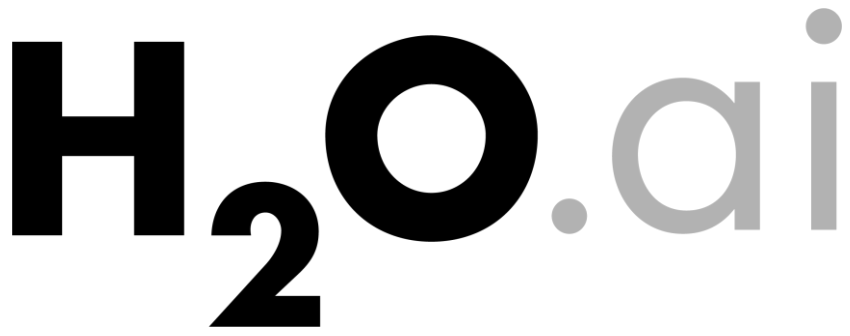
H_2O Deep Water

Making GPU Deep Learning Accessible to Everyone



Jo-fai (Joe) Chow

Data Scientist

joe@h2o.ai

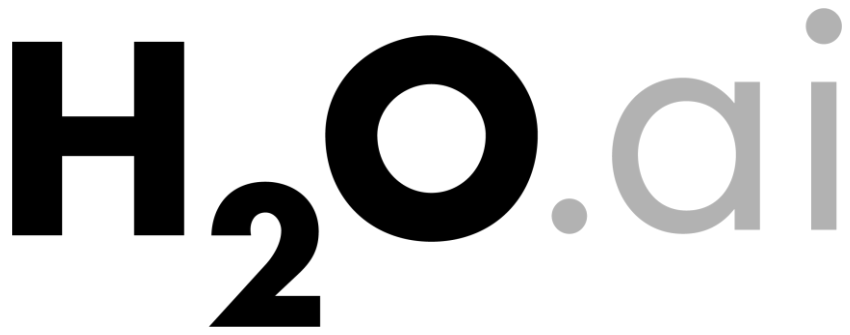
@matlabulous

Budapest Data Forum

13th June, 2017

H_2O Deep Water

Making GPU Deep Learning Accessible to Everyone



Jo-fai (Joe) Chow

Data Scientist

joe@h2o.ai

@matlabulous

All slides, data and code examples

http://bit.ly/h2o_meetups

Agenda

- Introduction
  - Company
  - Why H_2O?
  - H_2O Machine Learning Platform
- Deep Water
  - Motivation / Benefits
  - GPU Deep Learning Demos
- Other H_2O Developments

About Me

- Civil (Water) Engineer
  - 2010 – 2015
    - Consultant (UK)
      - Utilities
      - Asset Management
      - Constrained Optimization
    - EngD (Industrial PhD) (UK)
      - Infrastructure Design Optimization
      - Machine Learning + Water Engineering
      - Discovered H_2O in 2014
- Data Scientist
  - 2015 – 2016
    - Virgin Media (UK)
    - Domino Data Lab (Silicon Valley)
  - 2016 – Present
    - H_2O.ai (Silicon Valley)
  - How?
    - bit.ly/joe_kaggle_story

About H_2O.ai

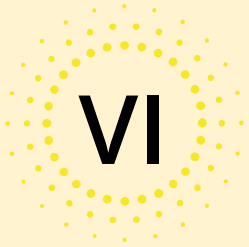
Company Overview

Founded	2011 Venture-backed, debuted in 2012
Products	• H_2O Open Source In-Memory AI Prediction Engine • Sparkling Water • Deep Water • Steam
Mission	Operationalize Data Science, and provide a platform for users to build beautiful data products
Team	70 employees • Distributed Systems Engineers doing Machine Learning • World-class visualization designers
Headquarters	Mountain View, CA




H_2O.ai Offers AI Open Source Platform

Product Suite to Operationalize Data Science with Visual Intelligence



Visual Intelligence and UX Framework For Data Interpretation and Story Telling on top of Beautiful Data Products

100% Open Source

H_2O.ai

In-Memory, Distributed Machine Learning Algorithms with Speed and Accuracy

Deep Water

State-of-the-art Deep Learning on GPUs with TensorFlow, MXNet or Caffe with the ease of use of H2O

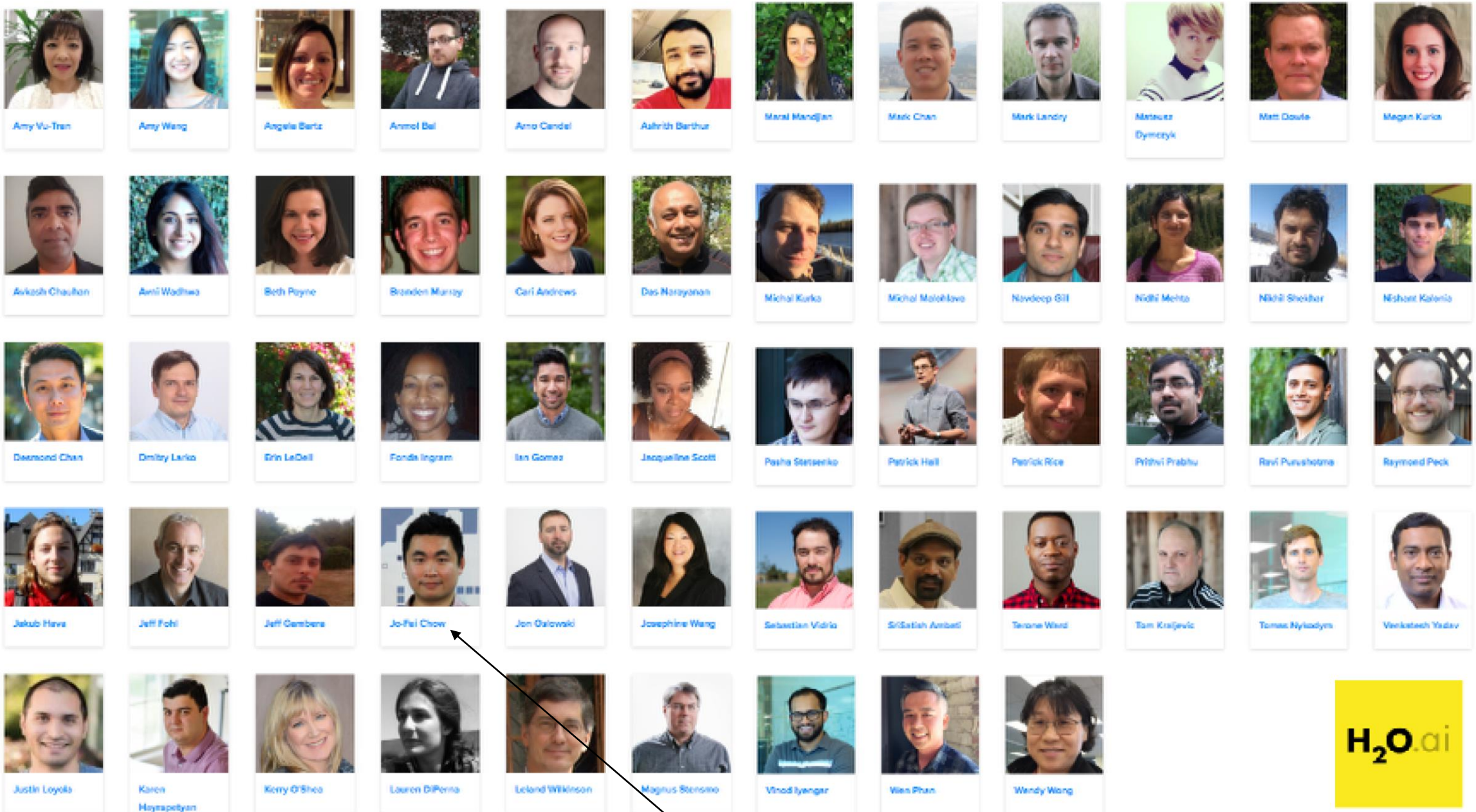
Spark + H_2O

SPARKLING WATER

H2O Integration with Spark. Best Machine Learning on Spark.

Steam

Operationalize and Streamline Model Building, Training and Deployment Automatically and Elastically


Amy Vu-Tran
Amy Wang
Angela Bartz
Anmol Bal
Arno Candel
Ashrith Barthur
Matt Dowle
Megan Kurka
Avkash Chauhan
Beth Payne
Brandon Murray
Cari Andrews
Das Narayanan
Michal Kurka
Michal Malohlava
Navdeep Gill
Nidhi Mehta
Nishant Kalonia
Desmond Chan
Dmitry Larko
Erin LeDell
Fonda Ingram
Ian Gomez
Jacqueline Scott
Pasha Stetsenko
Patrick Hall
Patrick Rice
Prithvi Prabhu
Ravi Purushotma
Raymond Peck
Jakub Hava
Jeff Fohl
Jeff Gambera
Jo-fai Chow
Jon Dalowski
Josephine Wang
Sebastian Vidrio
Terone Ward
Tom Kraljevic
Venkatesh Yadav
Justin Loyola
Karen Hovsepyan
Kerry O'Shea
Lauren DiPerna
Leland Wilkinson
Magnus Stensmo
Vinod Iyengar
Wen Phan
Wendy Wang

Joe (UK)

H2O.ai

Scientific Advisory Council

Dr. Trevor Hastie

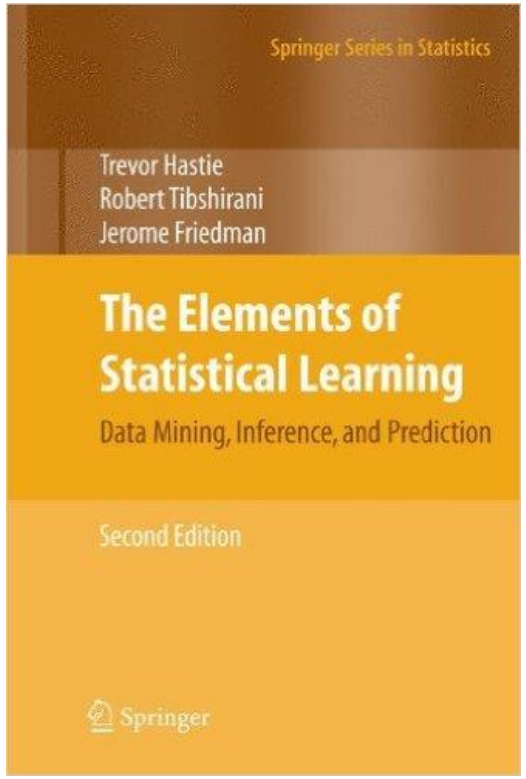
- John A. Overdeck Professor of Mathematics, Stanford University
- PhD in Statistics, Stanford University
- Co-author, *The Elements of Statistical Learning: Prediction, Inference and Data Mining*
- Co-author with John Chambers, *Statistical Models in S*
- Co-author, *Generalized Additive Models*

Dr. Robert Tibshirani

- Professor of Statistics and Health Research and Policy, Stanford University
- PhD in Statistics, Stanford University
- Co-author, *The Elements of Statistical Learning: Prediction, Inference and Data Mining*
- Author, *Regression Shrinkage and Selection via the Lasso*
- Co-author, *An Introduction to the Bootstrap*

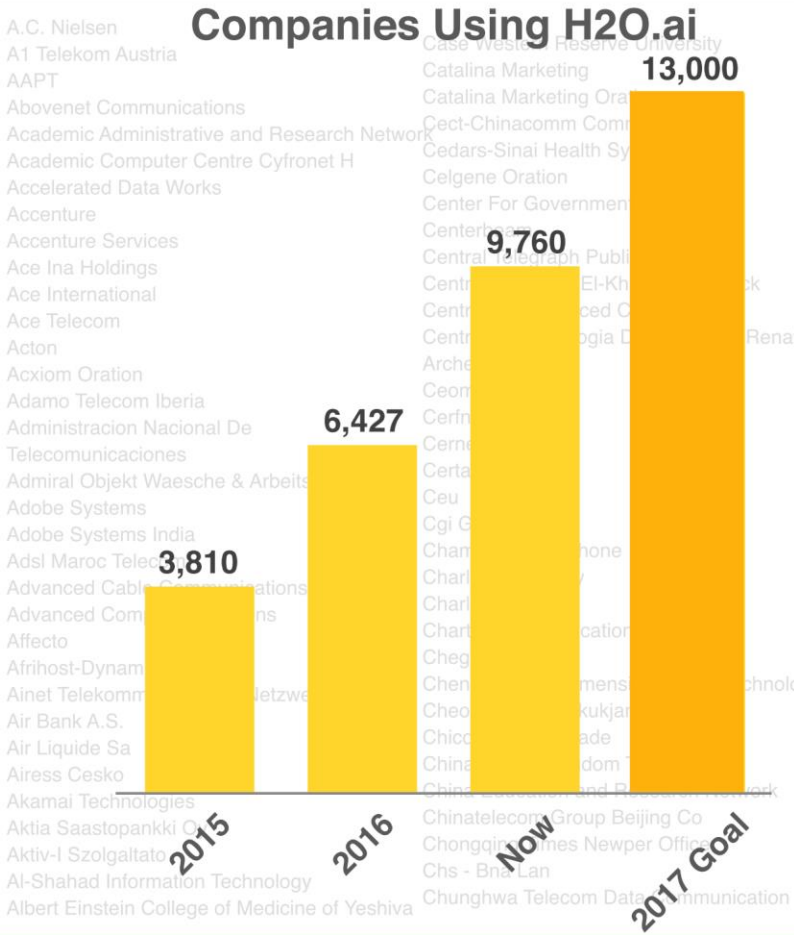
Dr. Steven Boyd

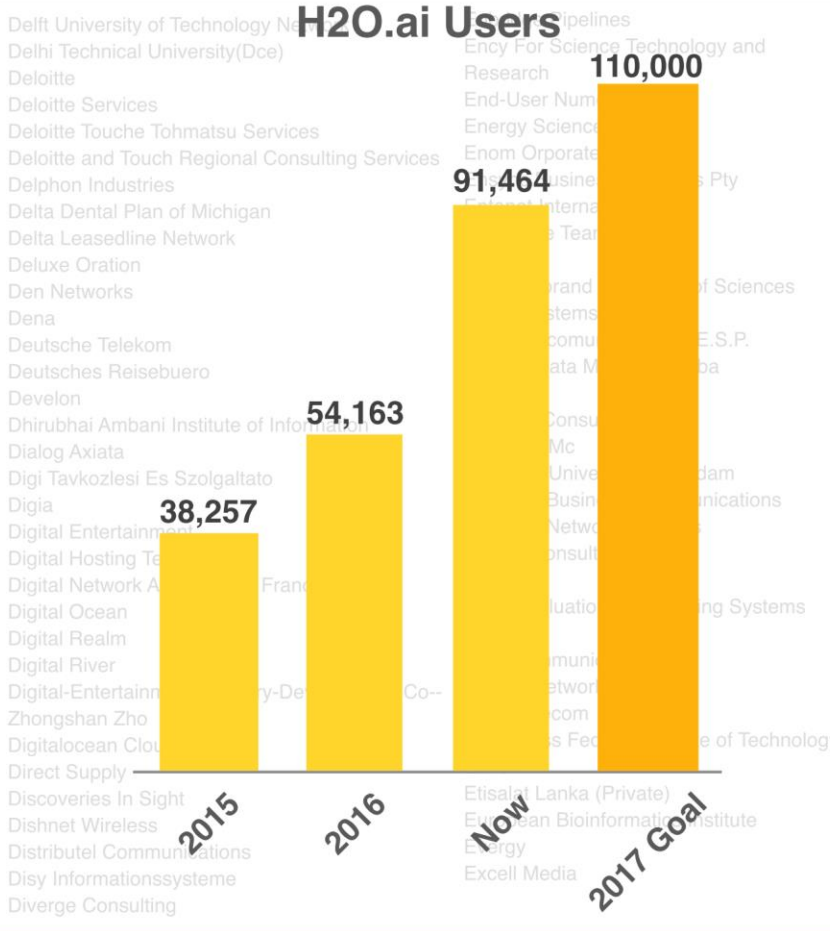
- Professor of Electrical Engineering and Computer Science, Stanford University
- PhD in Electrical Engineering and Computer Science, UC Berkeley
- Co-author, *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*
- Co-author, *Linear Matrix Inequalities in System and Control Theory*
- Co-author, *Convex Optimization*



H2O.ai

H2O Community & Fortune 100 customers



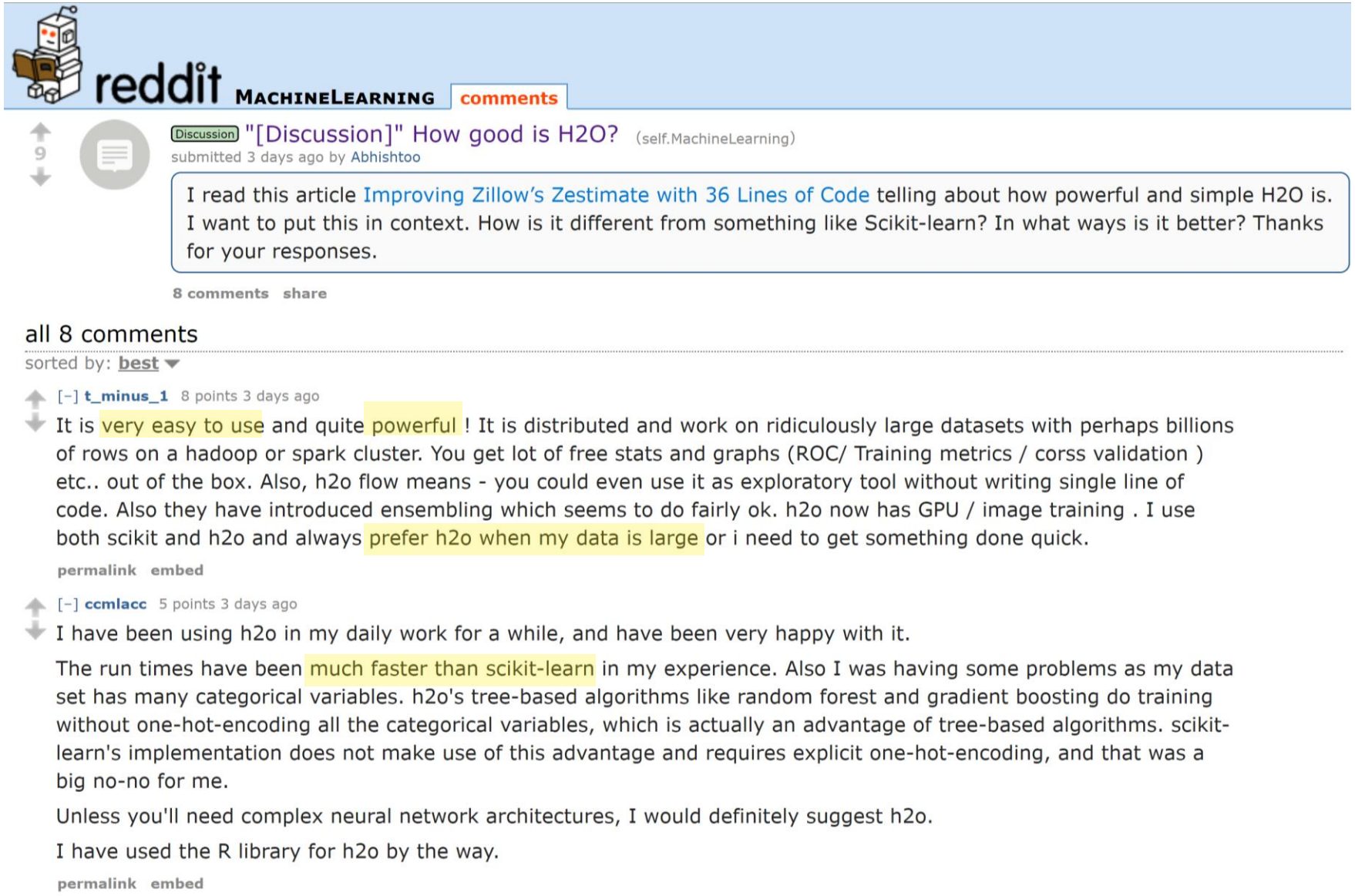


Select Reference Customers:

"Overall customer satisfaction is very high." - Gartner

CapitalOne PayPal Experian WELLS FARGO ADP COMCAST at&t KAISER PERMANENTE PROGRESSIVE Nielsen Catalina Booking.com macy's H2O.ai

Why H_2O?



reddit MACHINELEARNING comments

9

Discussion "[Discussion]" How good is H2O? (self.MachineLearning)

submitted 3 days ago by Abhishtoo

I read this article Improving Zillow's Zestimate with 36 Lines of Code telling about how powerful and simple H2O is. I want to put this in context. How is it different from something like Scikit-learn? In what ways is it better? Thanks for your responses.

8 comments share

all 8 comments

sorted by: best

[−] t_minus_1 8 points 3 days ago

It is very easy to use and quite powerful ! It is distributed and work on ridiculously large datasets with perhaps billions of rows on a hadoop or spark cluster. You get lot of free stats and graphs (ROC/ Training metrics / corss validation) etc.. out of the box. Also, h2o flow means - you could even use it as exploratory tool without writing single line of code. Also they have introduced ensembling which seems to do fairly ok. h2o now has GPU / image training . I use both scikit and h2o and always prefer h2o when my data is large or i need to get something done quick.

permalink embed

[−] ccmlacc 5 points 3 days ago

I have been using h2o in my daily work for a while, and have been very happy with it.

The run times have been much faster than scikit-learn in my experience. Also I was having some problems as my data set has many categorical variables. h2o's tree-based algorithms like random forest and gradient boosting do training without one-hot-encoding all the categorical variables, which is actually an advantage of tree-based algorithms. scikit-learn's implementation does not make use of this advantage and requires explicit one-hot-encoding, and that was a big no-no for me.

Unless you'll need complex neural network architectures, I would definitely suggest h2o.

I have used the R library for h2o by the way.

permalink embed

Very easy to use

Powerful

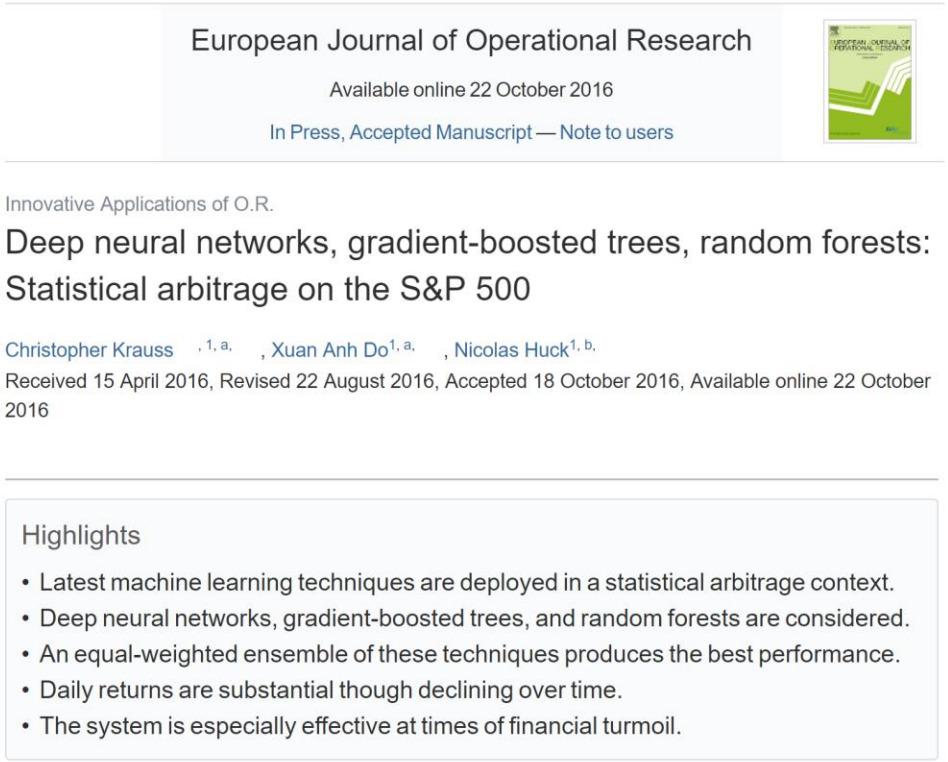
Prefer H2O when data is large

Much faster than scikit-learn

https://www.reddit.com/r/MachineLearning/comments/6g5yjh/discussion_how_good_is_h2o/

H2O.ai

H_2O for Academic Research



European Journal of Operational Research

Available online 22 October 2016

In Press, Accepted Manuscript — Note to users

Innovative Applications of O.R.

Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500

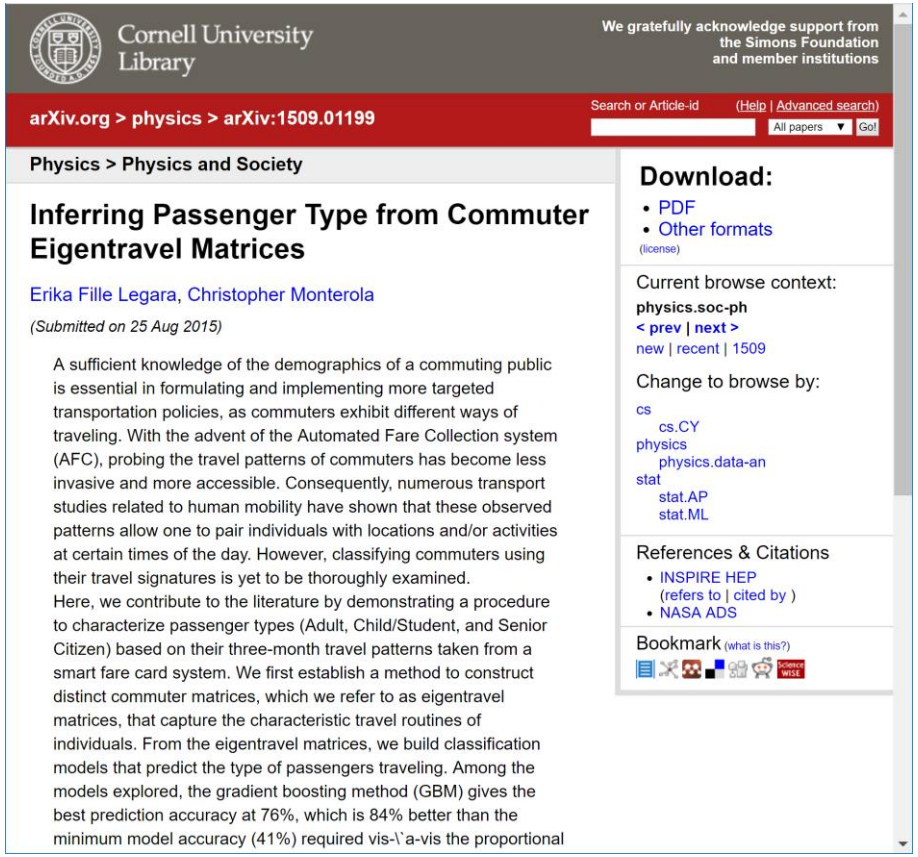
Christopher Krauss [1, a], Xuan Anh Do [1, a], Nicolas Huck [1, b]

Received 15 April 2016, Revised 22 August 2016, Accepted 18 October 2016, Available online 22 October 2016

Highlights

- Latest machine learning techniques are deployed in a statistical arbitrage context.
- Deep neural networks, gradient-boosted trees, and random forests are considered.
- An equal-weighted ensemble of these techniques produces the best performance.
- Daily returns are substantial though declining over time.
- The system is especially effective at times of financial turmoil.

http://www.sciencedirect.com/science/article/pii/S0377221716308657



Cornell University Library

We gratefully acknowledge support from the Simons Foundation and member institutions

arXiv.org > physics > arXiv:1509.01199

Search or Article-id (Help | Advanced search) All papers Go!

Physics > Physics and Society

Inferring Passenger Type from Commuter Eigentravel Matrices

Erika Fille Legara, Christopher Monterola

(Submitted on 25 Aug 2015)

A sufficient knowledge of the demographics of a commuting public is essential in formulating and implementing more targeted transportation policies, as commuters exhibit different ways of traveling. With the advent of the Automated Fare Collection system (AFC), probing the travel patterns of commuters has become less invasive and more accessible. Consequently, numerous transport studies related to human mobility have shown that these observed patterns allow one to pair individuals with locations and/or activities at certain times of the day. However, classifying commuters using their travel signatures is yet to be thoroughly examined. Here, we contribute to the literature by demonstrating a procedure to characterize passenger types (Adult, Child/Student, and Senior Citizen) based on their three-month travel patterns taken from a smart fare card system. We first establish a method to construct distinct commuter matrices, which we refer to as eigentravel matrices, that capture the characteristic travel routines of individuals. From the eigentravel matrices, we build classification models that predict the type of passengers traveling. Among the models explored, the gradient boosting method (GBM) gives the best prediction accuracy at 76%, which is 84% better than the minimum model accuracy (41%) required vis-\`a-vis the proportional

Download:

- PDF
- Other formats

(license)

Current browse context: physics.soc-ph

< prev | next >

new | recent | 1509

Change to browse by:

cs
cs.CY
physics
physics.data-an
stat
stat.AP
stat.ML

References & Citations

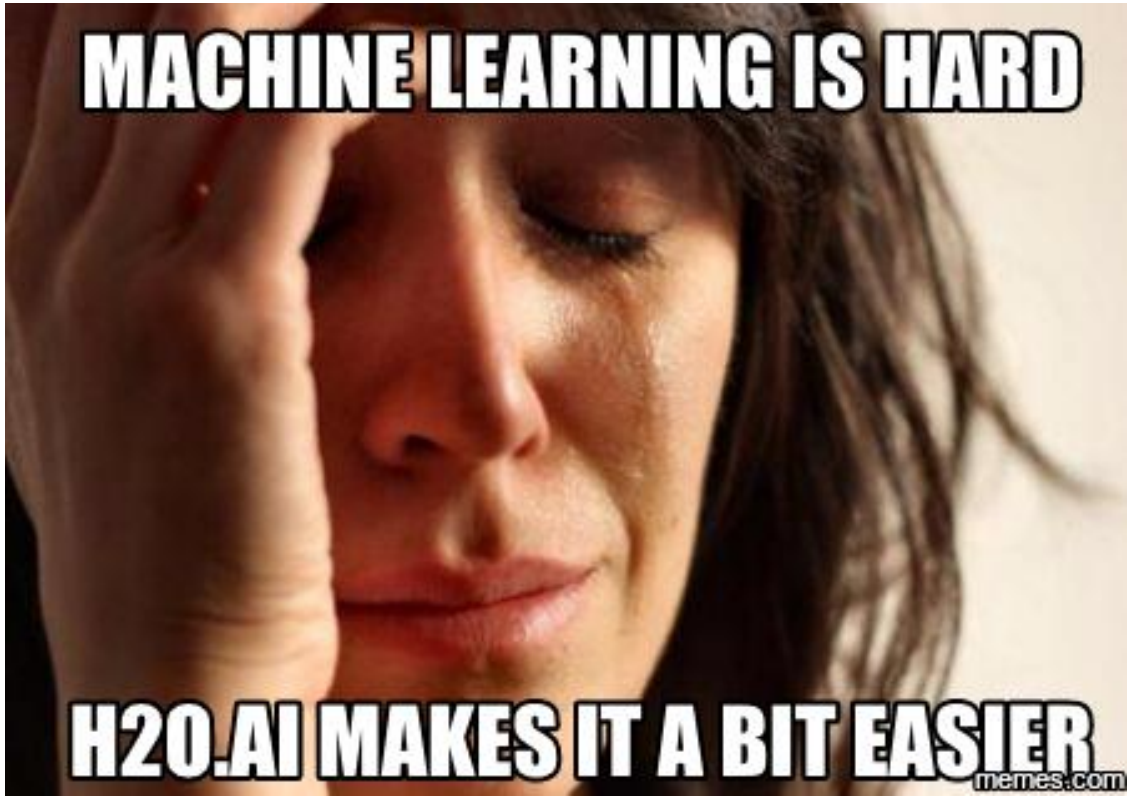
- INSPIRE HEP (refers to | cited by)
- NASA ADS

Bookmark (what is this?)

https://arxiv.org/abs/1509.01199

H_2O.ai

Szilard Pafka's Comment



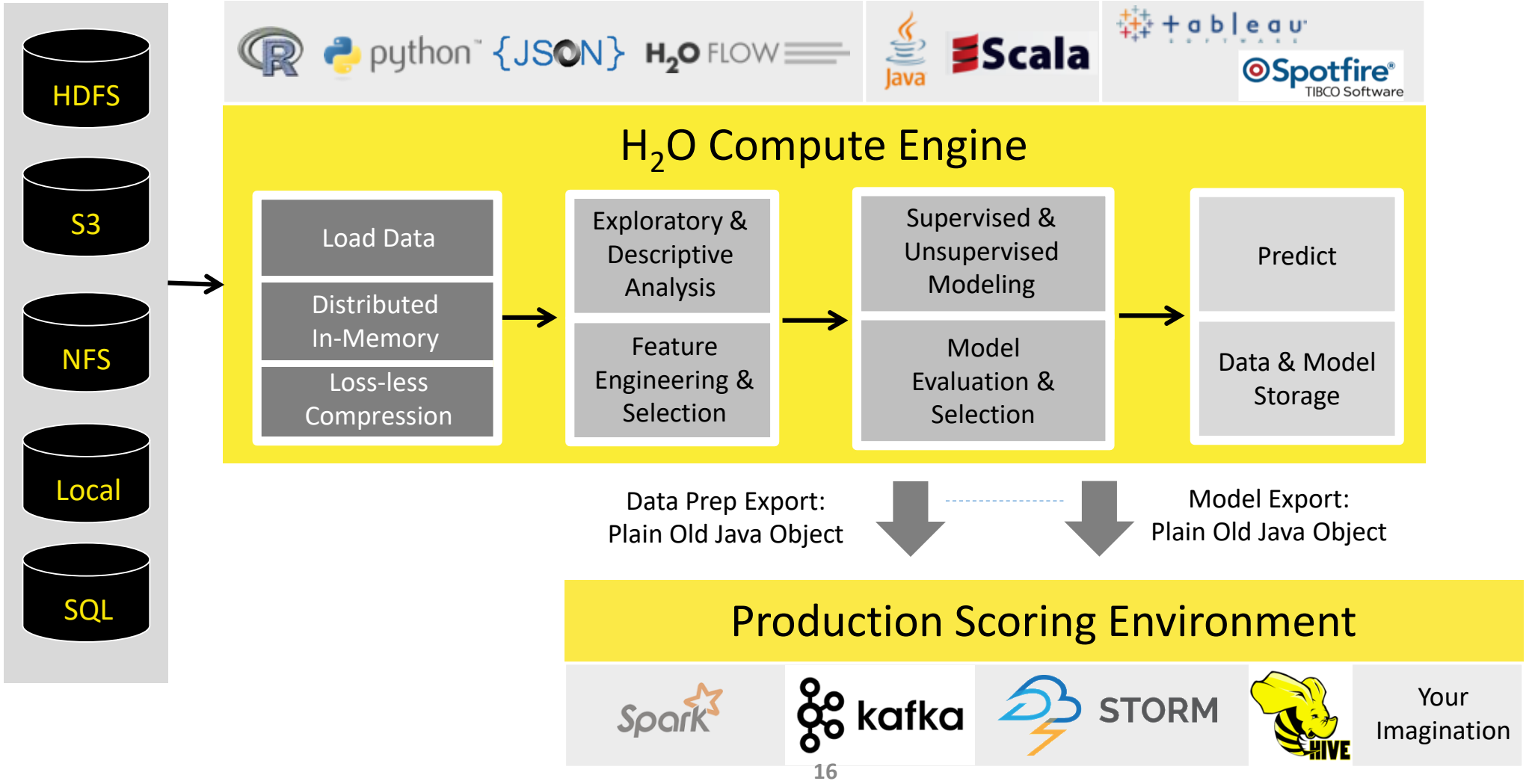
https://speakerdeck.com/szilard/machine-learning-with-h2o-dot-ai-budapest-data-science-meetup-july-2016

https://github.com/szilard/benchm-ml

H2O.ai

H_2O Machine Learning Platform

High Level Architecture



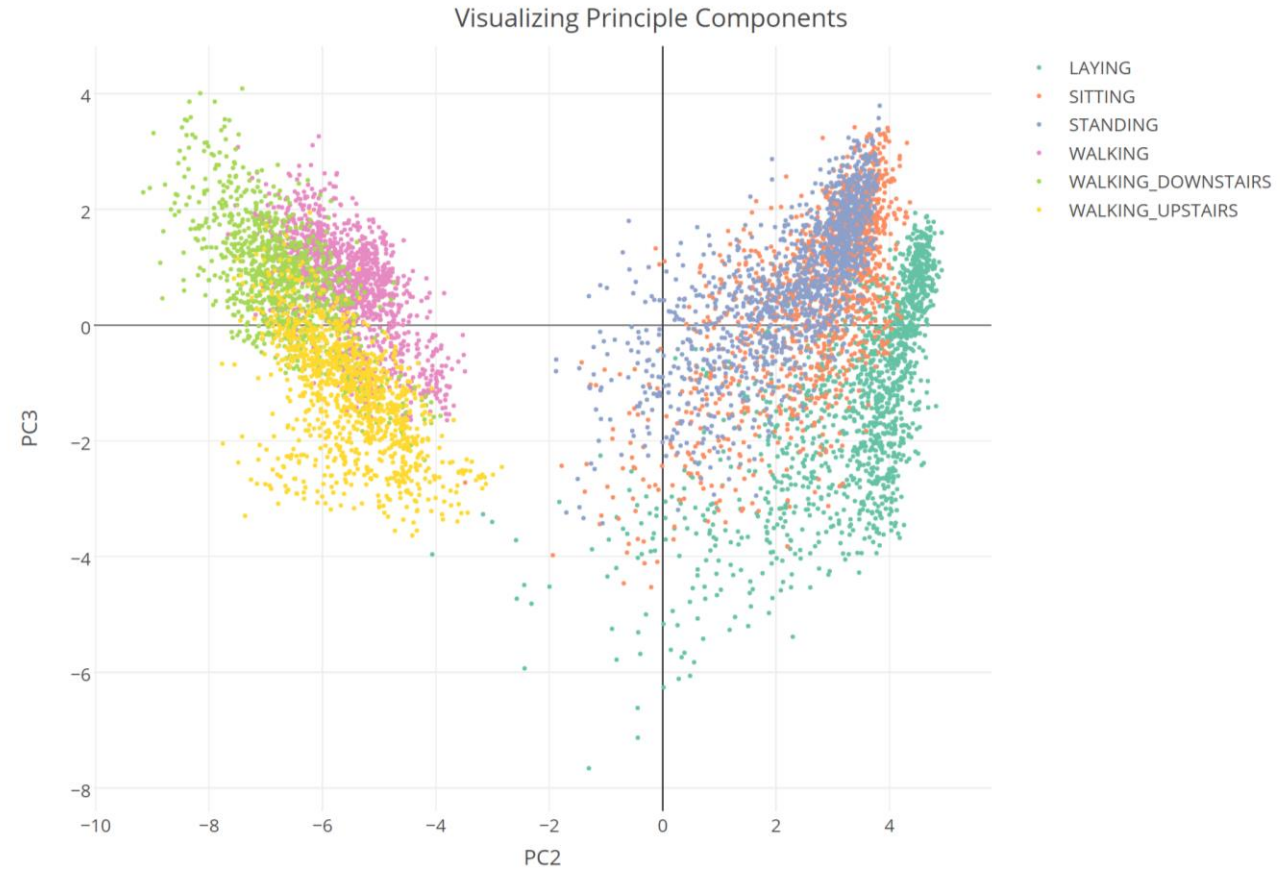
H_2O + R

```
# ------------------------------------------------------------
# Train a H2O Model
# ------------------------------------------------------------

# Train three basic H2O models
model_drf <- h2o.randomForest(x = features,
                              y = target,
                              model_id = "iris_random_forest",
                              training_frame = d_iris)

model_gbm <- h2o.gbm(x = features,
                     y = target,
                     model_id = "iris_gbm",
                     training_frame = d_iris)

model_dnn <- h2o.deeplearning(x = features,
                              y = target,
                              model_id = "iris_deep_learning",
                              training_frame = d_iris)
```

Use H_2O with Other R Packages

1. Train a PCA model with H_2O
2. Visualize PCs with plotly
3. Deploy as a Shiny app

```
p <- plot_ly(data = d_pca, x = ~PC2, y = ~PC3, color = ~activity,
             type = "scatter", mode = "markers", marker = list(size = 3)) %>%
     layout(title = "Visualizing Principle Components")
p
```



From the graph above, we can see that:

- it could be difficult to distinguish between **Standing** and **Sitting** as there are large overlaps in their sensor data.
- **Laying** has its own cluster so it should be easy to classify.
- **Walking**, **Walking Upstairs** and **Walking Downstairs** are understandably closer to each other yet they are quite different to **Sitting**, **Standing** and **Laying**.

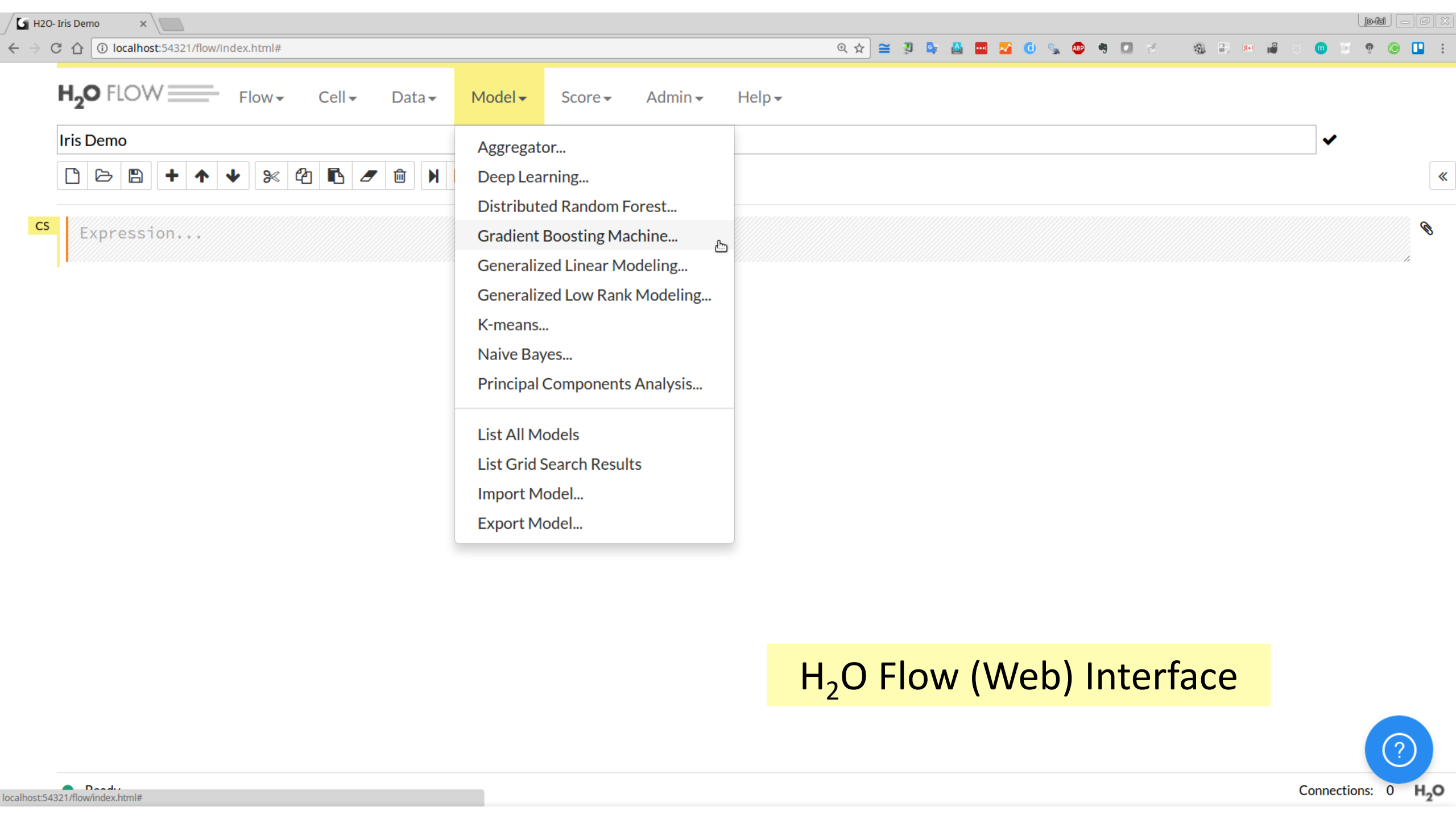
https://github.com/woobe/h2o_demos/tree/master/human_activitiy_recognition_with_smartphones

H_2O + Python

Gradient Boosting Machines

```
# Build a Gradient Boosting Machines (GBM) model with default settings

# Import the function for GBM
from h2o.estimators.gbm import H2OGradientBoostingEstimator

# Set up GBM for regression
# Add a seed for reproducibility
gbm_default = H2OGradientBoostingEstimator(model_id = 'gbm_default', seed = 1234)

# Use .train() to build the model
gbm_default.train(x = features,
                  y = 'quality',
                  training_frame = wine_train)
```

```
gbm Model Build progress: |████████████████████████████████████████████████| 100%
```



H_2O Flow (Web) Interface

docs.h2o.ai

Languages

R

- Quick Start Video - R
- R Package Docs
- R Booklet
- Examples and Demos
- R FAQ
- Ensemble R Package Readme
- RSparkling Readme
- Migrating from H2O-2

Python

- Quick Start Video - Python
- Python Module Docs
- Python Booklet
- Examples and Demos
- Python FAQ
- PySparkling Readme 2.0 | 1.6
- skutil Docs

Java

- POJO and MOJO Model Javadoc
- H2O Core Javadoc
- H2O Algorithms Javadoc

Scala

Sparkling Water API	2.0	1.6
Sparkling Water Scaladoc	2.0	1.6
H2O Scaladoc	2.11	2.10

Tutorials, Examples, & Presentations

Tutorials and Blogs

- H2O Tutorials HTML | PDF
- H2O Blogs
- H2O University

Use Case Examples

Chicago crime prediction	R	Python	ScalaSW	PySW
Airlines delays prediction	R	Python	ScalaSW	PySW
Lending Club loan prediction	R	Python	ScalaSW	PySW
Ham or Spam	R	Python	ScalaSW	PySW
Prediction with prostate dataset	R	Python	ScalaSW	PySW

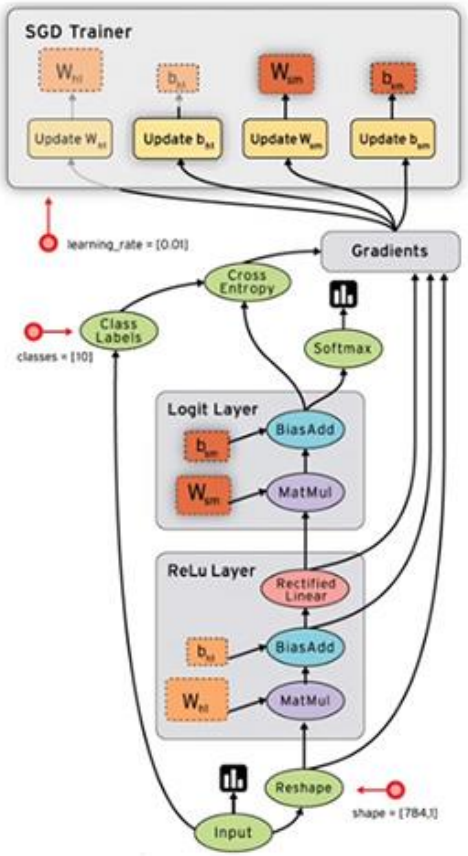
Presentations

- H2O Meetups
- H2O World 2014 Videos
- H2O World 2015 Videos
- Open Tour Chicago Videos
- Open Tour NYC Videos
- Open Tour Dallas Videos


H2O.ai
Products
Industries
Customers
Community
Documentation
Download
Deep Water
H2O.ai
Caffe
mxnet
TensorFlow

TensorFlow

- Open source machine learning framework by Google
- Python / C++ API
- TensorBoard
  - Data Flow Graph Visualization
- Multi CPU / GPU
  - v0.8+ distributed machines support
- Multi devices support
  - desktop, server and Android devices
- Image, audio and NLP applications
- **HUGE** Community
- Support for Spark, Windows …



https://github.com/tensorflow/tensorflow

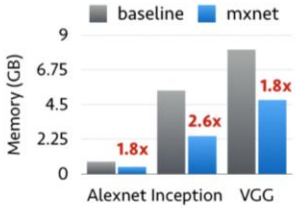
H2O.ai

dmlc mxnet for Deep Learning

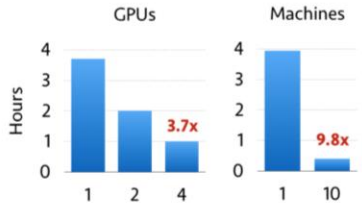
build passing | docs latest | license Apache 2.0

Portable

Efficient



Scalable



MXNet is a deep learning framework designed for both *efficiency* and *flexibility*. It allows you to ***mix*** the flavours of symbolic programming and imperative programming to ***maximize*** efficiency and productivity. In its core, a dynamic dependency scheduler that automatically parallelizes both symbolic and imperative operations on the fly. A graph optimization layer on top of that makes symbolic execution fast and memory efficient. The library is portable and lightweight, and it scales to multiple GPUs and multiple machines.

MXNet is also more than a deep learning project. It is also a collection of blue prints and guidelines for building deep learning system, and interesting insights of DL systems for hackers.

MXNet now chosen by Amazon as Deep Learning Framework

By Geneva Clark | 2016-11-24

19 0

Share this magazine

Amazon has announced that it has chosen MXNet as its deep learning framework of choice for its web services(AWS). Amazon extensively uses machine learning in areas like fraud detection, abusive review detection, and book classification. Amazon also uses it in application areas such as text and speech recognition, autonomous drones etc...

https://github.com/dmlc/mxnet
https://www.zeolearn.com/magazine/amazon-to-use-mxnet-as-deep-learning-framework

H2O.ai

Caffe

- **C**onvolution **A**rchitecture **F**or **F**eature **E**xtraction (CAFFE)
- Pure C++ / CUDA architecture for deep learning
- Command line, Python and MATLAB interface
- Model Zoo
  - Open collection of models

https://docs.google.com/presentation/d/1UeKXVgRvvxg9OUdh_UiC5G71UMscNPlvArsWER41PsU/

H2O.ai

H_2O Deep Learning in Action

116M rows, 6GB CSV file
800+ predictors (numeric + categorical)

airlines_all_selected_cols.hex

Actions: View Data | Split... | Build Model... | Predict | Download | Export

Rows	Columns	Compressed Size
116695259	12	2GB

Job

Run Time 00:00:36.712
Remaining Time 00:00:17.188
Type Model
Key deeplearning-dd2f42f7-81f7-42e8-9d98-e34437309828
Description DeepLearning
Status RUNNING
Progress 69%

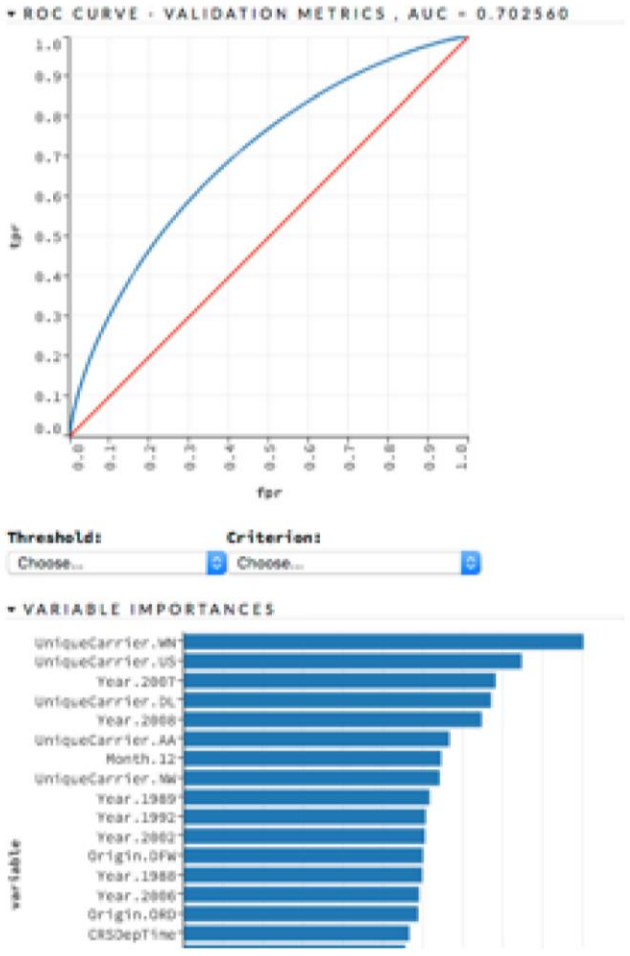
Iterations: 12. Epochs: 0.628821. Speed: 2,243,735 samples/sec. Estimated time left: 21.849 sec

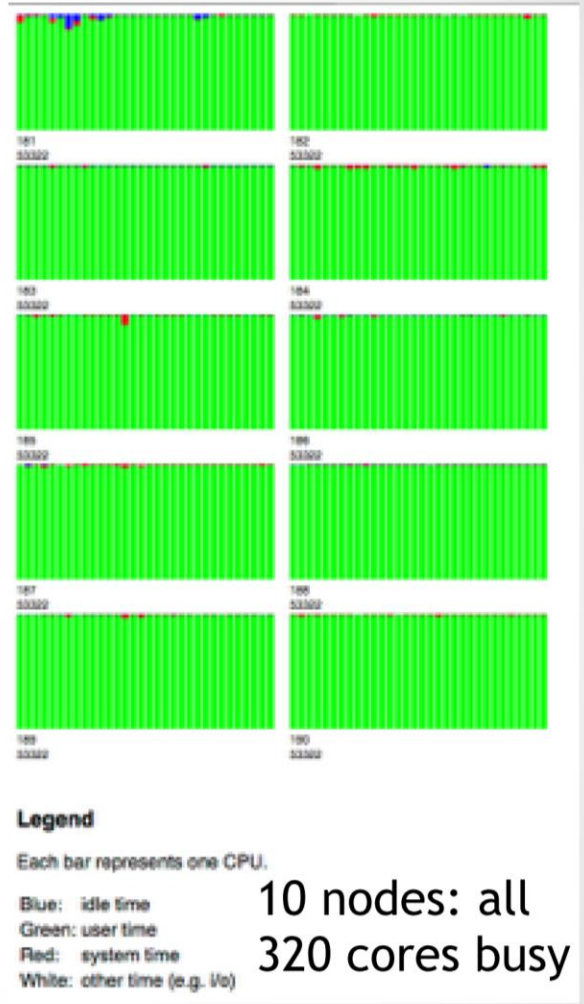
Actions: View | Cancel Job

model trained in <1 min:
2M+ samples/second

▾ OUTPUT - STATUS OF NEURON LAYERS (PREDICTING ISDEPDELAYED, 2-CLASS CLASSIFICATION, BERNOULLI DISTRIBUTION, CROSSENTROPY LOSS, 17,462 WEIGHTS/BIASES, 221.3 KB, 106,585,365 TRAINING SAMPLES, MINI-BATCH SIZE 1)

layer	units	type	dropout	l1	l2	mean_rate	rate_RMS	momentum	mean_weight	weight_RMS	mean_bias	bias_RMS
1	807	Input	0									
2	20	Rectifier	0	0	0	0.0493	0.2020	0	-0.0021	0.2111	-0.9139	1.0036
3	20	Rectifier	0	0	0	0.0197	0.0227	0	-0.1033	0.5362	-1.3988	1.5259
4	20	Rectifier	0	0	0	0.0517	0.0446	0	-0.1575	0.3068	-0.6846	0.6846
5	20	Rectifier	0	0	0	0.0761	0.0844	0	-0.0374	0.2275	-0.2647	0.2481
6	2	Softmax		0	0	0.0161	0.0083	0	0.0741	0.7260	0.4269	0.2056





10 nodes: all
320 cores busy

real-time, interactive
model inspection in Flow

H_2O.ai **Deep Learning Model**

TensorFlow , MXNet, Caffe and H_2O DL democratize the power of deep learning.

H_2O platform democratizes artificial intelligence & big data science.

There are other open source deep learning libraries like Theano and Torch too.

Let's have a party, this will be fun!

H_2O.ai

Deep Water

Next-Gen Distributed Deep Learning with H_2O

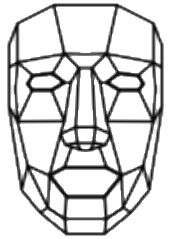
One Interface - GPU Enabled - Significant Performance Gains

Inherits All H_2O Properties in Scalability, Ease of Use and Deployment

TensorFlow Caffe H_2O.ai mxnet

Deep Water

H_2O integrates with existing GPU backends for significant performance gains



Convolutional Neural Networks enabling **Image, video, speech recognition**

Recurrent Neural Networks enabling **natural language processing, sequences, time series,** and more

Hybrid Neural Network Architectures enabling **speech to text translation, image captioning, scene parsing** and more

H_2O.ai

H_2O

H2O works with R, Python, Scala on Hadoop/Yarn, Spark or your laptop

The best machine learning platform on Spark and Hadoop

Interface using R, Python or intuitive web UI - Flow

Open Source Software, Apache 2.0 licensed

Data agnostic support for all common data and file types

Nanofast scoring engine

Export models as code or binary model files

Download H_2O Latest Stable Release

Download H_2O Nightly Bleeding Edge

H_2O is licensed under the Apache License, Version 2.0

H_2O with GPU-Enabled Machine Learning

H2O Deep Water Deep Learning with MXNet, TensorFlow and Caffe XGBoost Gradient Boosting

Requirements:
- Ubuntu 16.04
- CUDA 8.0
- cuDNN 5.1

H2O: wget http://s3.amazonaws.com/h2o-deepwater/public/nightly/latest/h2o.jar
java -jar h2o.jar

Python: pip install http://s3.amazonaws.com/h2o-deepwater/public/nightly/latest/h2o-3.11.0-py2.py3-none-any.whl

R: wget http://s3.amazonaws.com/h2o-deepwater/public/nightly/latest/h2o_3.11.0.tar.gz
R CMD INSTALL h2o_3.11.0.tar.gz

Docker image: Installation instruction: https://github.com/h2oai/deepwater#pre-release-docker-image

Sparkling Water

H2O Sparkling Water works with Spark 1.6 and Spark 2.0

The best machine learning platform on Spark and Hadoop

H2O - the killer app for Spark

Seamlessly transition back and forth between Spark and H2O

Open Source Software, Apache 2.0 licensed

Use Scala or Python to build models

Power of Spark combined with the speed of H2O

All the features of H2O included (Flow - UI, model export etc.)

Download Sparkling Water 2.1

Download Sparkling Water 2.0

Sparkling Water is licensed under the Apache License, Version 2.0

Steam

Build and deploy smart applications

The best machine learning platform on Spark and Hadoop

Start H2O Clusters on Yarn

Save models, inspect model metrics and compare models

Deploy models as pojo or war files

Deploy scoring service with pre-processing scripts

Open Source Software, AGPL 3.0 licensed

Download Steam

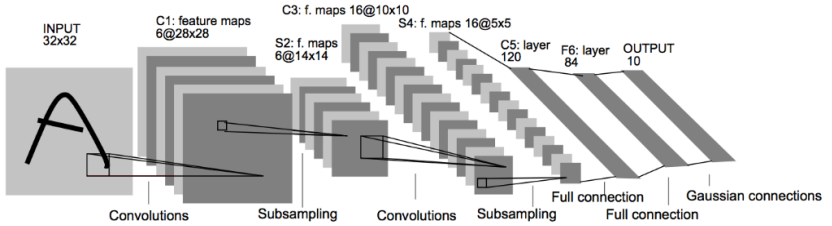
Steam is released under the Open Source AGPL license, Version 3.0. Customers, OEMs, ISVs and VARs can purchase commercial licenses.

www.h2o.ai/download/

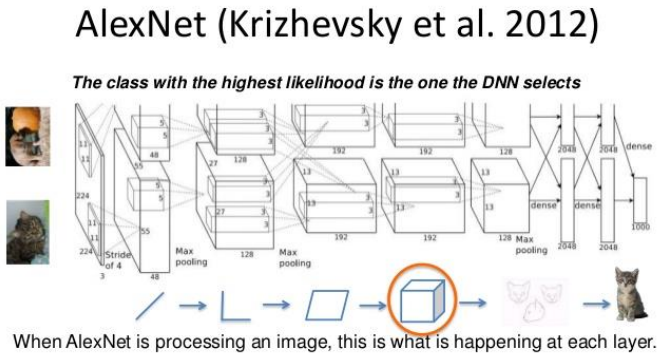
H_2O.ai

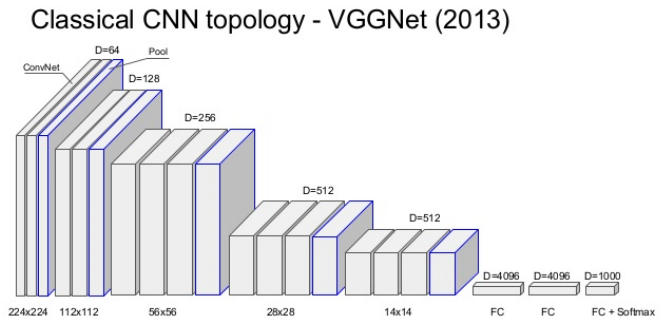
Available Networks in Deep Water

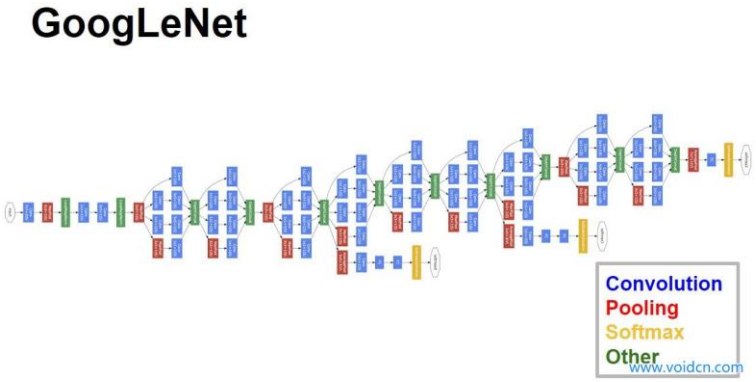
- LeNet
- AlexNet
- VGGNet
- Inception (GoogLeNet)
- ResNet (Deep Residual Learning)
- Build Your Own

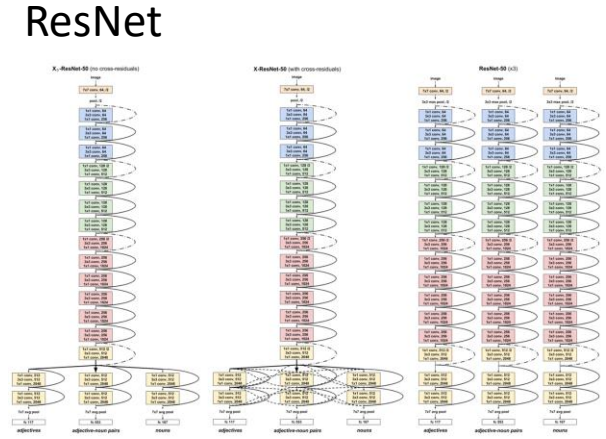


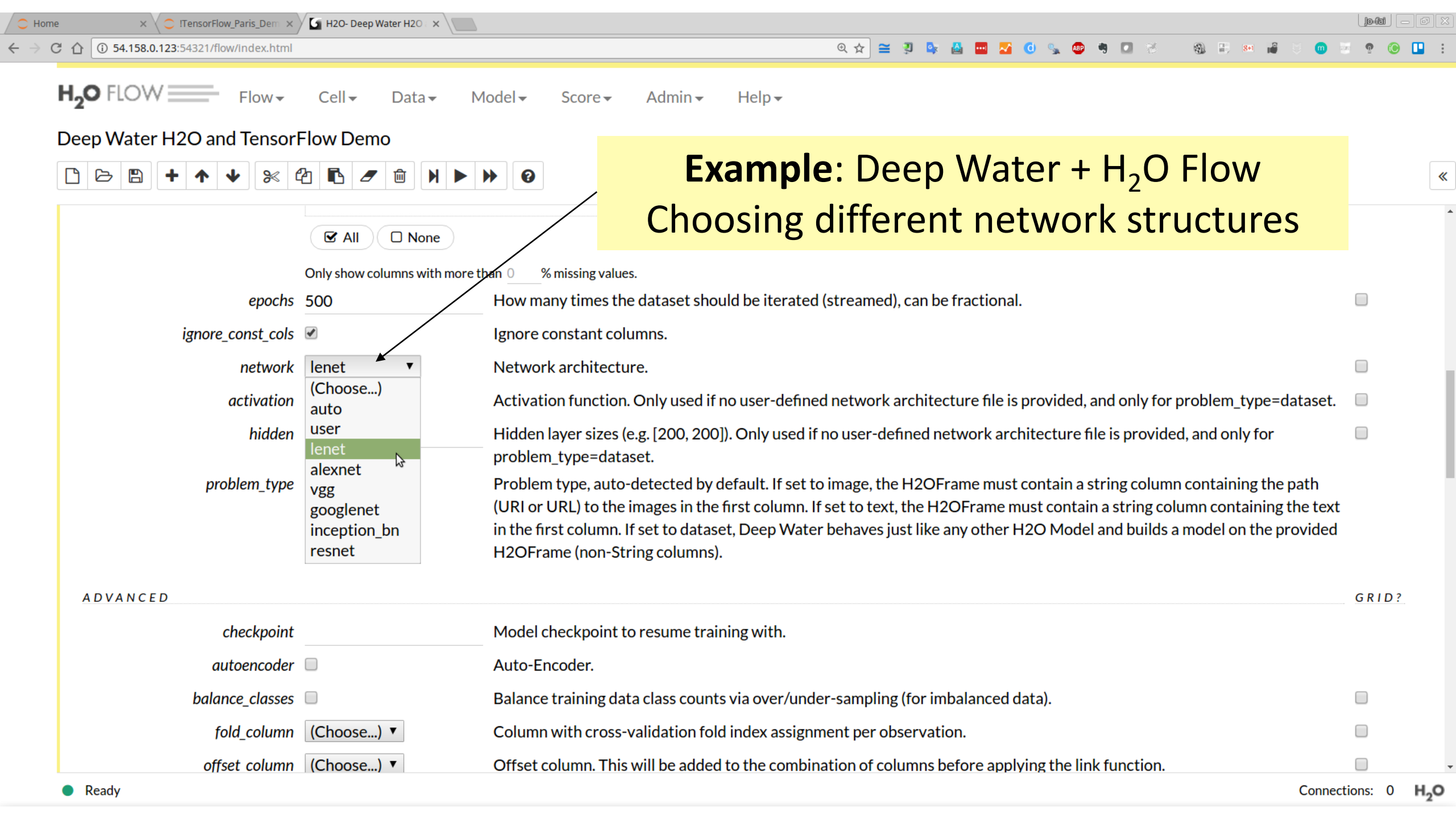
CNN called LeNet by Yann LeCun (1998)

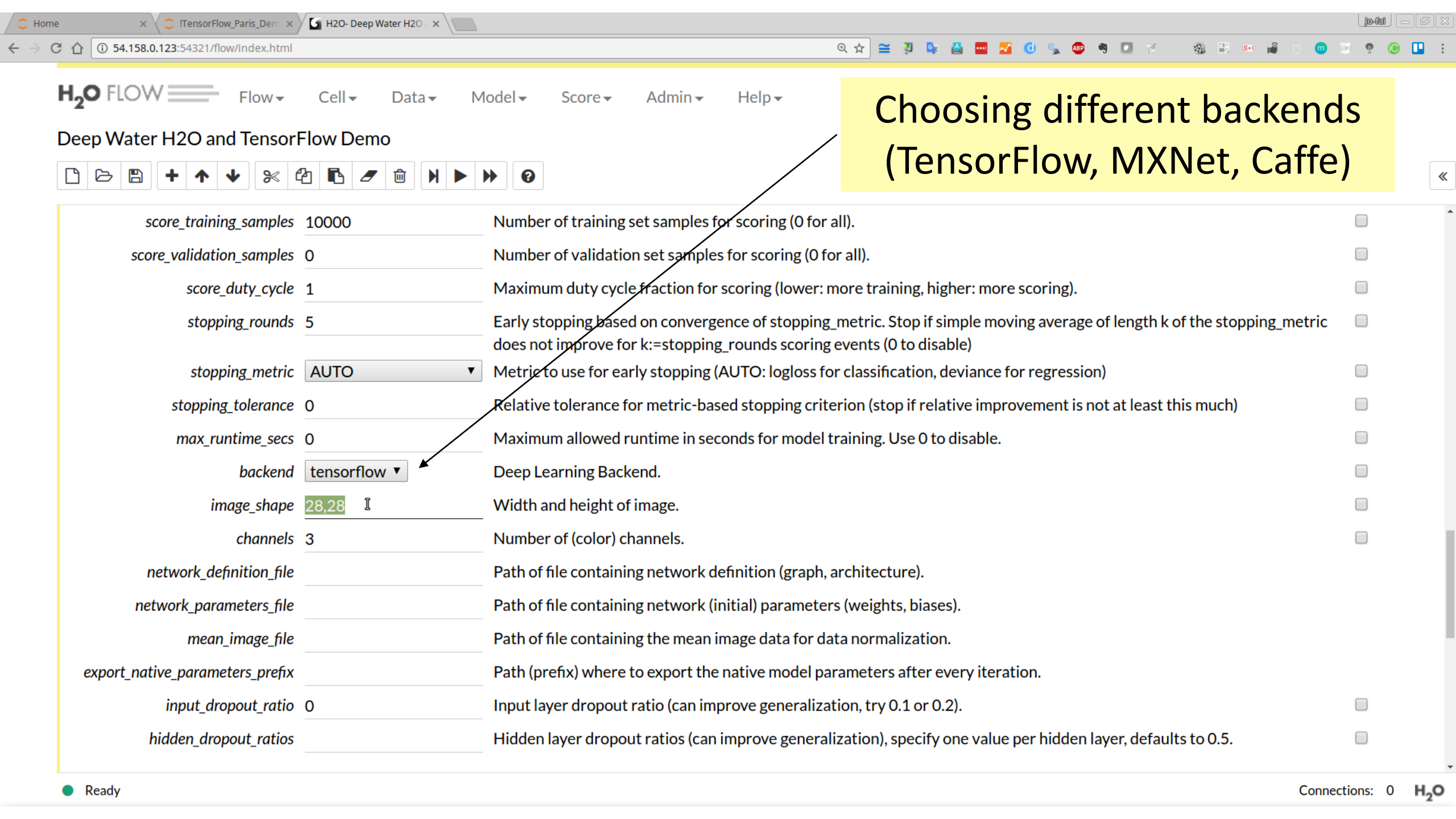









Home
ITensorFlow_Paris_Dem
H2O- Deep Water H2O
54.158.0.123:54321/flow/index.html
H2O FLOW
Flow
Cell
Data
Model
Score
Admin
Help
Deep Water H2O and TensorFlow Demo
Example: Deep Water + H2O Flow
Choosing different network structures
All
None
Only show columns with more than 0 % missing values.
epochs
500
How many times the dataset should be iterated (streamed), can be fractional.
ignore_const_cols
Ignore constant columns.
network
lenet
Network architecture.
activation
Activation function. Only used if no user-defined network architecture file is provided, and only for problem_type=dataset.
hidden
Hidden layer sizes (e.g. [200, 200]). Only used if no user-defined network architecture file is provided, and only for problem_type=dataset.
problem_type
Problem type, auto-detected by default. If set to image, the H2OFrame must contain a string column containing the path (URI or URL) to the images in the first column. If set to text, the H2OFrame must contain a string column containing the text in the first column. If set to dataset, Deep Water behaves just like any other H2O Model and builds a model on the provided H2OFrame (non-String columns).
(Choose...)
auto
user
lenet
alexnet
vgg
googlenet
inception_bn
resnet
ADVANCED
GRID?
checkpoint
Model checkpoint to resume training with.
autoencoder
Auto-Encoder.
balance_classes
Balance training data class counts via over/under-sampling (for imbalanced data).
fold_column
(Choose...)
Column with cross-validation fold index assignment per observation.
offset_column
(Choose...)
Offset column. This will be added to the combination of columns before applying the link function.
Ready
Connections: 0
H2O


Home
ITensorFlow_Paris_Dem
H2O- Deep Water H2O
54.158.0.123:54321/flow/index.html
H2O FLOW
Flow
Cell
Data
Model
Score
Admin
Help
Deep Water H2O and TensorFlow Demo
Choosing different backends
(TensorFlow, MXNet, Caffe)
score_training_samples 10000 Number of training set samples for scoring (0 for all).
score_validation_samples 0 Number of validation set samples for scoring (0 for all).
score_duty_cycle 1 Maximum duty cycle fraction for scoring (lower: more training, higher: more scoring).
stopping_rounds 5 Early stopping based on convergence of stopping_metric. Stop if simple moving average of length k of the stopping_metric does not improve for k:=stopping_rounds scoring events (0 to disable)
stopping_metric AUTO Metric to use for early stopping (AUTO: logloss for classification, deviance for regression)
stopping_tolerance 0 Relative tolerance for metric-based stopping criterion (stop if relative improvement is not at least this much)
max_runtime_secs 0 Maximum allowed runtime in seconds for model training. Use 0 to disable.
backend tensorflow Deep Learning Backend.
image_shape 28,28 Width and height of image.
channels 3 Number of (color) channels.
network_definition_file Path of file containing network definition (graph, architecture).
network_parameters_file Path of file containing network (initial) parameters (weights, biases).
mean_image_file Path of file containing the mean image data for data normalization.
export_native_parameters_prefix Path (prefix) where to export the native model parameters after every iteration.
input_dropout_ratio 0 Input layer dropout ratio (can improve generalization, try 0.1 or 0.2).
hidden_dropout_ratios Hidden layer dropout ratios (can improve generalization), specify one value per hidden layer, defaults to 0.5.
Ready
Connections: 0
H2O

Unified Interface (Deep Water + R)

```
# Train a LeNet with basic parameters and MXNet
model_mxnet <- h2o.deepwater(x = path,
                             y = response,
                             training_frame = df,
                             epochs = 300,
                             learning_rate = 1e-3,
                             image_shape = c(28, 28),
                             channels = 3,
                             backend = "mxnet",
                             network = "lenet")
```

Choosing different network structures and backends

Unified Interface (Deep Water + Python)

Choosing different network structures

```
model = H2ODeepWaterEstimator(epochs      = 500,
                              network     = "lenet",
                              image_shape = [28,28],  ## provide image size
                              channels    = 3,
                              backend     = "tensorflow",
                              model_id    = "deepwater_tf_simple")

model.train(x = [0], # file path e.g. xxx/xxx/xxx.jpg
            y = 1, # label cat/dog/mouse
            training_frame = frame)

model.show()
```

Change backend to “mxnet”, “caffe” or “auto”

```
deepwater Model Build progress: |████████████████████████████████████████| 100%
Model Details
=============
H2ODeepWaterEstimator :  Deep Water
Model Key:  deepwater_tf_simple
```

Branch: master ▾ h2o-3 / examples / deeplearning / notebooks / Create new file | Upload files | Find file | History

mstensmo changing the name of deeplearning_credit_card_default_risk_prediction... ... Latest commit 5568350 11 days ago

..		
images	Add cat/dog/mouse lenet example.	3 months ago
README.md	Update README.md	2 months ago
deeplearning_anomaly_detection.ip...	Update notebooks, introduce local paths to ~/h2o-3/	3 months ago
deeplearning_benchmark_mnist.ipynb	Update lenet test to remove all. Update MNIST benchmark with comments.	3 months ago
deeplearning_cat_dog_mouse_incep...	Add credit card default risk model, update other notebooks.	3 months ago
deeplearning_cat_dog_mouse_lenet....	Add credit card default risk model, update other notebooks.	
deeplearning_cat_dog_mouse_lenet...	Add back model.plot() and scoring history.	
deeplearning_cifar10_vgg.ipynb	Rename notebooks.	
deeplearning_credit_card_default_ri...	changing the name of deeplearning_credit_card_default_risk_prediction...	
deeplearning_ensemble_boston_ho...	Ensemble demo using GBM, DRF and Deep Water (#676)	
deeplearning_grid_iris.ipynb	Add two new notebooks: Lenet for R and iris grid for python	3 months ago
deeplearning_grid_iris_R.ipynb	Update R py notebook.	3 months ago
deeplearning_image_reconstruction...	Update notebooks, introduce local paths to ~/h2o-3/	3 months ago
deeplearning_mnist_convnet.ipynb	Update notebooks, introduce local paths to ~/h2o-3/	3 months ago
deeplearning_mnist_introduction.ip...	Add missing file.	3 months ago
deeplearning_tensorflow_cat_dog_...	Add tensorflow example (#529)	2 months ago
deeplearning_tensorflow_mnist.ipynb	Added MNIST example for TensorFlow	a month ago

Deep Water Example notebooks

https://github.com/h2oai/h2o-3/tree/master/examples/deeplearning/notebooks

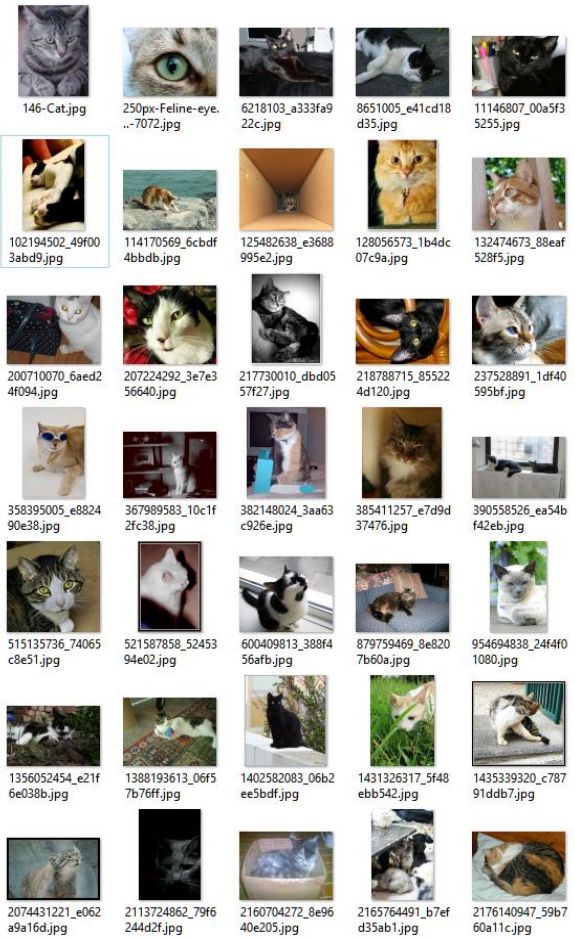
H2O.ai

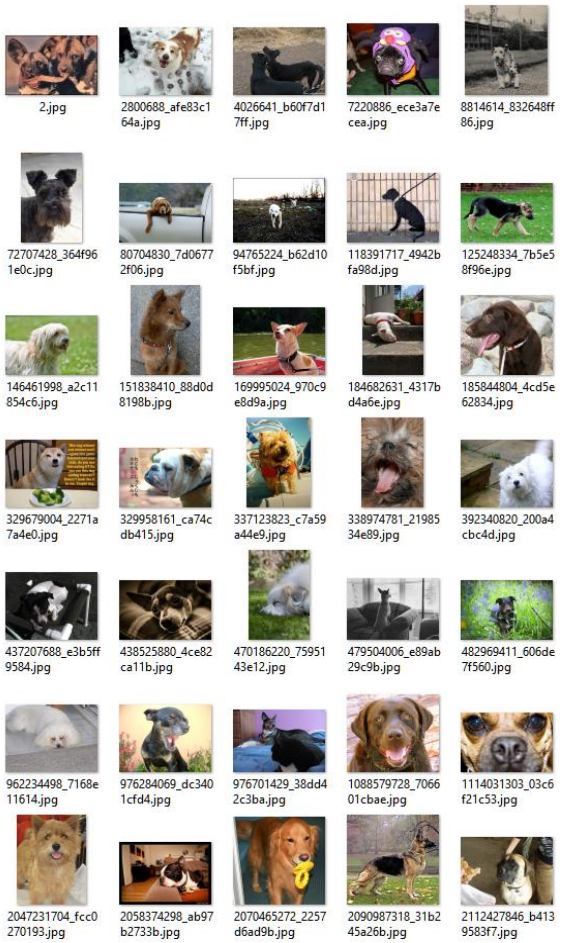
Deep Water Cat/Dog/Mouse Demo

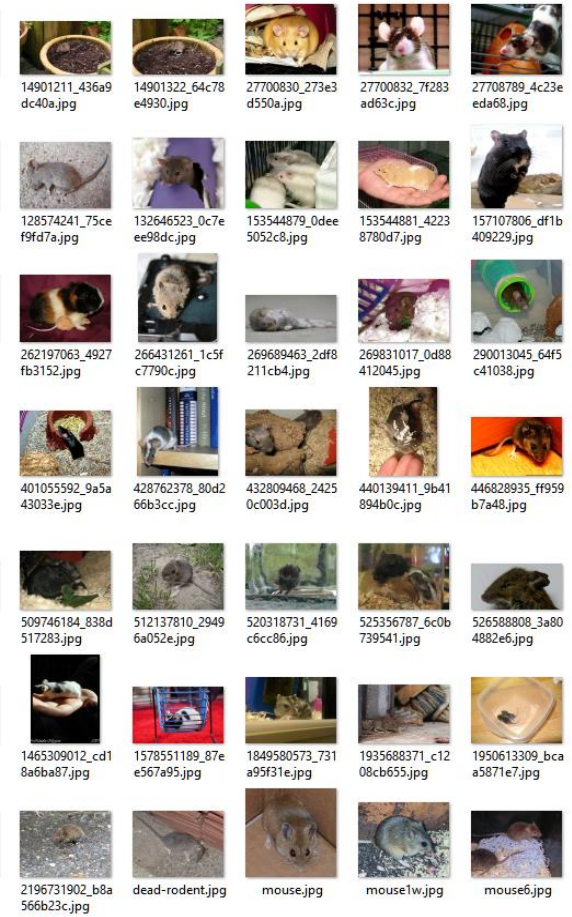
Deep Water R Demo

- H_2O + MXNet + TensorFlow
  - Dataset – Cat/Dog/Mouse
  - MXNet & TensorFlow as GPU backend
  - Train LeNet (CNN) models
  - R Demo
- Code and Data
  - github.com/h2oai/deepwater

Data – Cat/Dog/Mouse Images







H2O.ai

Data – CSV

	A	B
1	bigdata/laptop/deepwater/imagenet/cat/102194502_49f003abd9.jpg	cat
2	bigdata/laptop/deepwater/imagenet/cat/11146807_00a5f35255.jpg	cat
3	bigdata/laptop/deepwater/imagenet/cat/1140846215_70e326f868.jpg	cat
4	bigdata/laptop/deepwater/imagenet/cat/114170569_6cbdf4bbdb.jpg	cat
5	bigdata/laptop/deepwater/imagenet/cat/1217664848_de4c7fc296.jpg	cat
6	bigdata/laptop/deepwater/imagenet/cat/1241603780_5e8c8f1ced.jpg	cat
7	bigdata/laptop/deepwater/imagenet/cat/1241612072_27ececbdef.jpg	cat
8	bigdata/laptop/deepwater/imagenet/cat/1241613138_ef1d82973f.jpg	cat
9	bigdata/laptop/deepwater/imagenet/cat/1244562192_35becd66bd.jpg	cat
10	bigdata/laptop/deepwater/imagenet/cat/125482638_e3688995e2.jpg	cat
11	bigdata/laptop/deepwater/imagenet/cat/128056573_1b4dc07c9a.jpg	cat
12	bigdata/laptop/deepwater/imagenet/cat/12945197_75e607e355.jpg	cat
13	bigdata/laptop/deepwater/imagenet/cat/132474673_88eaf528f5.jpg	cat
14	bigdata/laptop/deepwater/imagenet/cat/1350530984_ecf3039cf0.jpg	cat
15	bigdata/laptop/deepwater/imagenet/cat/1351606235_c9fbebf634.jpg	cat
16	bigdata/laptop/deepwater/imagenet/cat/1356052454_e21f6e038b.jpg	cat
17	bigdata/laptop/deepwater/imagenet/cat/1388193613_06f57b76ff.jpg	cat

Deep Water – Basic Usage

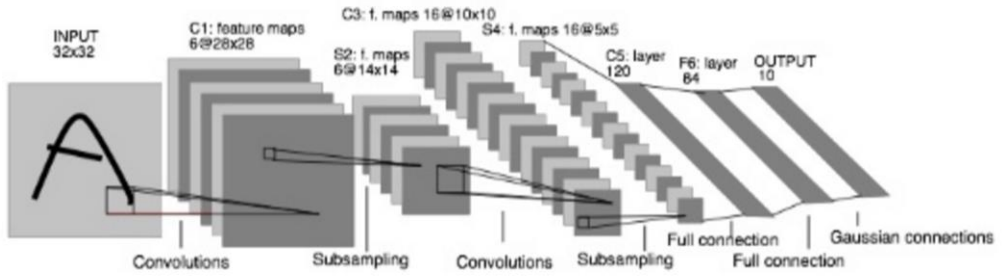
Live Demo if Possible

Import CSV

```
df <- h2o.importFile("/home/ubuntu/h2o-3/bigdata/laptop/deepwater/imagenet/cat_dog_mouse.csv")
print(head(df))
path = 1 ## must be the first column
response = 2
```

```
  |=============================================================| 100%
                                                          C1  C2
1  bigdata/laptop/deepwater/imagenet/cat/102194502_49f003abd9.jpg cat
2   bigdata/laptop/deepwater/imagenet/cat/11146807_00a5f35255.jpg cat
3 bigdata/laptop/deepwater/imagenet/cat/1140846215_70e326f868.jpg cat
4  bigdata/laptop/deepwater/imagenet/cat/114170569_6cbdf4bbdb.jpg cat
5 bigdata/laptop/deepwater/imagenet/cat/1217664848_de4c7fc296.jpg cat
6 bigdata/laptop/deepwater/imagenet/cat/1241603780_5e8c8f1ced.jpg cat
```

Train a CNN (LeNet) Model on GPU



LeNet: a layered model composed of convolution and subsampling operations followed by a holistic representation and ultimately a classifier for handwritten digits. [Yann LeCun; LeNet]

```
# Train a LeNet with basic parameters and MXNet
model_mxnet <- h2o.deepwater(x = path,
                             y = response,
                             training_frame = df,
                             epochs = 300,
                             learning_rate = 1e-3,
                             image_shape = c(28, 28),
                             channels = 3,
                             backend = "mxnet",
                             network = "lenet")

# Train a LeNet with basic parameters and TensorFlow
model_tf <- h2o.deepwater(x = path,
                          y = response,
                          training_frame = df,
                          epochs = 300,
                          learning_rate = 1e-3,
                          image_shape = c(28, 28),
                          channels = 3,
                          backend = "tensorflow",
                          network = "lenet")
```

Easy Switch

H2O.ai

Deep Water – Custom Network

If you'd like to build your own LeNet network architecture, then this is easy as well. In this example script, we are using the 'mxnet' backend. Models can easily be imported/exported between H2O and MXNet since H2O uses MXNet's format for model definition.

In [5]:
```
get_symbol <- function(num_classes = 1000) {
  library(mxnet)
  data <- mx.symbol.Variable('data')
  # first conv
  conv1 <- mx.symbol.Convolution(data = data, kernel = c(5, 5), num_filter = 20)

  tanh1 <- mx.symbol.Activation(data = conv1, act_type = "tanh")
  pool1 <- mx.symbol.Pooling(data = tanh1, pool_type = "max", kernel = c(2, 2), stride = c(2, 2))

  # second conv
  conv2 <- mx.symbol.Convolution(data = pool1, kernel = c(5, 5), num_filter = 50)
  tanh2 <- mx.symbol.Activation(data = conv2, act_type = "tanh")
  pool2 <- mx.symbol.Pooling(data = tanh2, pool_type = "max", kernel = c(2, 2), stride = c(2, 2))
  # first fullc
  flatten <- mx.symbol.Flatten(data = pool2)
  fc1 <- mx.symbol.FullyConnected(data = flatten, num_hidden = 500)
  tanh3 <- mx.symbol.Activation(data = fc1, act_type = "tanh")
  # second fullc
  fc2 <- mx.symbol.FullyConnected(data = tanh3, num_hidden = num_classes)
  # loss
  lenet <- mx.symbol.SoftmaxOutput(data = fc2, name = 'softmax')
  return(lenet)
}
```

Configure custom network structure (MXNet syntax)

In [7]:
```
nclasses = h2o.nlevels(df[,response])
network <- get_symbol(nclasses)
cat(network$as.json(), file = "/tmp/symbol_lenet-R.json", sep = '')
```

Saving the custom network structure as a file

Train a Custom Network

```
model = h2o.deepwater(x=path, y=response, training_frame = df,
                      epochs=500, ## early stopping is on by default and might trigger before
                      network_definition_file="/tmp/symbol_lenet-R.json",  ## specify the model
                      image_shape=c(28,28),                                ## provide expected (or matchin
g) image size
                      channels=3)                                          ## 3 for color, 1 for monochrom
e
```

Point it to the custom network structure file

Conclusions

Project “Deep Water”

- H_2O + TF + MXNet + Caffe
  - A powerful combination of widely used open source machine learning libraries.
- All Goodies from H_2O
  - Inherits all H_2O properties in scalability, ease of use and deployment.
- Unified Interface
  - Allows users to build, stack and deploy deep learning models from different libraries efficiently.

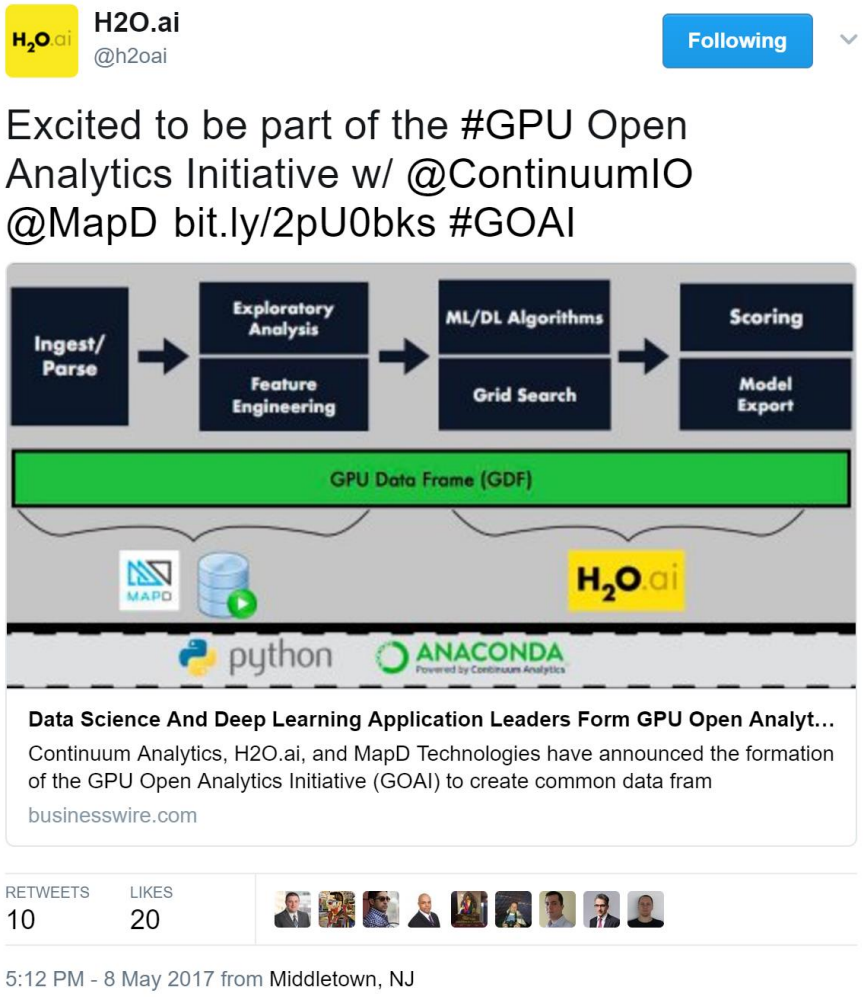
- Download and Try Deep Water
  - https://www.h2o.ai/download/
- 100% Open Source
  - The party will get bigger!

Latest H_2O Developments

H2O + xgboost, Automatic Machine Learning (AutoML), word2vec …

Other H_2O Developments

- H_2O + xgboost [Link]
- Stacked Ensembles [Link]
- Automatic Machine Learning [Link]
- Time Series [Link]
- High Availability Mode in Sparkling Water [Link]
- word2vec [Link]



H2O.ai

Thank you!

- Conference Organizers & Sponsors

- Code, Slides & Documents
  - bit.ly/h2o_meetups
  - docs.h2o.ai
- Contact
  - joe@h2o.ai
  - @matlabulous
  - github.com/woobe
- Please search/ask questions on **Stack Overflow**
  - Use the tag `**h2o**` (not H2 zero)