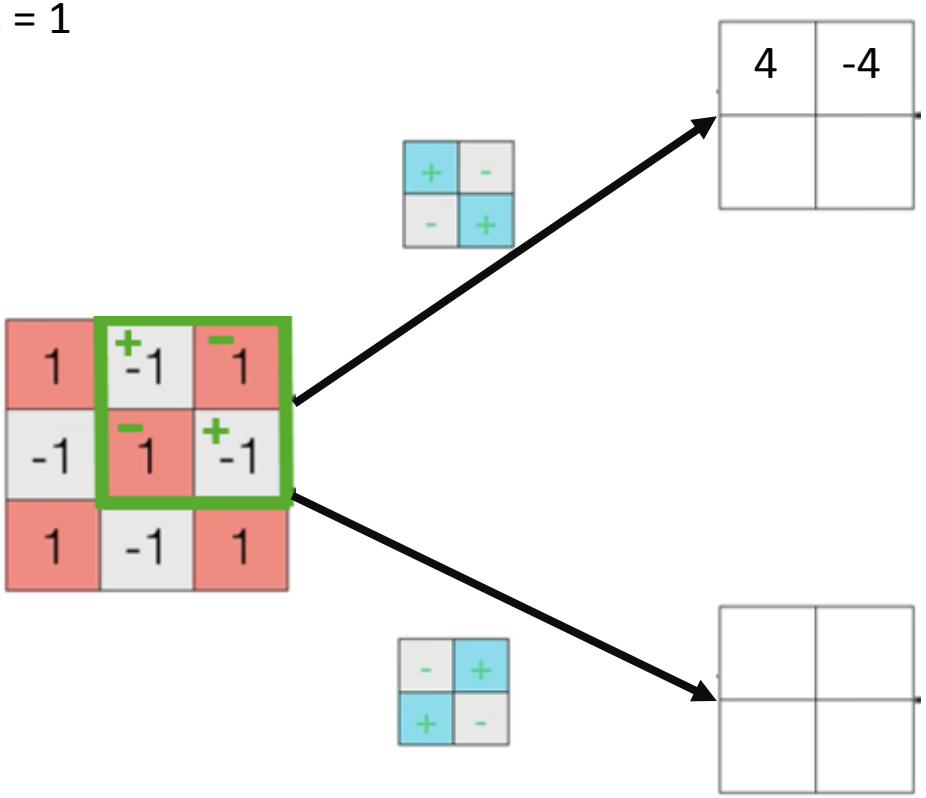
Convolution - Filters

Kernel (filter) size = 2x2

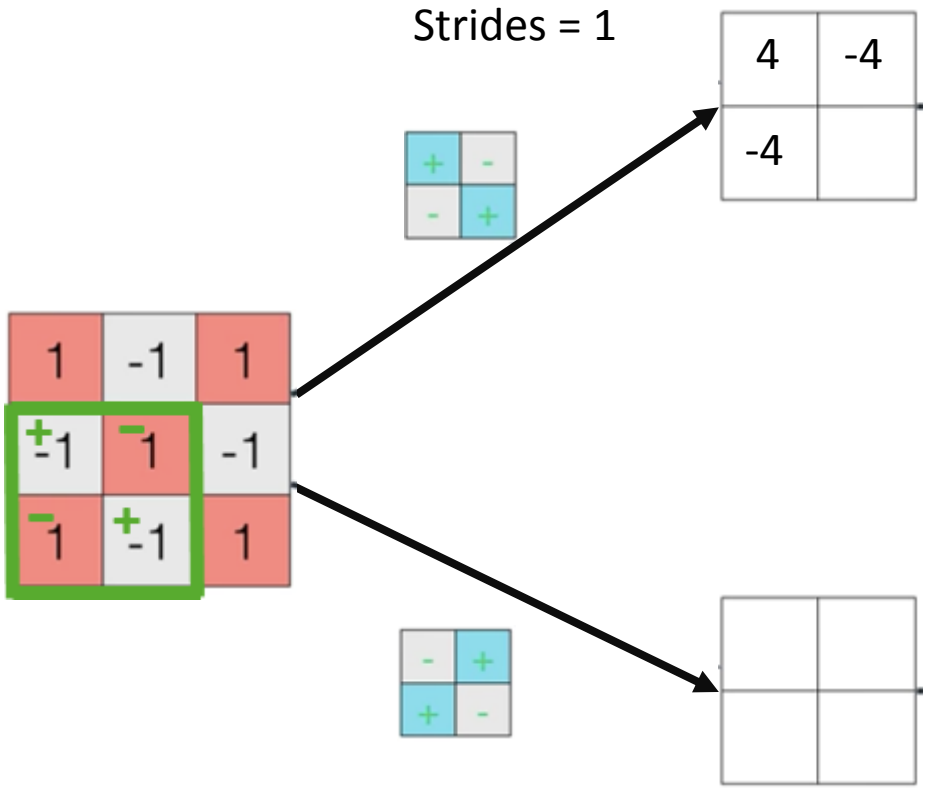


Convolution - Filters

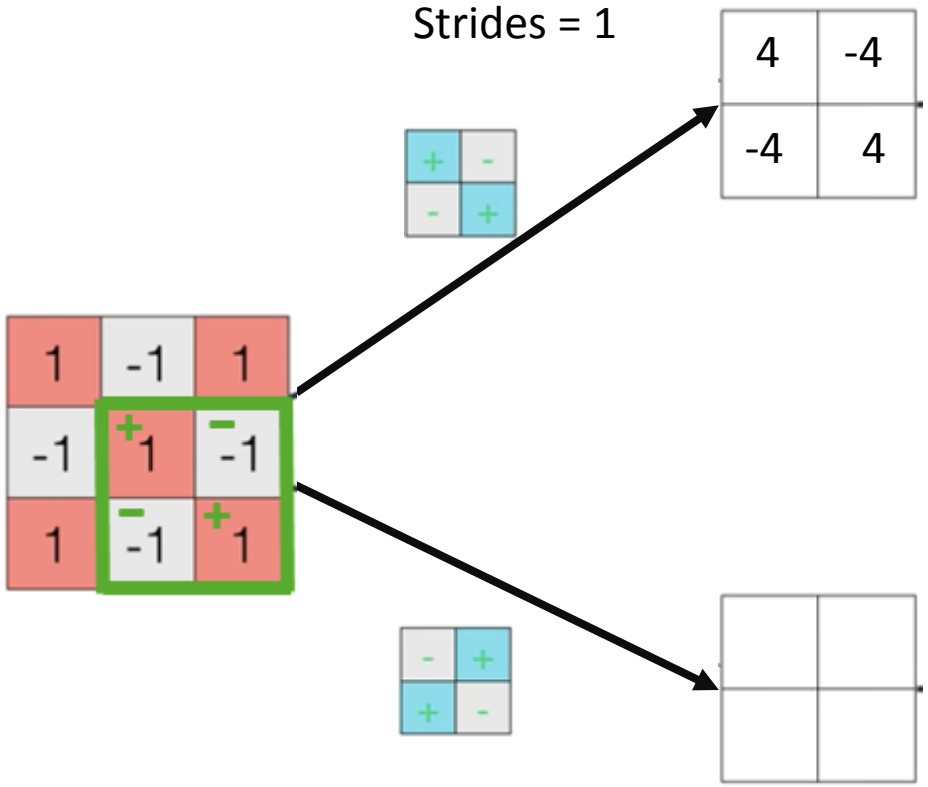
Strides = 1



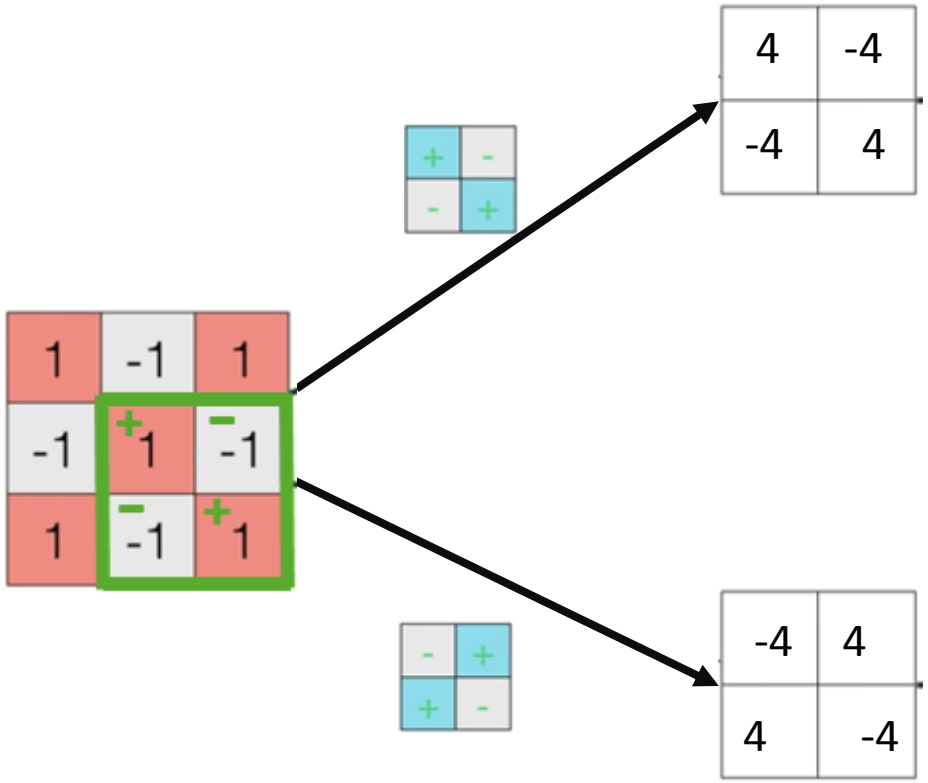
Convolution - Filters



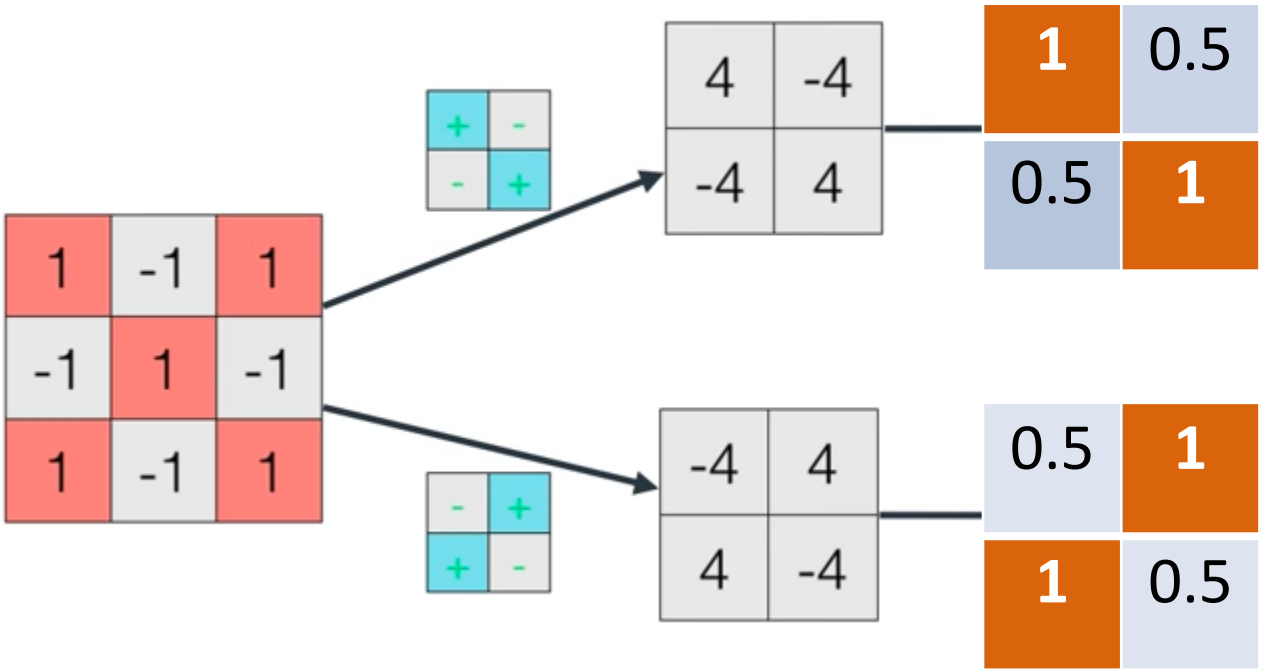
Convolution - Filters



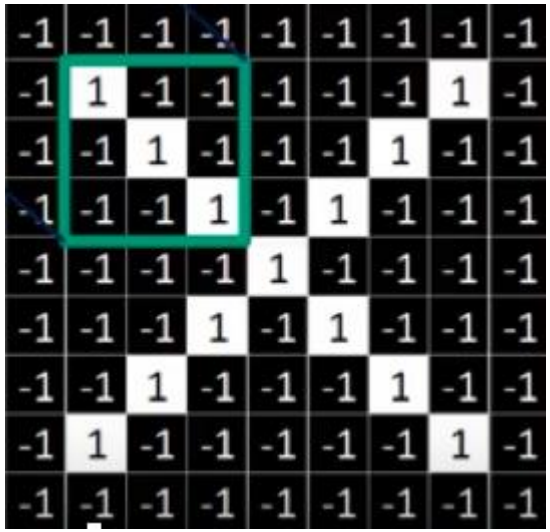
Convolution - Filters



Convolution - Normalization



Multi Layer CNN



Convolution

ReLU

Pooling

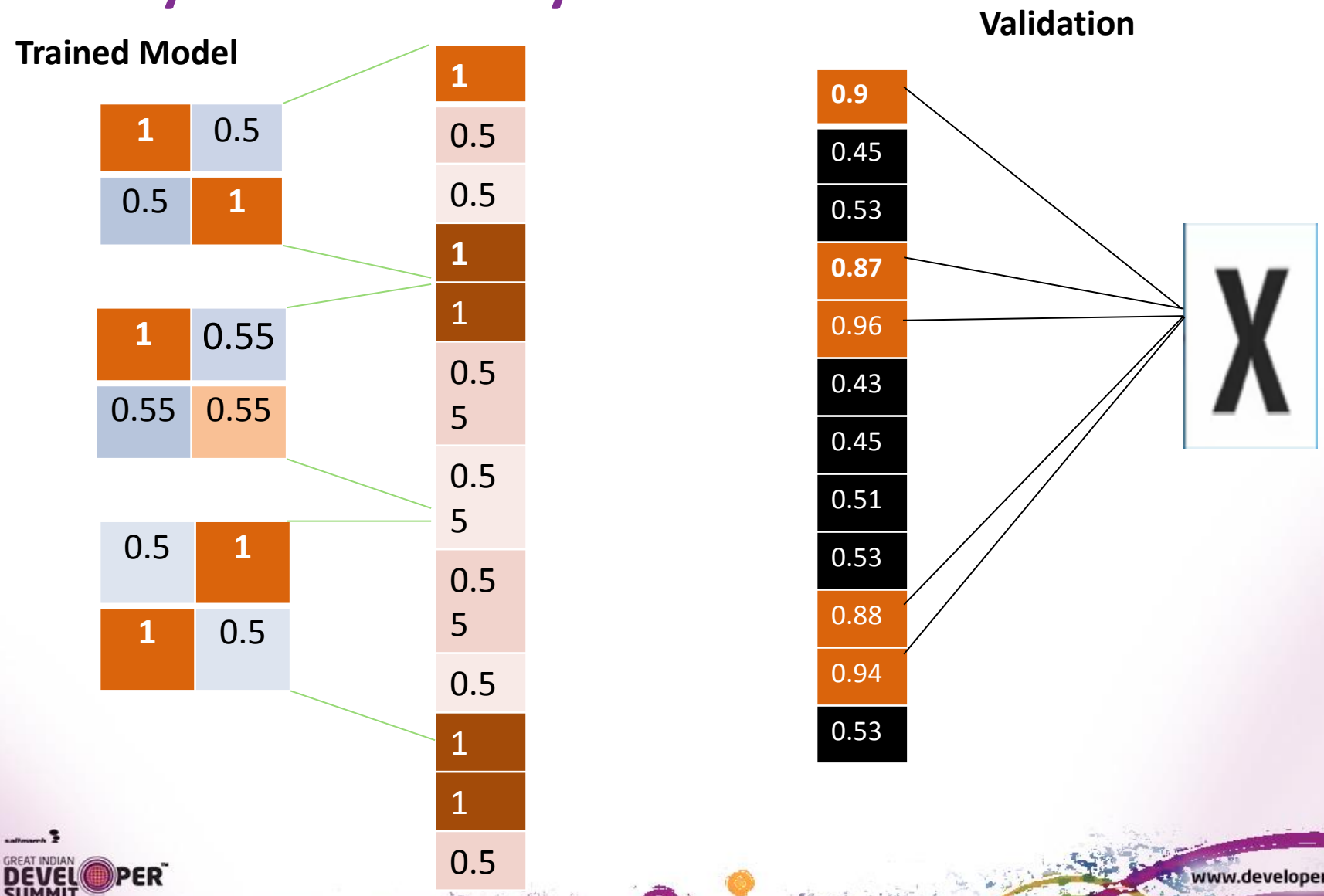
X "N"

1	0.5
0.5	1

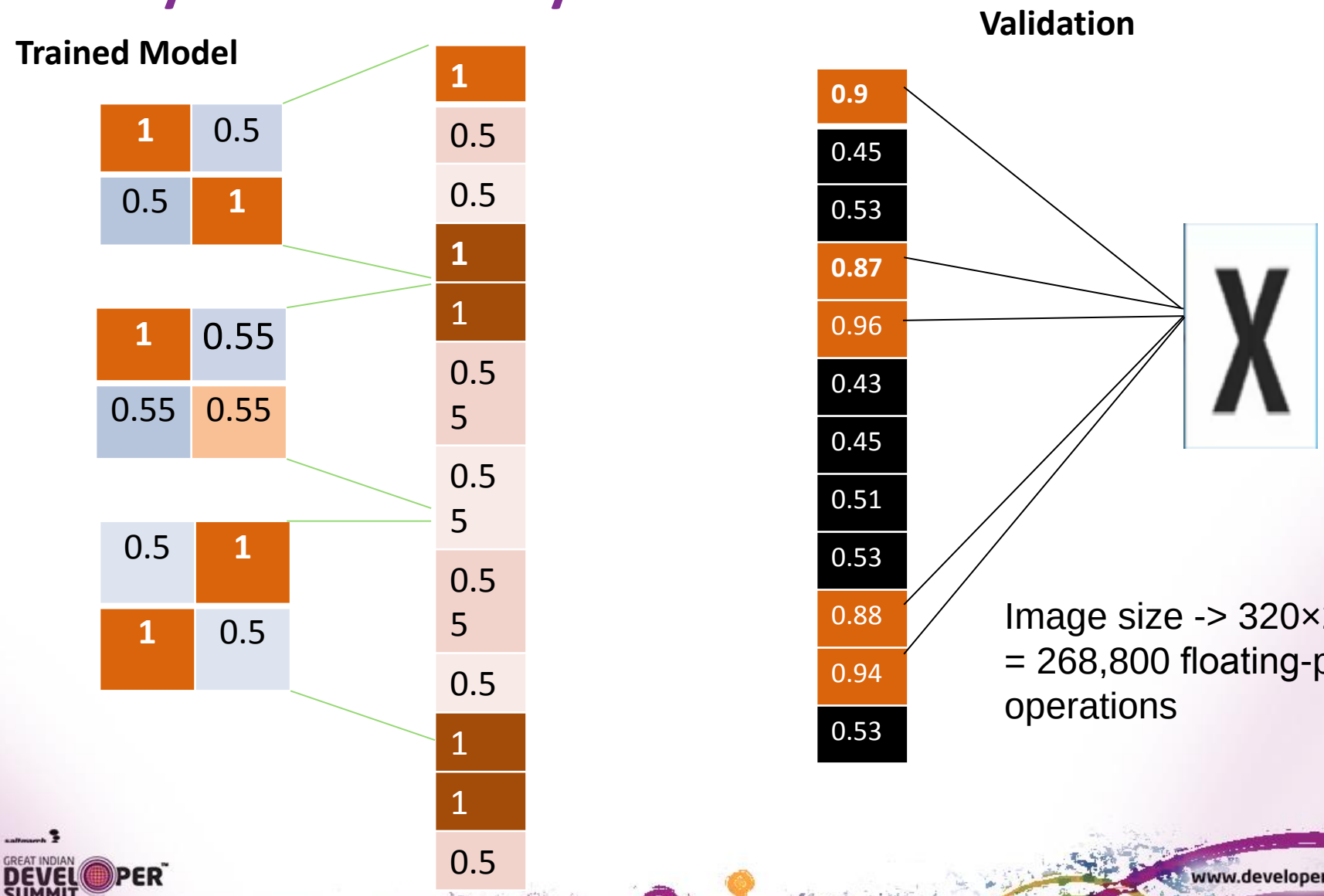
1	0.55
0.55	0.55

0.5	1
1	0.5

Fully Connected Layer



Fully Connected Layer



Need for High compute environment

Solving problem for a Traffic police department

- Scenario 1
  - To identify whether a vehicle is crossing the stop line or not
- Scenario 2
  - To capture the vehicle number
- Scenario 3
  - To detect whether the car driver is wearing the seat belt or not
- Scenario 4
  - To identify the driver of the vehicle

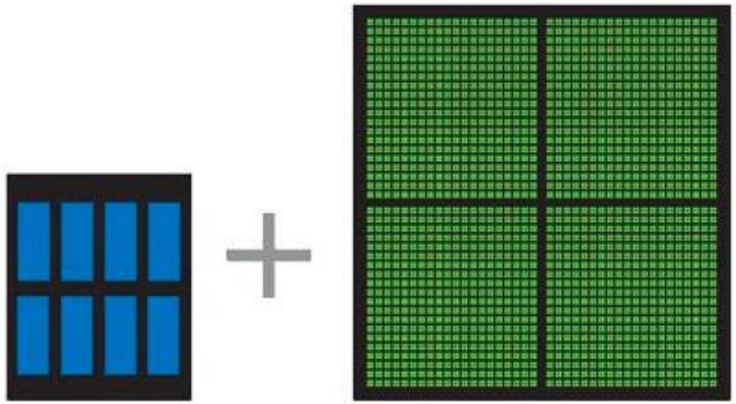
Which Scenario is feature rich ?

How to accelerate training

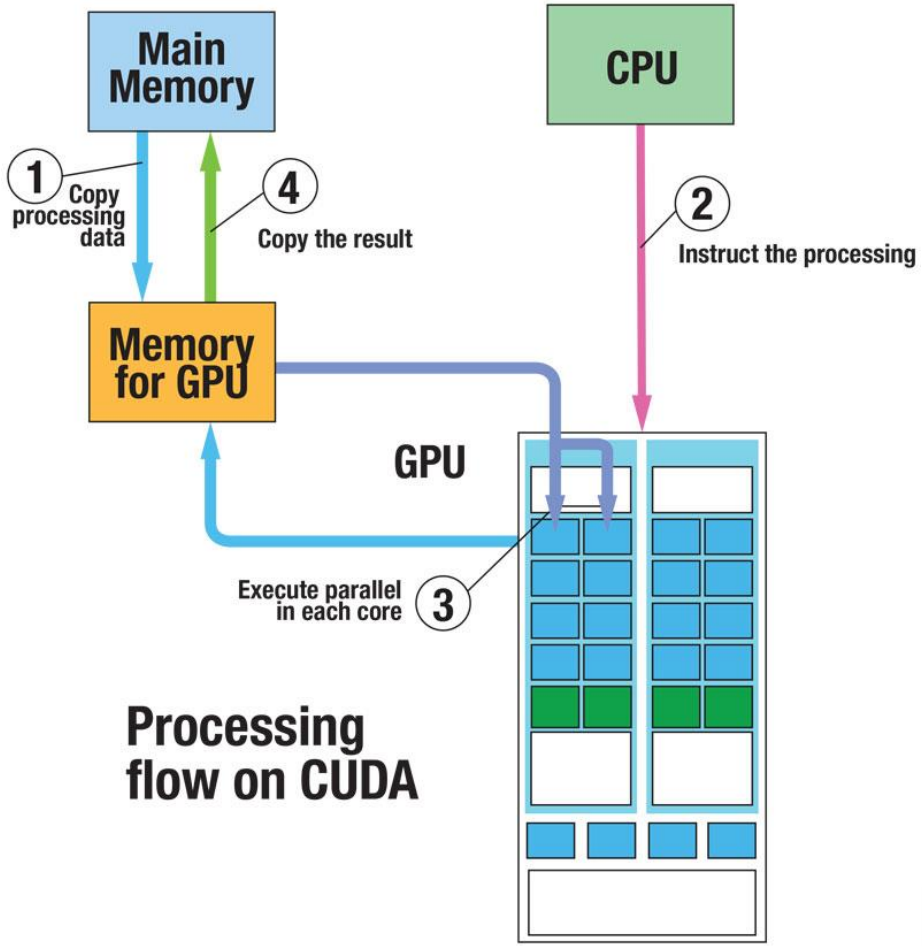
- Memory and flops computation intensive
- Size of Data Vs Depth of Model network
  - Optimal Size of the data
  - Optimal strides , kernel , batch size, epoch, model depth etc.,
  - Parallel programming environment with multiple threads – GPU / CUDA
  - Transfer Learning…

Benefits of GPU

- GPU offers capabilities for parallelism, good for operations such as large scale hashing and matrix calculations – foundations of ML workloads
- CUDA (from NVIDIA) – provides massive parallel architecture used for numerical computation



CUDA (compute unified device arch)



CUDA and Python / C++. – Parallelizing the kernel functions

- CUDA Python
  - *from numbapro import vectorize*
  - *@vectorize(['float32(float32, float32)'], **target='cuda')***
  - *def pow(a, b):*
  - *return a ** b*
  - **" @decorator "**
- **C / C++**
  - **malloc() -> cudamallocmanaged()**
  - **free() -> cudafree()**
  - **Functionname <<< Thread block >>> (). < kernel starts >**
  - **CudaDeviceSynchronize()**
  - **Function definition add "_global"**

CUDA Driver API

```
/* Allocate memory for Filter and ImageBatch, fill with data */
cudaMalloc( &ImageInBatch , ... );
cudaMalloc( &Filter , ... );
...

/* Set descriptors */
cudnnSetTensor4dDescriptor( InputDesc, CUDNN_TENSOR_NCHW, 128, 96, 221, 221);
cudnnSetFilterDescriptor( FilterDesc, 256, 96, 7, 7 );
cudnnSetConvolutionDescriptor( convDesc, InputDesc, FilterDesc,
    pad_x, pad_y, 2, 2, 1, 1, CUDNN_CONVOLUTION);

/* query output layout */
cudnnGetOutputTensor4dDim(convDesc, CUDNN_CONVOLUTION_FWD, &n_out, &c_out, &h_out, &w_out);

/* Set and allocate output tensor descriptor */
cudnnSetTensor4dDescriptor( &OutputDesc, CUDNN_TENSOR_NCHW, n_out, c_out, h_out, w_out);
cudaMalloc(&ImageBatchOut, n_out * c_out * h_out * w_out * sizeof(float));

/* launch convolution on GPU */
cudnnConvolutionForward( handle, InputDesc, ImageInBatch, FilterDesc, Filter, convDesc,
    OutputDesc, ImageBatchOut, CUDNN_RESULT_NO_ACCUMULATE);
```

Configuring GPU / CUDA environment for DL

- Keras , TensorFlow , etc., provides high level api to effectively utilize the GPU/CUDA environment
- Pre-req
  - **$ lsb_release -a**
    - Description: Ubuntu 16.04.2 LTS
    - Release: **16.04**
  - **uname -r**
    - 4.4.0-79-generic
  - **Python 2.7.12**
  - **NVIDIA requirements** – CUDA Toolkit , NVIDIA Drivers , GPU card with CUDA compute capability
  - **$lscpi | grep –i nvidia**
    - *0002:04:00.0 3D controller: NVIDIA Corporation GK210GL [Tesla K80] (rev a1) …*
  - **pip install tensorflow-gpu**

How to verify Tensorflow is running with GPU

```
from tensorflow.python.client import device_lib

print(device_lib.list_local_devices())
```

```
Using TensorFlow backend.
2018-04-24 09:25:28.554088: I tensorflow/core/cc
name: Tesla K80
major: 3 minor: 7 memoryClockRate (GHz) 0.8235
pciBusID 0000:03:00.0
Total memory: 11.17GiB
Free memory: 10.77GiB
2018-04-24 09:25:28.613508: W tensorflow/stream_
s with that.
2018-04-24 09:25:28.615216: I tensorflow/core/cc
name: Tesla K80
major: 3 minor: 7 memoryClockRate (GHz) 0.8235
pciBusID 0000:04:00.0
Total memory: 11.17GiB
Free memory: 11.11GiB
```

```
[name: "/cpu:0"
device_type: "CPU"
memory_limit: 268435456
locality {
}
incarnation: 14839384038527459444
, name: "/gpu:0"
device_type: "GPU"
memory_limit: 10981838029
locality {
  bus_id: 1
}
incarnation: 7336279740307820125
physical_device_desc: "device: 0, name: Tesla K80, pci bus id: 0000:03:00.0"
, name: "/gpu:1"
device_type: "GPU"
memory_limit: 11331983770
locality {
  bus_id: 1
}
```

How to find Keras is using GPU ?

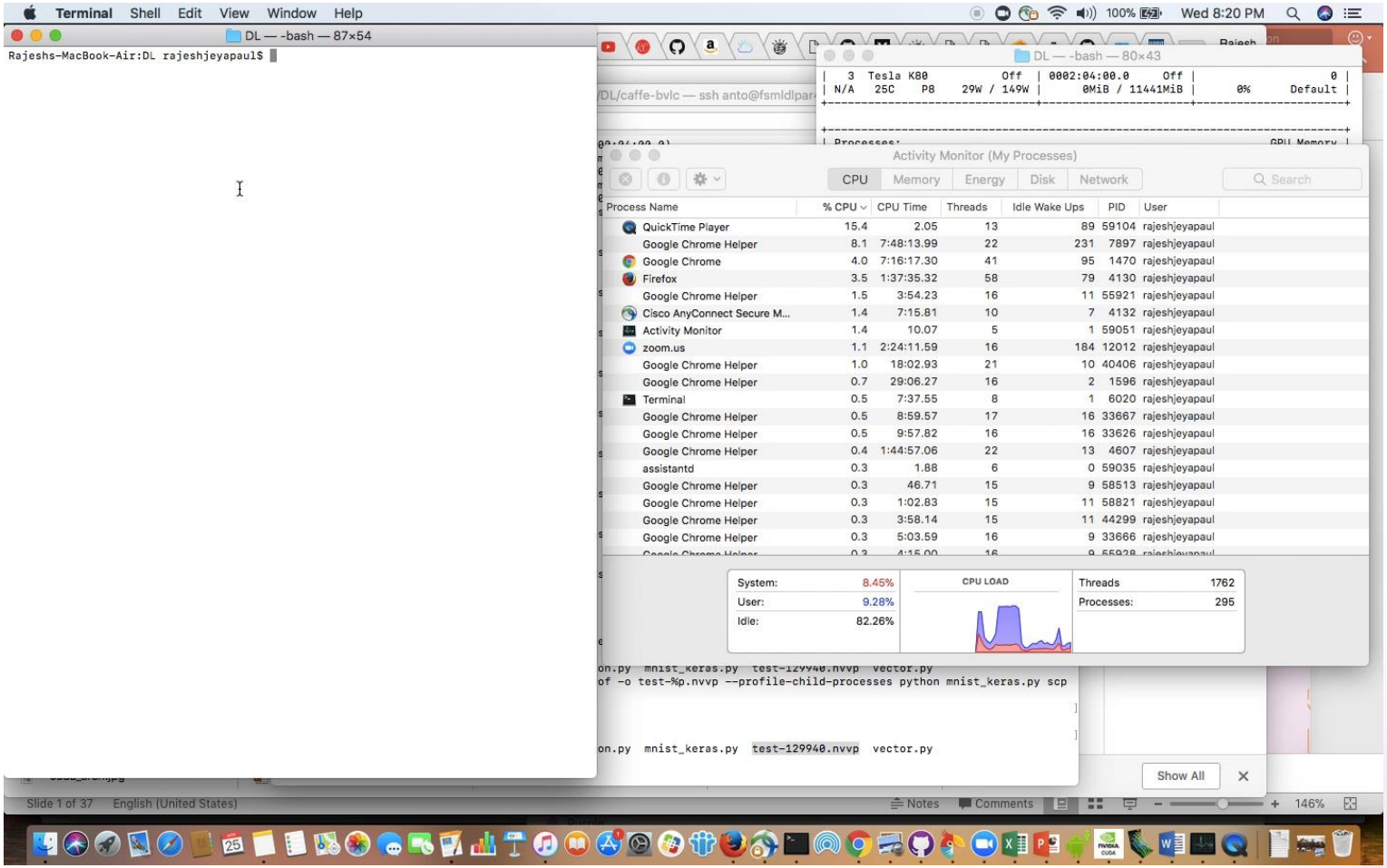
```
import keras
from keras import backend as K
K.tensorflow_backend._get_available_gpus()
```

```
Using TensorFlow backend.
2018-04-25 04:03:22.91602
name: Tesla K80
major: 3 minor: 7 memoryC
pciBusID 0000:03:00.0
Total memory: 11.17GiB
Free memory: 10.77GiB
2018-04-25 04:03:22.97463
s with that.
2018-04-25 04:03:22.97634
name: Tesla K80
major: 3 minor: 7 memoryC
pciBusID 0000:04:00.0
Total memory: 11.17GiB
Free memory: 11.11GiB
ajor: 3 minor: 7 memoryClockRate (GHz) 0.8235
ciBusID 0002:04:00.0
otal memory: 11.17GiB
ree memory: 11.11GiB
2018-04-25 04:03:23.101397: I tensorflow/core/common_runtime/gpu/gpu_device.cc:832] Peer access not supported between device ordinals 0 and 2
2018-04-25 04:03:23.101412: I tensorflow/core/common_runtime/gpu/gpu_device.cc:832] Peer access not supported between device ordinals 0 and 3
2018-04-25 04:03:23.101445: I tensorflow/core/common_runtime/gpu/gpu_device.cc:832] Peer access not supported between device ordinals 1 and 2
2018-04-25 04:03:23.101457: I tensorflow/core/common_runtime/gpu/gpu_device.cc:832] Peer access not supported between device ordinals 1 and 3
2018-04-25 04:03:23.101468: I tensorflow/core/common_runtime/gpu/gpu_device.cc:832] Peer access not supported between device ordinals 2 and 0
2018-04-25 04:03:23.101478: I tensorflow/core/common_runtime/gpu/gpu_device.cc:832] Peer access not supported between device ordinals 2 and 1
2018-04-25 04:03:23.102118: I tensorflow/core/common_runtime/gpu/gpu_device.cc:832] Peer access not supported between device ordinals 3 and 0
2018-04-25 04:03:23.102131: I tensorflow/core/common_runtime/gpu/gpu_device.cc:832] Peer access not supported between device ordinals 3 and 1
2018-04-25 04:03:23.102204: I tensorflow/core/common_runtime/gpu/gpu_device.cc:961] DMA: 0 1 2 3
2018-04-25 04:03:23.102215: I tensorflow/core/common_runtime/gpu/gpu_device.cc:971] 0:   Y Y N N
2018-04-25 04:03:23.102224: I tensorflow/core/common_runtime/gpu/gpu_device.cc:971] 1:   Y Y N N
2018-04-25 04:03:23.102232: I tensorflow/core/common_runtime/gpu/gpu_device.cc:971] 2:   N N Y Y
2018-04-25 04:03:23.102241: I tensorflow/core/common_runtime/gpu/gpu_device.cc:971] 3:   N N Y Y
2018-04-25 04:03:23.102266: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1030] Creating TensorFlow device (/gpu:0) -> (device: 0, name: Tesla K80,
2018-04-25 04:03:23.102278: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1030] Creating TensorFlow device (/gpu:1) -> (device: 1, name: Tesla K80,
2018-04-25 04:03:23.102289: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1030] Creating TensorFlow device (/gpu:2) -> (device: 2, name: Tesla K80,
2018-04-25 04:03:23.102299: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1030] Creating TensorFlow device (/gpu:3) -> (device: 3, name: Tesla K80,
2018-04-25 04:03:23.436862: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1030] Creating TensorFlow device (/gpu:0) -> (device: 0, name: Tesla K80,
2018-04-25 04:03:23.436904: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1030] Creating TensorFlow device (/gpu:1) -> (device: 1, name: Tesla K80,
2018-04-25 04:03:23.436923: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1030] Creating TensorFlow device (/gpu:2) -> (device: 2, name: Tesla K80,
2018-04-25 04:03:23.436933: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1030] Creating TensorFlow device (/gpu:3) -> (device: 3, name: Tesla K80
```

Model - Simple Feed-Forward Network for MNIST

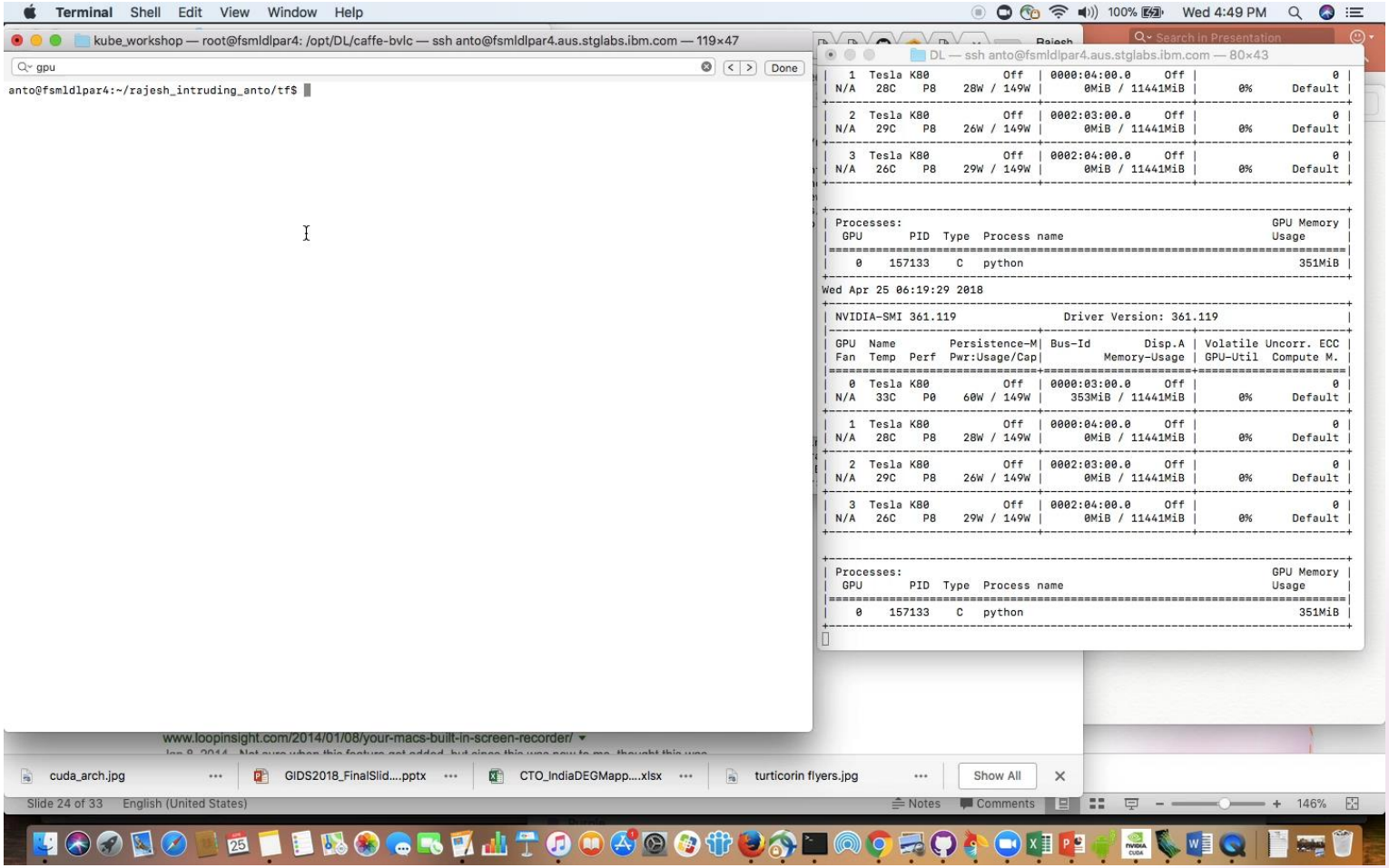
- Data size
  - Training data – 60K
  - Test data – 10k
  - Image size – 28 x 28
- Model
  - Sequential
  - Activation – Relu and softmax
  - Batch size – 128
  - Epcoh – 10
- Training Time
  - CPU – 51 secs
  - GPU – 28 secs

Performance Results - CPU



Performance Results - GPU

nvidia-smi --loop=2



Alex's CIFAR-10 tutorial, Caffe style

```
[root@aistation caffe-bvlc]# grep "Iteration" CPU.log | grep iters
I0219 11:36:49.950510 32988 solver.cpp:218] Iteration 0 (0 iter/s, 29.549s/100 iters), loss = 2.30247
I0219 11:38:02.069957 32988 solver.cpp:218] Iteration 100 (1.3866 iter/s, 72.119s/100 iters), loss = 1.75591
I0219 11:39:14.131213 32988 solver.cpp:218] Iteration 200 (1.38771 iter/s, 72.061s/100 iters), loss = 1.59993
I0219 11:40:26.189925 32988 solver.cpp:218] Iteration 300 (1.38777 iter/s, 72.058s/100 iters), loss = 1.31135
```

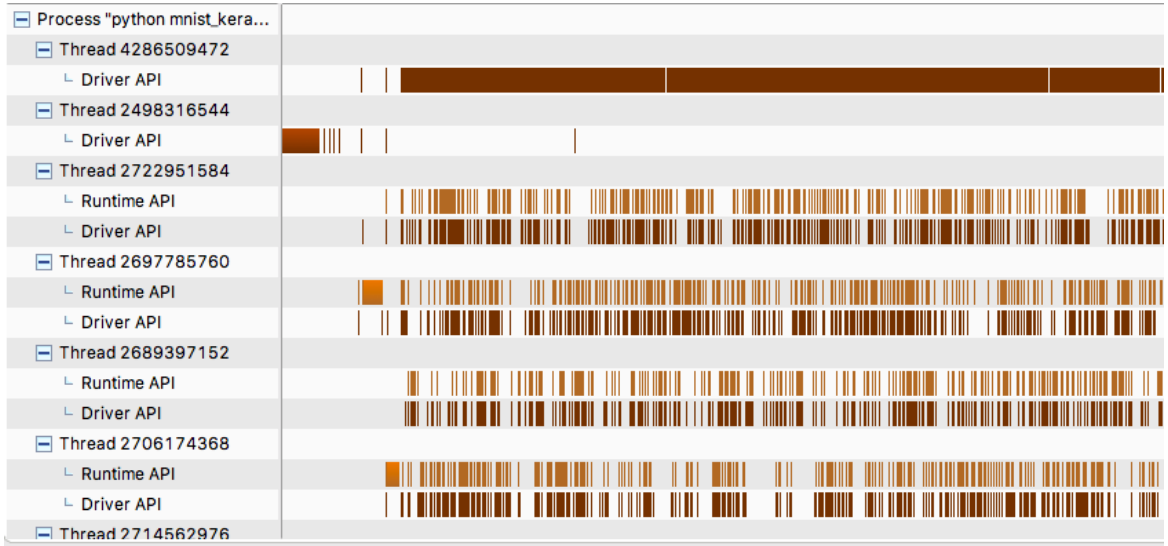
```
[root@aistation caffe-bvlc]# grep "Iteration" GPU.log | grep iters
I0219 11:52:51.256464 33291 solver.cpp:218] Iteration 0 (0 iter/s, 0.154748s/100 iters), loss = 2.30216
I0219 11:52:51.512588 33291 solver.cpp:218] Iteration 100 (390.5 iter/s, 0.256082s/100 iters), loss = 1.69603
I0219 11:52:51.753172 33291 solver.cpp:218] Iteration 200 (415.78 iter/s, 0.240512s/100 iters), loss = 1.53713
I0219 11:52:51.992805 33291 solver.cpp:218] Iteration 300 (417.425 iter/s, 0.239564s/100 iters), loss = 1.26168
I0219 11:52:52.232151 33291 solver.cpp:218] Iteration 400 (417.914 iter/s, 0.239284s/100 iters), loss = 1.23572
```

GPU Profiler

- Run the model -> **nvprof** -o test-%p.nvvp --profile-child-processes python mnist_keras.py
  - Output: test-129940.nvvp

```
Epoch 9/10
60000/60000 [==============================] - 3s 50us/step - loss
al_acc: 0.0980
Epoch 10/10
60000/60000 [==============================] - 3s 52us/step - loss
al_acc: 0.0980
Test loss: 1.19209303762e-07
Test accuracy: 0.098
35.9161610603
```

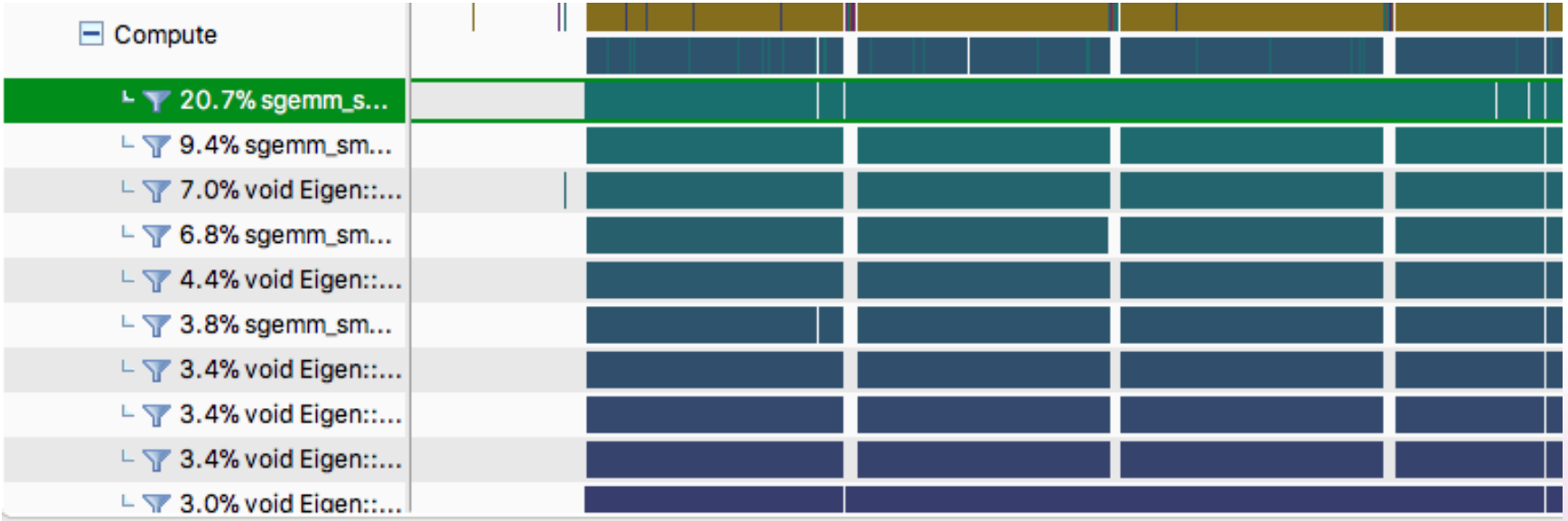
GPU Profiler



Analysis | GPU Details (Summary) | CPU Details | OpenACC Details | Console | Settings

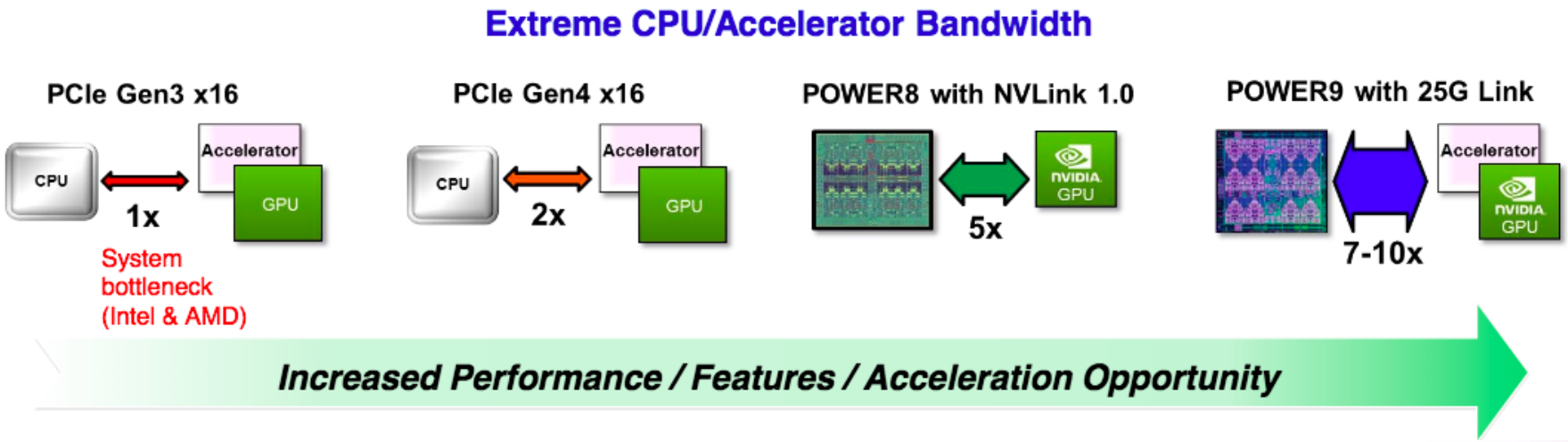
Name	Invocations	Avg. Duration	Regs	Static SMem	Avg. Dynamic SMem
memset (0)	0	0 ns	0	0	0
void Eigen::internal::EigenMetaKernel<Eigen::TensorEvaluator<Eigen::Tenso...	9380	2.951 µs	11	0	0
void Eigen::internal::EigenMetaKernel<Eigen::TensorEvaluator<Eigen::Tenso...	11586	3.124 µs	11	0	0
void Eigen::internal::EigenMetaKernel<Eigen::TensorEvaluator<Eigen::Tenso...	4690	3.132 µs	8	0	0
void Eigen::internal::EigenMetaKernel<Eigen::TensorEvaluator<Eigen::Tenso...	4690	3.205 µs	12	0	0
void Eigen::internal::EigenMetaKernel<Eigen::TensorEvaluator<Eigen::Tenso...	4690	3.218 µs	12	0	0
void Eigen::internal::EigenMetaKernel<Eigen::TensorEvaluator<Eigen::Tenso...	28140	3.253 µs	17	0	0
void Eigen::internal::EigenMetaKernel<Eigen::TensorEvaluator<Eigen::Tenso...	5793	3.287 µs	12	0	0
void Eigen::internal::EigenMetaKernel<Eigen::TensorEvaluator<Eigen::Tenso...	15173	3.403 µs	13	0	0
void Eigen::internal::EigenMetaKernel<Eigen::TensorEvaluator<Eigen::Tenso...	5793	3.686 µs	14	0	0
void Eigen::internal::EigenMetaKernel<Eigen::TensorEvaluator<Eigen::Tenso...	9380	3.821 µs	12	0	0
void Eigen::internal::FullReductionKernel<int=256, int=128, Eigen::TensorE...	17379	4.16 µs	14	0	0
void Eigen::internal::EigenMetaKernel<Eigen::TensorEvaluator<Eigen::Tenso...	18760	4.226 µs	16	0	0

GPU Profiler



Name	Invocations	Avg. Duration	Regs	Static SMem	Avg. Dyn
sgemm_sm35_ldg_nn_64x16x64x16x16	17379	117.944 µs	71	8468	

IBM POWERAI (Power9) – Ideal for Acceleration !!

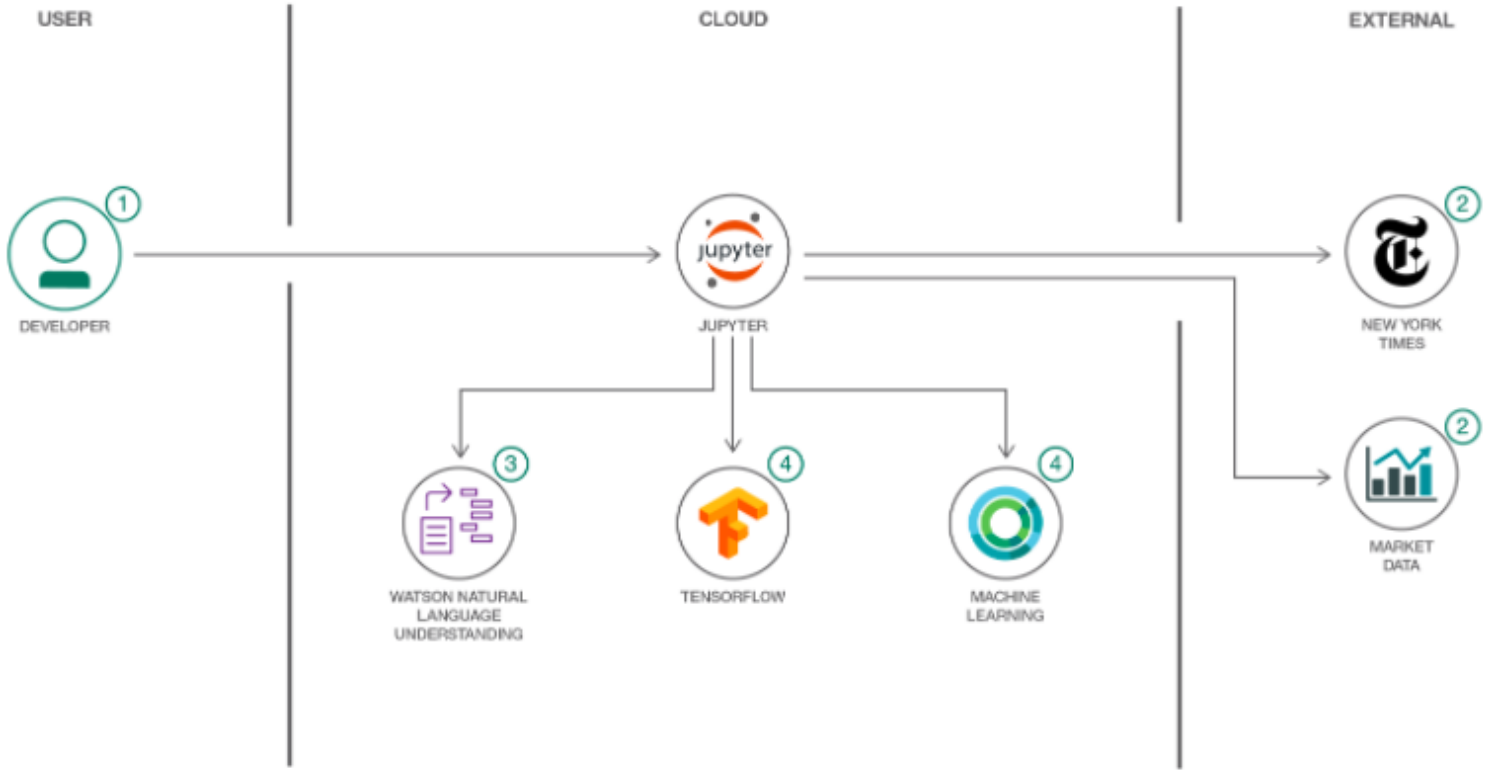


GPU / CPU platform environment:

- PowerAI through cloud - **https://www.ibm.com/us-en/marketplace/deep-learning-platform - Free Trial access**
- **IBM cloud (Softlayer) – Virtual server & Bare Metal option (Bluemix.net)**
- **IBM Cloud Private (ICP) on Power9 : Kube cluster based environment**
- **Complete CPU based environment – (cloud based) IBM Watson Studio (DSX) , with Python notebook , Apache Spark , Object storage components**
- **IBM Cloud – for complete AI based solution (Bluemix.net)**

Try it out – Code pattern – developer.ibm.com/code

https://developer.ibm.com/code/patterns/accelerate-training-of-machine-learning-algorithms/



- Meetup - meetup.com/IBMDevConnect-Bangalore/
- Youtube channel – Rajesh Jeyapaul
- Linked In – RajeshJeyapaul
- Blog – rajeshkj.blog

Thank you

Stay Connected and continue coding !

Code & instructions

https://github.com/IBMDevConnect
https://github.com/IBM
https://github.com/IBM-Cloud
https://ibm-cloud.github.io/#!/
http://ibm.github.io
https://github.com/watson-developer-cloud
https://github.com/ibm-bluemix-mobile-services

developerWorks

https://developer.ibm.com/in/
https://developer.ibm.com/tv/

Recipes

https://developer.ibm.com/recipes/

Join our Slack team and stay in touch with the experts

https://ibmdevconnect.slack.com

Send in your request

http://ibm.biz/slackrequest

Apply for IBM Global Entrepreneur Program

https://developer.ibm.com/startups

Join our Meetup groups

Bangalore :
https://www.meetup.com/IBMDevConnect-Bangalore

Delhi / Gurugram / Noida :
https://www.meetup.com/ibmcloudecosystem/

Mumbai / Pune :
https://www.meetup.com/Cloud-Mumbai-Meetup/

Hyderabad / Vishakapatnam:
https://www.meetup.com/Hyderabad-Cognitive-with-Cloud

Backup slides

Deep Learning Training pipeline

- Kind of Data and its availability
  - CSV , text , images , public dataset, etc.,
- Pre-process the data
  - Removing unwanted column/images , filling up missing values , rescale the images(Data Augmentation) , etc.,
- Choose appropriate activation function
  - Sigmoid, tanh , ReLU , etc.,
- Number of hidden units / layers , Weights Initialization , Learning rate ,
- Number of epochs / Training Iterations

Future Read – Transfer Learning

- From “Generalization of a specific task “ To “Generalization for other co-related tasks”

  - The pattern “Straight line” and “circle” can be used to identify other similar objects
  - The keyword in a chat dialog can be used for “next word prediction” , “speech classification”
- Re-uses the features of a trained model , time saved in labelling, training etc.,..

Artificial intelligence brings new tools to astronomy

- **Objective** - To recognize three key phases of galaxy evolution
  - Young, hot stars emit short "blue" wavelengths of light
  - Older, cooler stars emit more "red" light
- **Training Data** - computer simulations of galaxy formation
- **Validation Data** - images of galaxies from the Hubble Space Telescope
- **Observation** - "blue nugget" phase only occurs in galaxies with masses within a certain range. This is followed by quenching of star formation in the central region, leading to a compact "red nugget" phase
- “Deep learning algorithm is identifying on its own a pattern”

Edge detection

Image size= 320 x 280
RGB = 3

320×280×3
= 268,800 floating-point operations


GREAT INDIAN
DEVELOPER™
SUMMIT 2019
Conference : April 23-26, Bangalore
12th EDITION
#gids19
Register early and get the best discounts!
www.developersummit.com
@greatindiandev
bit.ly/gidslinkedin
facebook.com/gids19
bit.ly/saltmarchyoutube
flickr.com/photos/saltmarch/