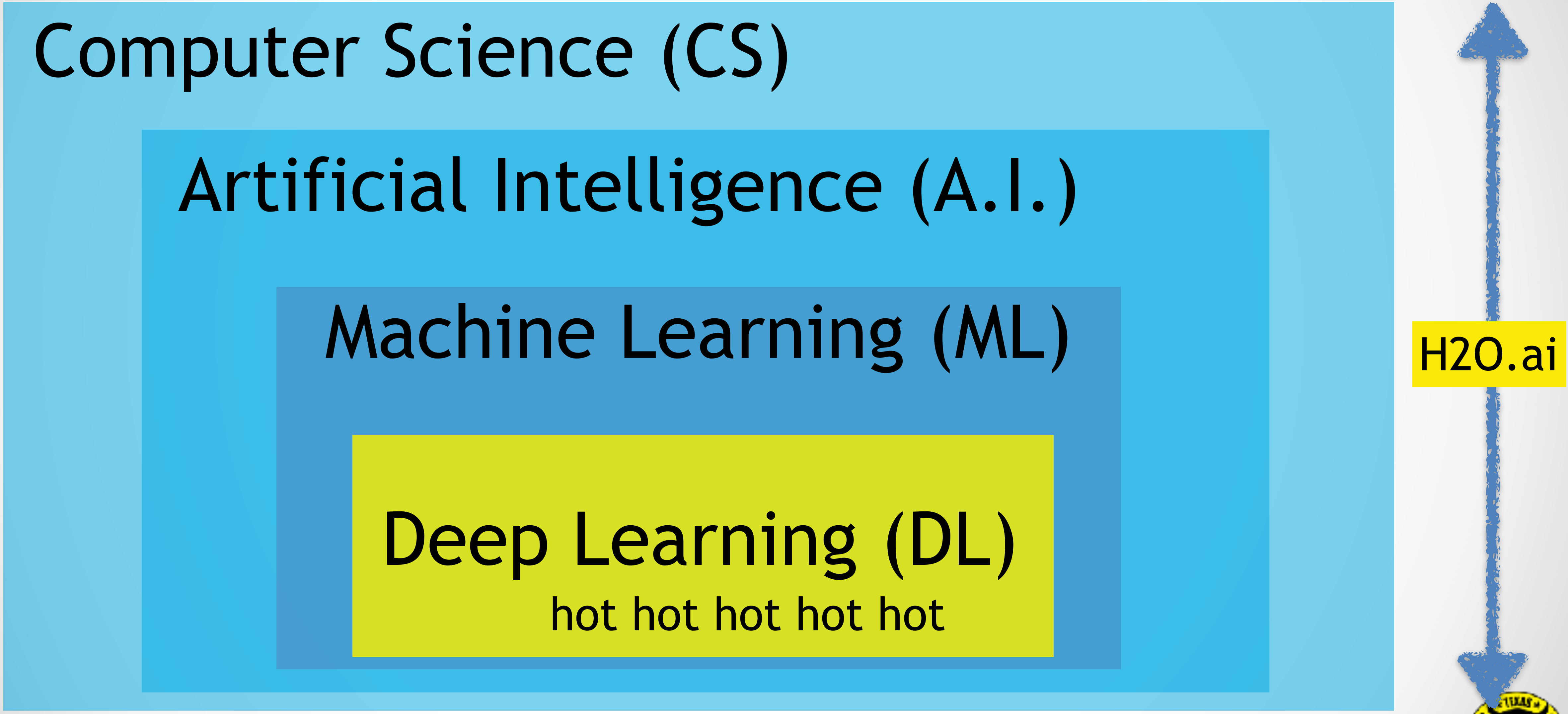
Deep Water

GPU Deep Learning for H2O

Arno Candel, PhD
Chief Architect, Physicist & Hacker, H2O.ai
@ArnoCandel

The Open Tour, Dallas, TX
Oct 26 2016

Overview


Computer Science (CS)
Artificial Intelligence (A.I.)
Machine Learning (ML)
Deep Learning (DL)
hot hot hot hot hot
H2O.ai

A Simple Deep Learning Model: Artificial Neural Network

from 1970s, now rebranded as DL

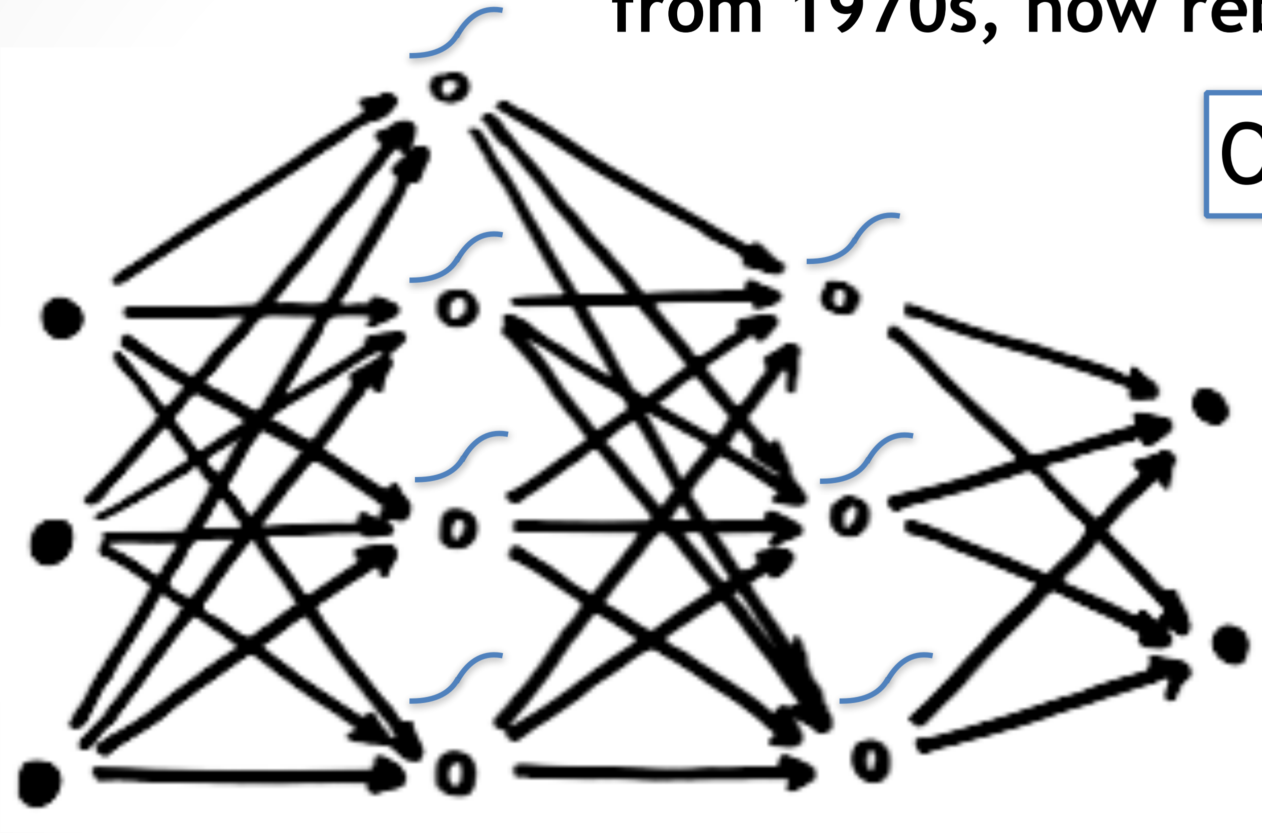
IN: data

OUT: prediction

heartbeat

oxygen

blood pressure



send to intensive care unit (ICU)

send to regular care

nodes : neuron activations (real numbers) — represent features

arrows : connecting weights (real numbers) — learned during training

: non-linearity x -> f(x) — adds model complexity

H2O.ai

A step back: A.I. was coined over 60 years ago

John McCarthy
Princeton, Bell Labs, Dartmouth, later: MIT, Stanford

1955: "A proposal for the Dartmouth summer research project on Artificial Intelligence"

with Marvin Minsky (MIT), Claude Shannon (Bell Labs) and Nathaniel Rochester (IBM)

http://www.asiapacific-mathnews.com/04/0403/0015_0020.pdf

1955 proposal for the Dartmouth summer research project on A.I.

“**We propose** that **a 2-month, 10-man study of artificial intelligence** be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. The study is to proceed on the basis **of the conjecture that every aspect of learning** and any other feature of intelligence **can in principle be so precisely described that a machine can be made to simulate it**. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves. **We think that a significant advance can be made in** one or more of these problems if a carefully selected group of scientists work on it together for **one summer.**”

Step 1: Great Algorithms + Fast Computers

"No computer will ever beat me at playing chess."

1997: Playing Chess
(IBM Deep Blue beats Kasparov)

Computer Science

30 custom CPUs, 60 billion moves in 3 mins

http://nautil.us/issue/18/genius/why-the-chess-computer-deep-blue-played-like-a-human

H2O.ai

"It takes a human to drive a car!"

2005: Self-driving Cars
DARPA Grand Challenge, 132 miles
(won by Stanford A.I. lab*)

Sensors & Computer Science

video, radar, laser, GPS, 7 Pentium computers

http://cs.stanford.edu/group/roadrunner/old/presskit.html

*A.I. lab was established by McCarthy et al. in the early '60s

"Computers can't answer arbitrary questions!"

2011: Jeopardy (IBM Watson)

In-Memory Analytics/ML

4 TB of data (incl. wikipedia), 90 servers, 16 TB RAM, Hadoop, 6 million logic rules

https://www.youtube.com/watch?v=P18EdAKuC1U

https://en.wikipedia.org/wiki/Watson_(computer)

Note: IBM Watson received the question in electronic written form, and was often able to press the answer button faster than the competing humans.

“Computers don’t understand our language!”

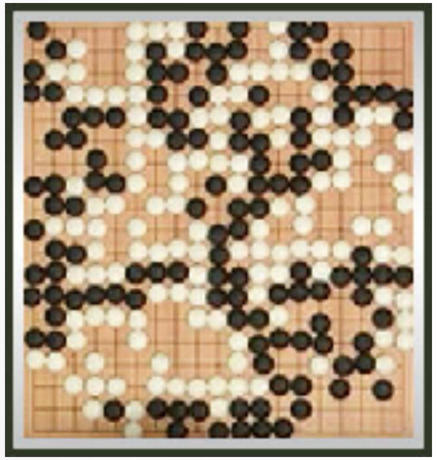
2014: Google
(acquired Quest Visual)

Deep Learning

Convolutional and Recurrent Neural Networks, with training data from users

- Translate between 103 languages by typing
- Instant camera translation: Use your camera to translate text instantly in 29 languages
- Camera Mode: Take pictures of text for higher-quality translations in 37 languages
- Conversation Mode: Two-way instant speech translation in 32 languages
- Handwriting: Draw characters instead of using the keyboard in 93 languages

“Go is too complex for computers to master!”



2014: Atari Games (DeepMind)

trained from raw pixel values, no human rules

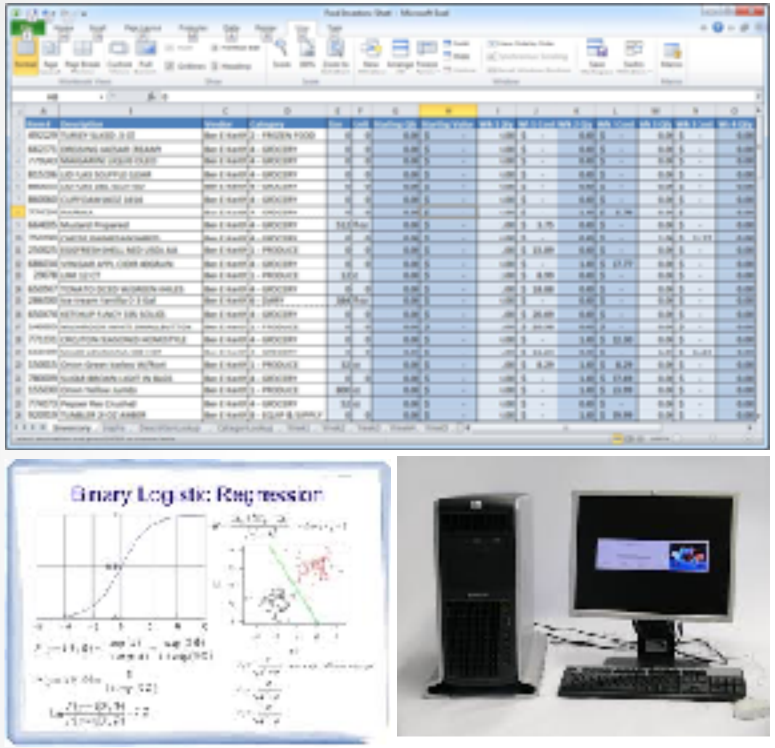
Deep Learning

+ reinforcement learning, tree search,
Monte Carlo, GPUs, playing against itself, …

2016: AlphaGo (Google DeepMind)

Go board has approx.
200,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,
000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000 (2E170) possible positions.

Things are Changing Quickly



Yesterday: Small Data (<GB)

Data + Skills

are good for business

Today: Big Data (TeraBytes, ExaBytes)

Data + Machine Learning

ARE the business

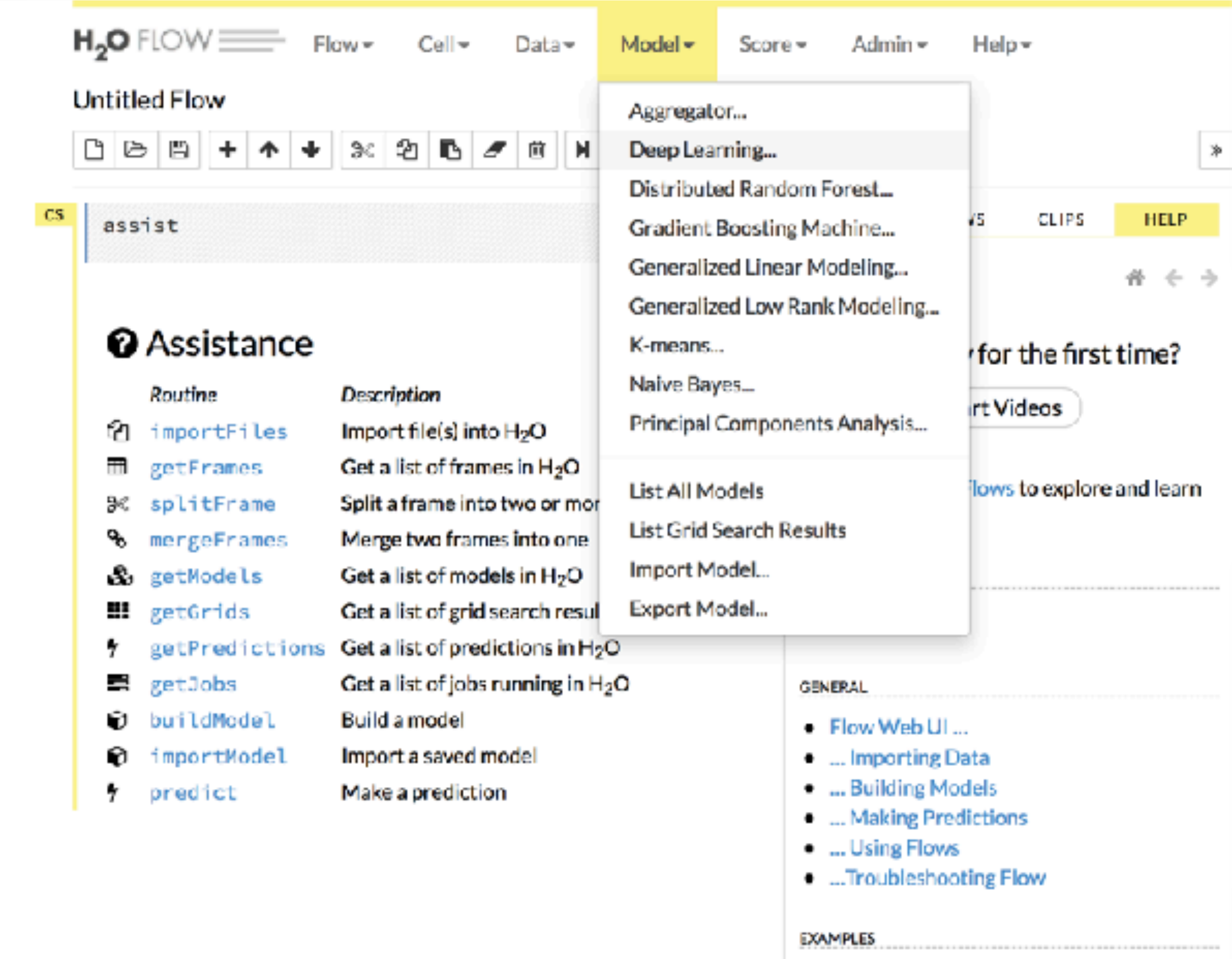
Deep Learning is in the Center of This Revolution

- conceptually simple
- solves many problems
- benefits from big data
- super-human results

However:

- hard to interpret — but solutions exist
- lots of architectural choices — require lucky PhDs, open-source helps
- lots of hyper-parameters — AutoML can do the tuning for you
- slow to train on big data — dedicated hardware helps (GPU clusters)
- rapidly changing landscape — Deep Water unifies open-source APIs

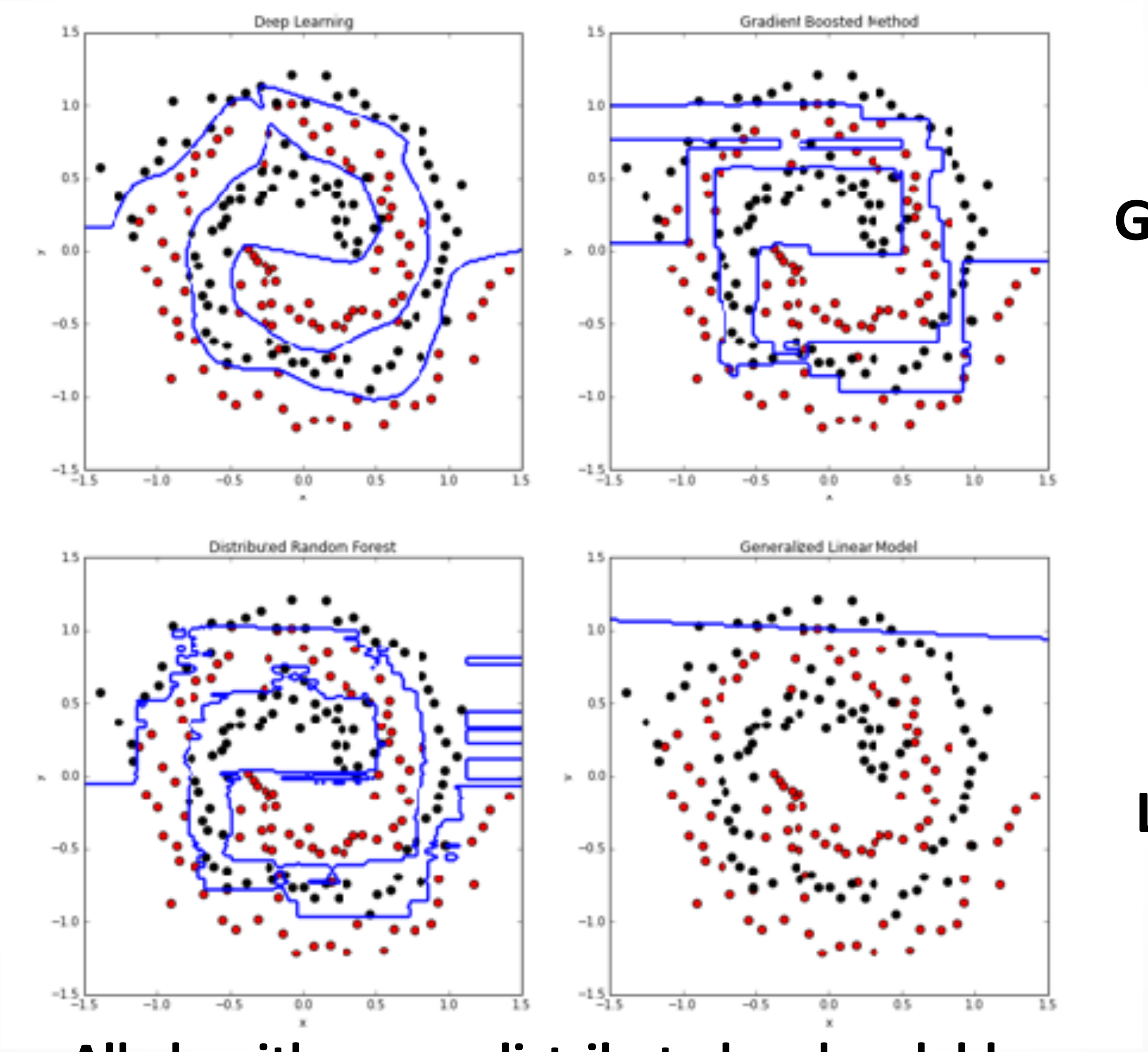
H2O pioneered Easy-To-Use Open Source Deep Learning



H2O pioneered Easy-To-Use Open Source Deep Learning

Deep Learning

Gradient Boosting Machine



Distributed Random Forest

Generalized Linear Modeling

All algorithms are distributed and scalable

H2O Deep Learning Is Widely Used

The usage of Hadoop/Big Data tools grew to 39%, up from 29% in 2015 (and 17% in 2014), driven by Apache Spark, MLlib (Spark Machine Learning Library) and H2O.

See also

- KDnuggets interview with Spark Creator Matei Zaharia
- KDnuggets interview with Arno Candel, H2O.ai on How to Quick Start Deep Learning with H2O

http://www.kdnuggets.com

H2O and TensorFlow are tied

usage of Deep Learning tools in past year

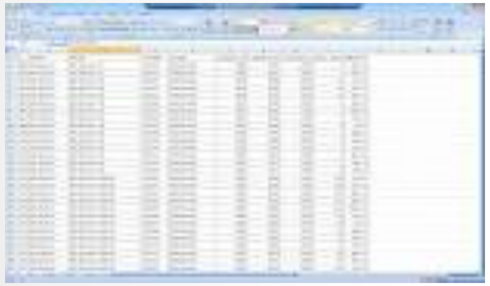
Deep Learning Tools & Platforms: Caffe (63)	0%
Cuda-convnet (22)	0%
Convnet.js (8)	0%
darch (2)	0%
Deeplearning4j (46)	0%
H2O (190)	1%
MATLAB Deep Learning Toolbox (57)	0%
Microsoft CNTK (25)	0%
mxnet (16)	0%
Nervana (2)	0%
Tensorflow (190)	1%
Theano ecosystem including Keras, Lasagne, Pylearn2 (140)	1%
Torch (28)	0%
Veles (2)	0%
Other Deep Learning Tools (104)	1%

Deep Water opens the Floodgates for state-of-the-art Deep Learning

H2O Deep Learning:

simple multi-layer networks, CPUs

1-5 layers
MBs/GBs of data

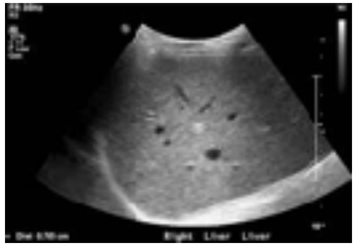


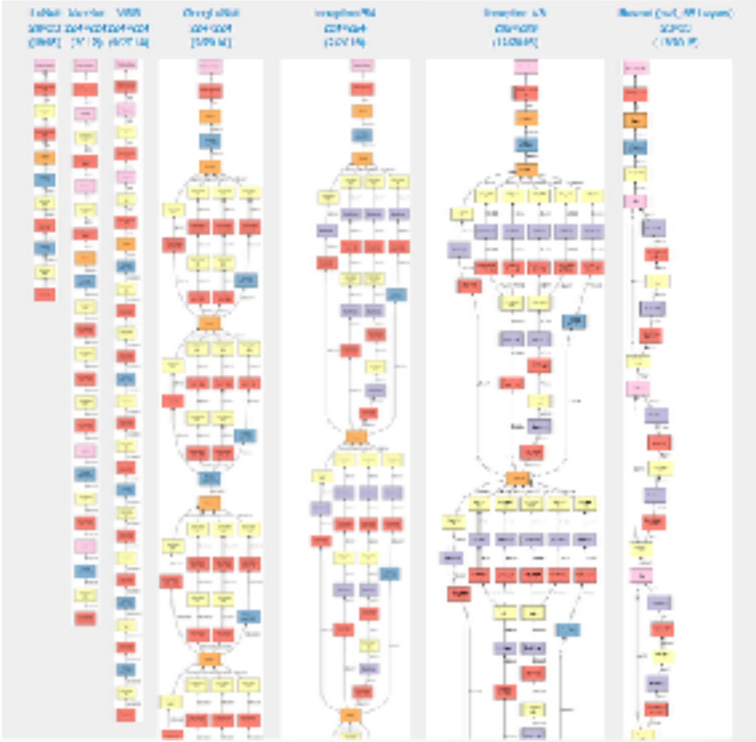
Limited to business analytics,
statistical models (CSV data)

H2O Deep Water:

arbitrary networks, CPUs or GPUs

1-1000 layers
GBs/TBs of data





Large networks for big data
(e.g. image 1000x1000x3 -> 3m inputs per observation)

Deep Water: Best Open-Source Deep Learning

Enterprise Deep Learning for Business Transformation

Deep Water: **THE open-source Deep Learning Platform**	H2O integrates the top open-source DL tools
Native GPU support	is up to 100x faster than
Enterprise Ready	Easy to **train, compare and deploy**, **interactive** and **scalable**, from **Flow, R, Python, Spark/Scala**, Java, REST
New Big Data Use Cases (previously impossible or difficult in H2O)	**Image** - social media, manufacturing, healthcare, … **Video** - UX/UI, security, automotive, social media, … **Sound** - automotive, security, call centers, healthcare, … **Text** - NLP, sentiment, security, finance, fraud, … **Time Series** - security, IoT, finance, e-commerce, …

Deep Water Architecture

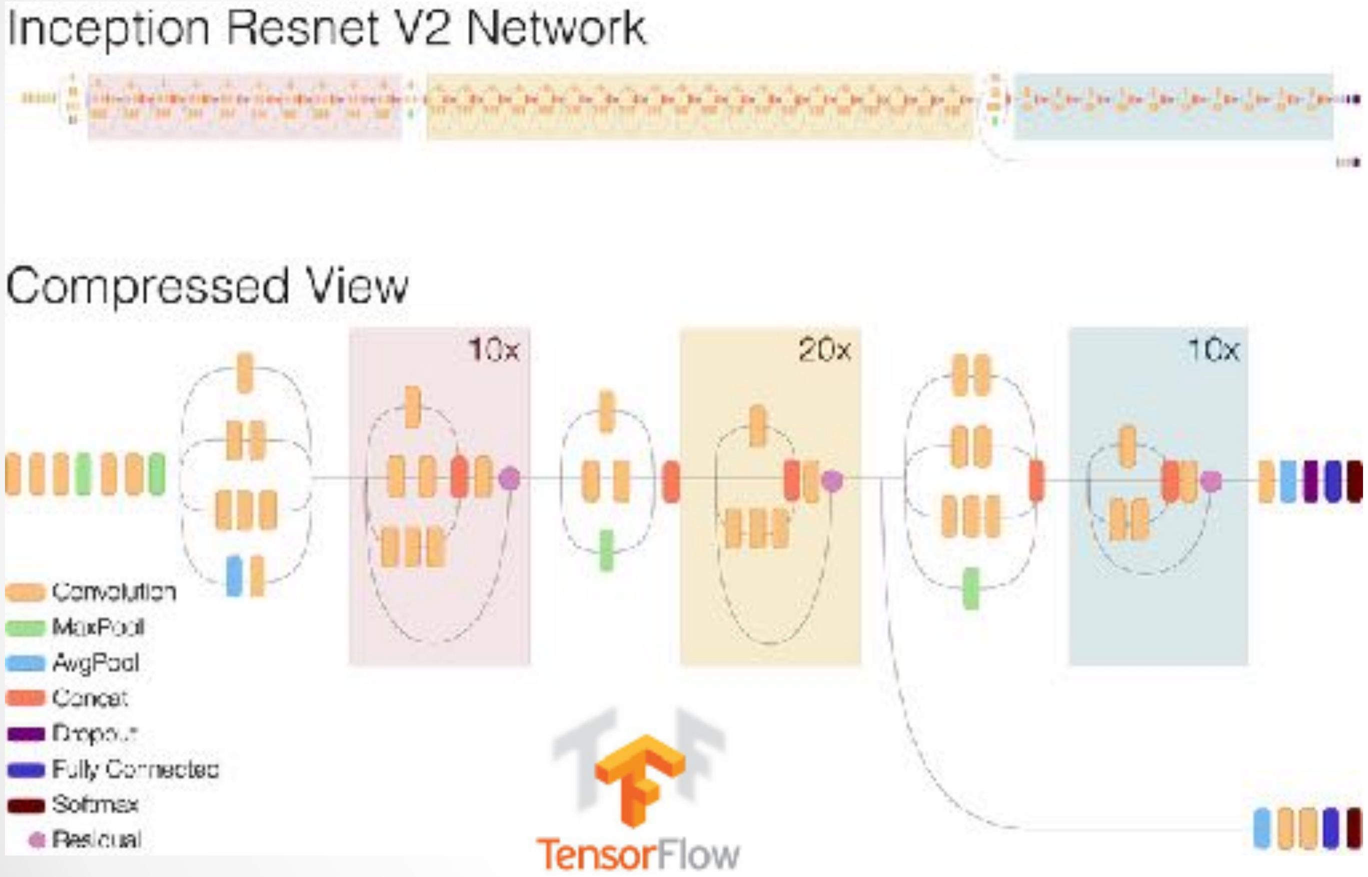
Data Science API: R, Python, Flow GUI

REST API
Web server

Scala Spark	H2O Java Execution Engine	H2O RPC / other comm.	H2O Java Execution Engine	Scala Spark
	TensorFlow/mxnet/Caffe C++		TensorFlow/mxnet/Caffe C++	
	GPU / CPU		GPU / CPU	
	Node 1		Node N	

Step 1: Leverage Research Community Code, Data and Models

World's best Image Classifier (Google + Microsoft, Aug 2016)



https://research.googleblog.com/2016/08/improving-inception-and-image.html

mxnet

open-source implementation

```
stem_2_1x1 = Conv(data=concat1, num_filter=num_4_1,
stem_2_7x1 = Conv(data=stem_2_1x1, num_filter=num_4_
            suffix='_conv_2')
stem_2_1x7 = Conv(data=stem_2_7x1, num_filter=num_4_
            suffix='_conv_3')
stem_2_3x3 = Conv(data=stem_2_1x7, num_filter=num_4_

concat2 = mx.sym.Concat(*[stem_1_3x3, stem_2_3x3], :

pool2 = mx.sym.Pooling(data=concat2, kernel=(3, 3),
            name=('%s_%s_pool2' % ('max',
stem_3_3x3 = Conv(data=concat2, num_filter=num_5_1,
            suffix='_conv_1', withRelu=False)

concat3 = mx.sym.Concat(*[pool2, stem_3_3x3], name=(
bn1 = mx.sym.BatchNorm(data=concat3, name=('%s_bn1'
act1 = mx.sym.Activation(data=bn1, act_type='relu',

return act1

def InceptionResnetV2A(data,
                num_1_1,
                num_2_1, num_2_2,
                num_3_1, num_3_2, num_3_3,
                proj,
                name,
                scaleResidual=True):
    import mxnet as mx
    init = data

    a1 = Conv(data=data, num_filter=num_1_1, name=('%s_

    a2 = Conv(data=data, num_filter=num_2_1, name=('%s_
    a2 = Conv(data=a2, num_filter=num_2_2, kernel=(3, 3)

    a3 = Conv(data=data, num_filter=num_3_1, name=('%s_
    a3 = Conv(data=a3, num_filter=num_3_2, kernel=(3, 3)
    a3 = Conv(data=a3, num_filter=num_3_3, kernel=(3, 3)
```

H2O.ai

Step 2: Train Models with Familiar APIs

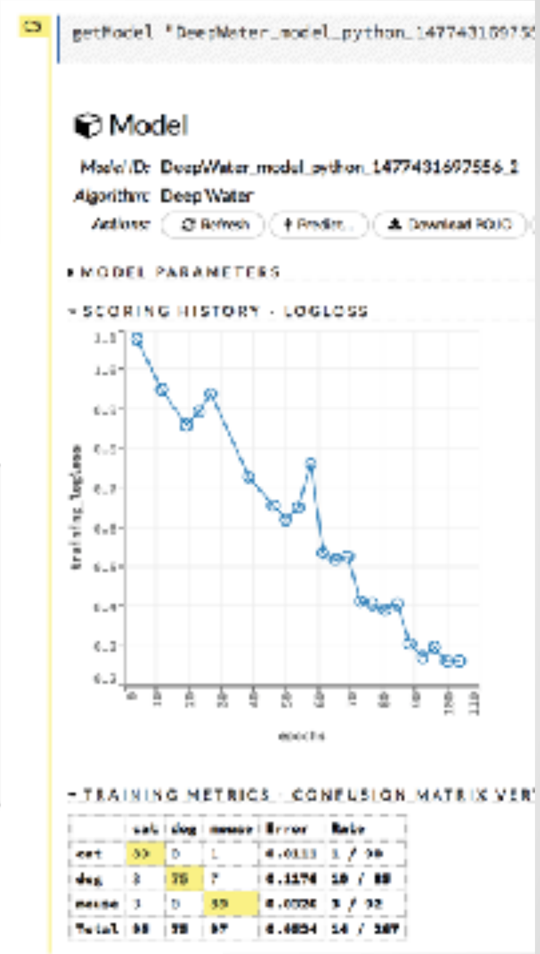
```
frame = h2o.import_file(PATH+"/bigdata/laptop/deepwater/imagenet/cat_dog_mouse.csv")
print(frame.head(5))
```

```
Parse progress: |████████████████████████████████████████████████████████| 100%
```

C1	C2
bigdata/laptop/deepwater/imagenet/cat/102194502_49f003abd9.jpg	cat
bigdata/laptop/deepwater/imagenet/cat/11146807_00a5f35255.jpg	cat
bigdata/laptop/deepwater/imagenet/cat/1140846215_70e326f868.jpg	cat

```
nclasses = frame[1].nlevels()[0]
get_symbol(nclasses).save("/tmp/symbol_inception_resnet_v2-py.json")
model = H2ODeepWaterEstimator(epochs=20, learning_rate=1e-3, learning_rate_annealing=1e-5,
                              mini_batch_size=16,
                              network_definition_file="/tmp/symbol_inception_resnet_v2-py.json",
                              image_shape=[299,299],
                              channels=3)
model.train(x=[0],y=1, training_frame=frame)
model.show()
```

```
deepwater Model Build progress: |████████████████████████████████████████| 100%
Model Details
=============
H2ODeepWaterEstimator :  Deep Water
Model Key:  DeepWater_model_python_1477179782032_5
Status of Deep Learning Model: user, 116.1 MB, predicting C2, 3-class classification, 5,632 training samples, mini-ba
tch size 16
```

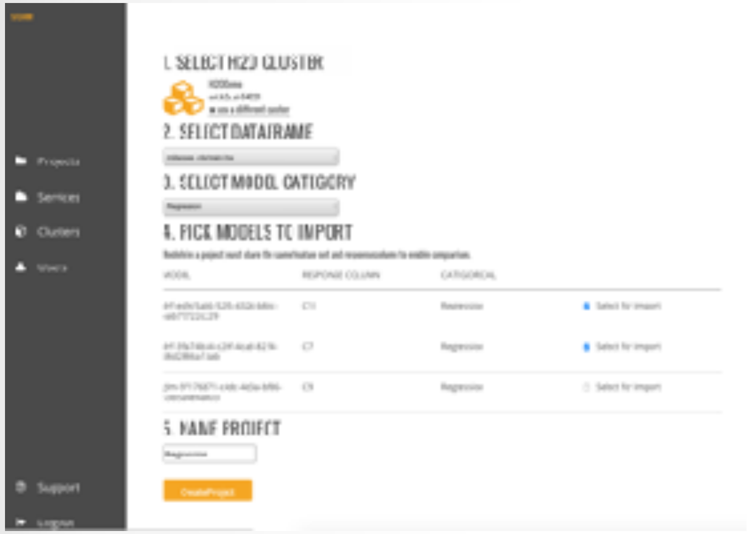


H2O.ai

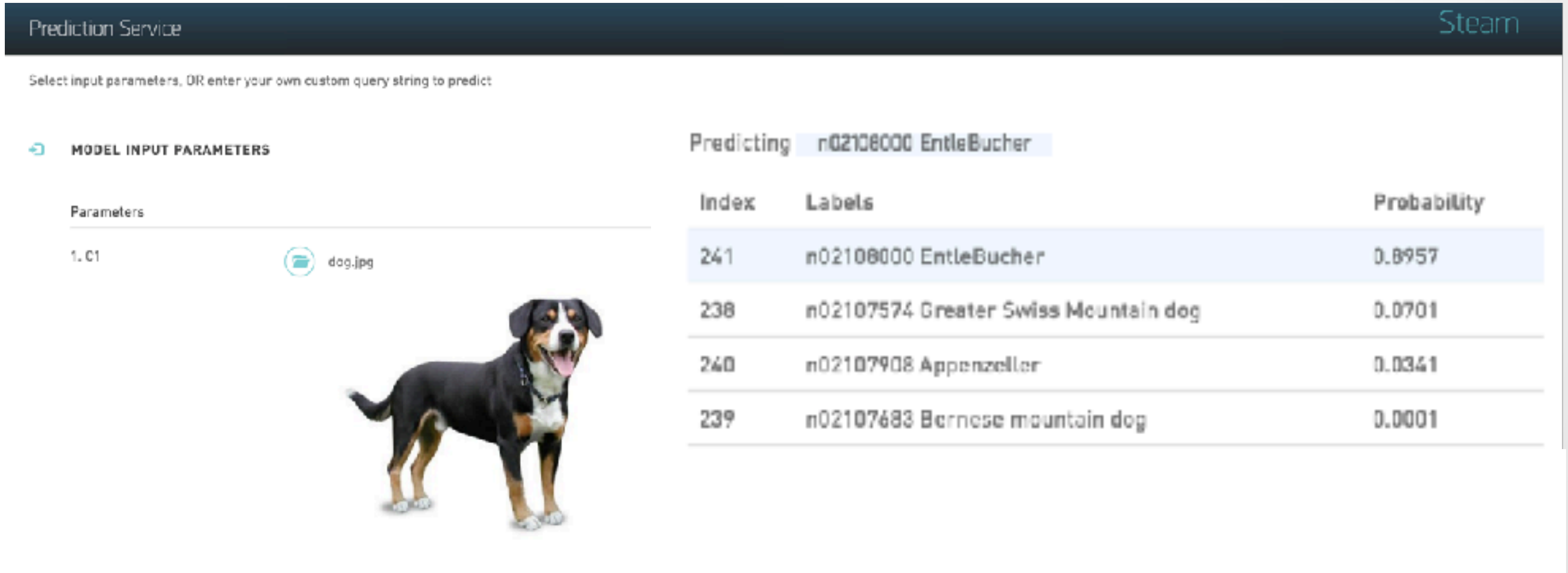
https://github.com/h2oai/h2o-3/blob/master/h2o-py/tests/testdir_algos/deepwater/pyunit_inception_resnet_v2_deepwater.py

Build your own models in the Deep Water Hands-On Session Today!

Pretrained Image Classifier - Standalone Scoring Server Deployed in Seconds



http://deepwater.h2o.ai/classy-demo/



Steam
Compare Models

Scoring Server
Build smarter applications and data products

H2O.ai

High-Performance Multi-GPU Training with Caffe

Training on a 16-GPU EC2 instance with Caffe

https://aws.amazon.com/blogs/aws/new-p2-instance-type-for-amazon-ec2-up-to-16-gpus/

Instance Name	GPU Count	vCPU Count	Memory	Parallel Processing Cores	GPU Memory	Network Performance
p2.xlarge	1	4	61 GiB	2,496	12 GiB	High
p2.8xlarge	8	32	488 GiB	19,968	96 GiB	10 Gigabit
p2.16xlarge	16	64	732 GiB	39,936	192 GiB	20 Gigabit

Caffe launched from Java:

java -cp target/dependency/*:target/classes deepwater.examples.ImageNet

H2O integration is in progress

```
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 367.48                 Driver Version: 367.48                    |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla K80           On   | 0000:00:0F.0     Off |                    0 |
| N/A   67C    P0   134W / 149W |   3246MiB / 11439MiB |     55%      Default |
+-------------------------------+----------------------+----------------------+
|   1  Tesla K80           On   | 0000:00:10.0     Off |                    0 |
| N/A   57C    P0   131W / 149W |   2750MiB / 11439MiB |     71%      Default |
+-------------------------------+----------------------+----------------------+
|   2  Tesla K80           On   | 0000:00:11.0     Off |                    0 |
| N/A   65C    P0   144W / 149W |   2801MiB / 11439MiB |     58%      Default |
+-------------------------------+----------------------+----------------------+
|   3  Tesla K80           On   | 0000:00:12.0     Off |                    0 |
| N/A   63C    P0   147W / 149W |   2750MiB / 11439MiB |     92%      Default |
+-------------------------------+----------------------+----------------------+
|   4  Tesla K80           On   | 0000:00:13.0     Off |                    0 |
| N/A   67C    P0   147W / 149W |   3852MiB / 11439MiB |     89%      Default |
+-------------------------------+----------------------+----------------------+
|   5  Tesla K80           On   | 0000:00:14.0     Off |                    0 |
| N/A   58C    P0   143W / 149W |   2750MiB / 11439MiB |     93%      Default |
+-------------------------------+----------------------+----------------------+
|   6  Tesla K80           On   | 0000:00:15.0     Off |                    0 |
| N/A   65C    P0   150W / 149W |   2801MiB / 11439MiB |     85%      Default |
+-------------------------------+----------------------+----------------------+
|   7  Tesla K80           On   | 0000:00:16.0     Off |                    0 |
| N/A   59C    P0   154W / 149W |   2750MiB / 11439MiB |     92%      Default |
+-------------------------------+----------------------+----------------------+
|   8  Tesla K80           On   | 0000:00:17.0     Off |                    0 |
| N/A   67C    P0   147W / 149W |   2903MiB / 11439MiB |     92%      Default |
+-------------------------------+----------------------+----------------------+
|   9  Tesla K80           On   | 0000:00:18.0     Off |                    0 |
| N/A   57C    P0   157W / 149W |   2750MiB / 11439MiB |     93%      Default |
+-------------------------------+----------------------+----------------------+
|  10  Tesla K80           On   | 0000:00:19.0     Off |                    0 |
| N/A   60C    P0   151W / 149W |   2801MiB / 11439MiB |     97%      Default |
+-------------------------------+----------------------+----------------------+
|  11  Tesla K80           On   | 0000:00:1A.0     Off |                    0 |
| N/A   60C    P0   153W / 149W |   2750MiB / 11439MiB |     97%      Default |
+-------------------------------+----------------------+----------------------+
|  12  Tesla K80           On   | 0000:00:1B.0     Off |                    0 |
| N/A   68C    P0   119W / 149W |   3852MiB / 11439MiB |     75%      Default |
+-------------------------------+----------------------+----------------------+
|  13  Tesla K80           On   | 0000:00:1C.0     Off |                    0 |
| N/A   60C    P0   147W / 149W |   2750MiB / 11439MiB |     97%      Default |
+-------------------------------+----------------------+----------------------+
|  14  Tesla K80           On   | 0000:00:1D.0     Off |                    0 |
| N/A   65C    P0   124W / 149W |   2801MiB / 11439MiB |     85%      Default |
+-------------------------------+----------------------+----------------------+
|  15  Tesla K80           On   | 0000:00:1E.0     Off |                    0 |
| N/A   63C    P0   128W / 149W |   2750MiB / 11439MiB |     97%      Default |
+-------------------------------+----------------------+----------------------+
```

Caffe

All 16 GPUs are busy training

H2O.ai

Roadmap for Deep Water (Q4 2016):

Finish TensorFlow integration (C++/Python/Java):
Package Python on the backend to create trainable graphs

Finish Caffe integration (pure C++/Java):
Optimized Multi-GPU training (NVIDIA NCCL)

Add multi-GPU support for mxnet

Add more capabilities to H2O Deep Water:
Text/NLP, Time Series, LSTM, AutoEncoder,
Feature Extraction, Input/Output shape mapping, etc.

Breakout Tracks on Deep Water & GPU Backends

11:05-11:30 Dmitry Larko — Credit Card Default Prediction

2:40-3:00 Fabrizio Milo — TensorFlow internals, Wide & Deep models

3:00-4:00 Arno Candel — Hands-On Workshop with
Image Classification, Credit Card Default Prediction,
Benchmarking and Hyper-Parameter Search in Flow, Python and R

Live Demo