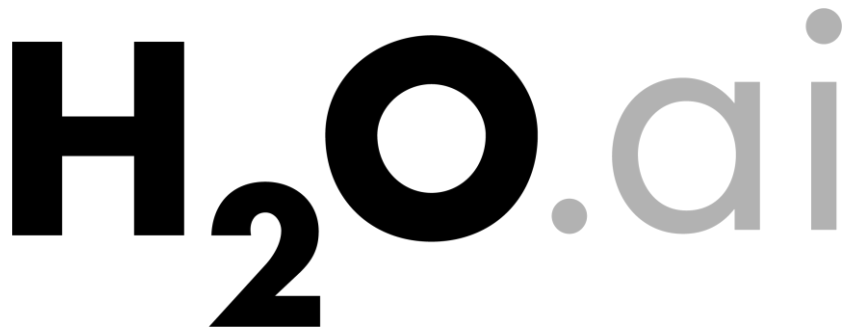
H_2O at BelgradeR Meetup

Introduction to ML with H_2O, Deep Water and New Developments



Jo-fai (Joe) Chow

Data Scientist

joe@h2o.ai

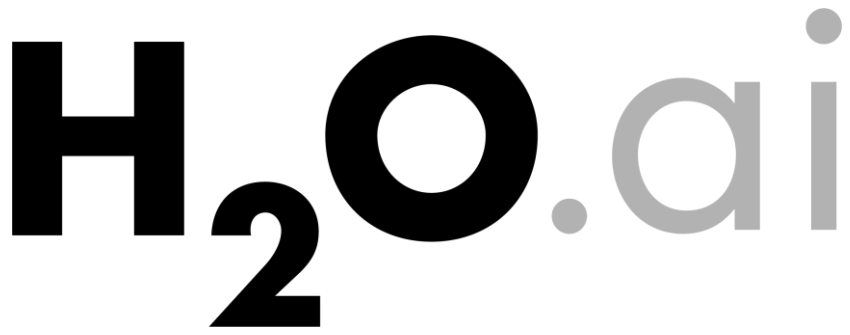
@matlabulous

BelgradeR at Seven Bridges

9th May, 2017

H_2O at BelgradeR Meetup

Introduction to ML with H_2O, Deep Water and New Developments



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
- Latest H_2O Developments
  - H_2O + xgboost
  - Automatic Machine Learning (AutoML)
  - NLP: word2vec in H_2O

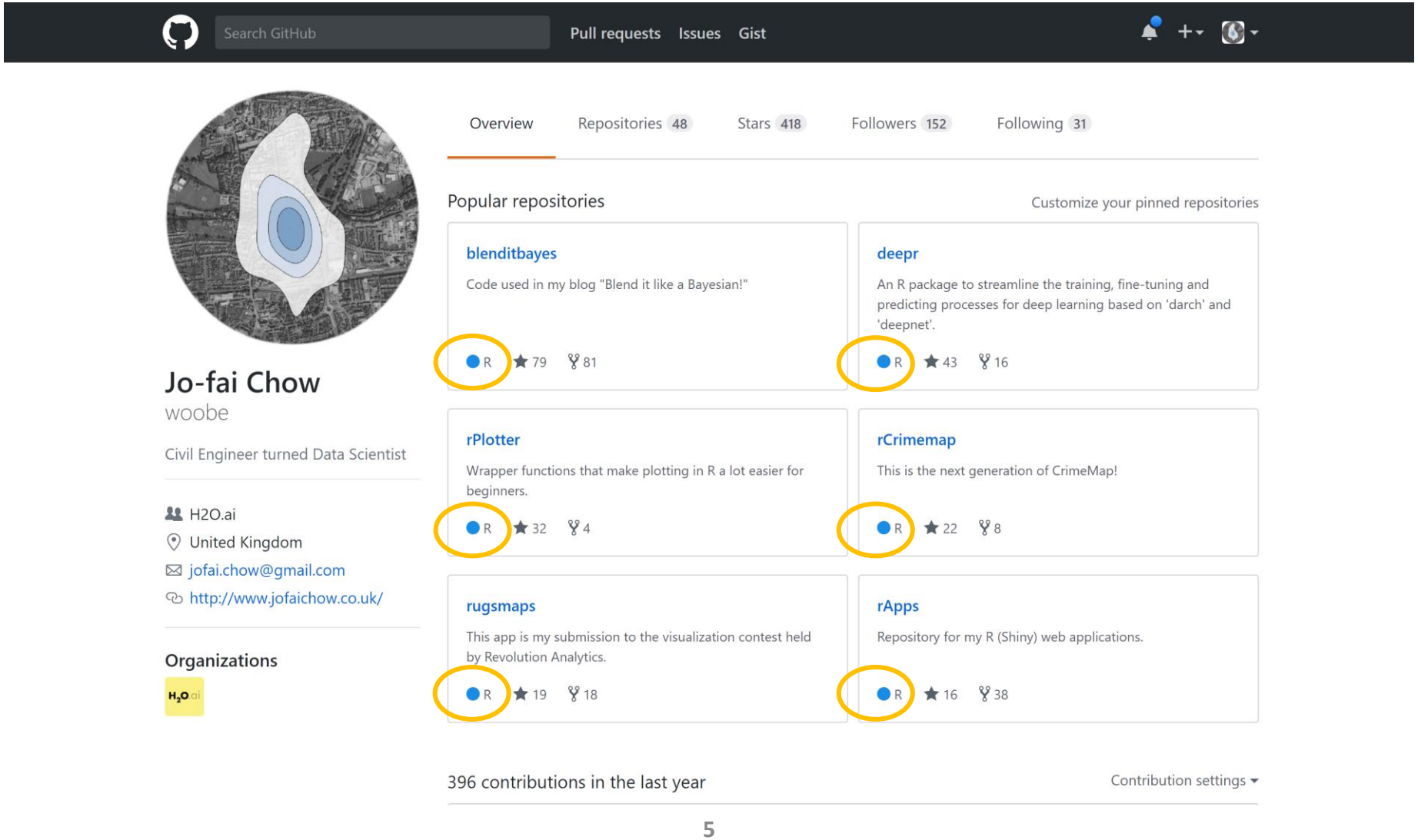
H_2O.ai

Seven Bridges

About Me

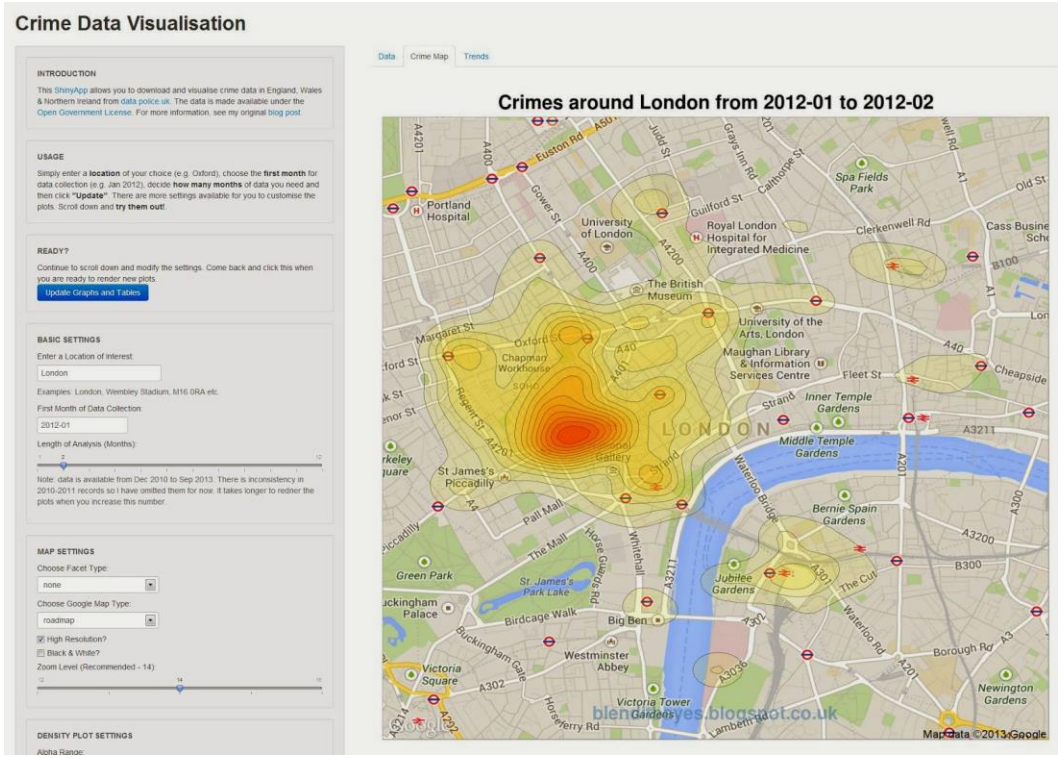
- Civil (Water) Engineer
  - 2010 – 2015
  - Consultant (UK)
    - Utilities
    - Asset Management
    - Constrained Optimization
  - EngD (Industrial PhD) (UK)
    - Infrastructure Design Optimization
    - Machine Learning + Water Engineering
    - Discovered H_2O in 2014
- Data Scientist
  - 2015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

About Me – I😍R

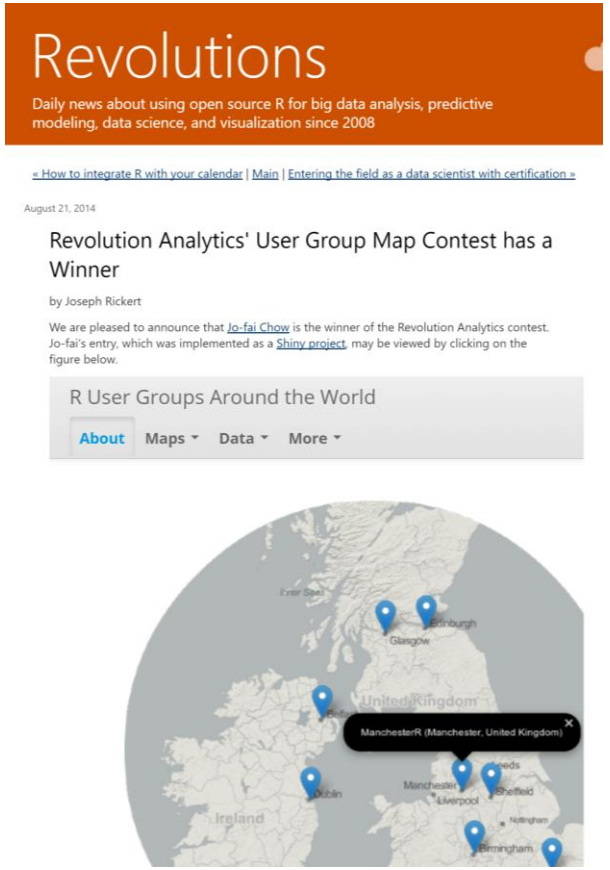


H2O.ai

About Me – I Love DataViz



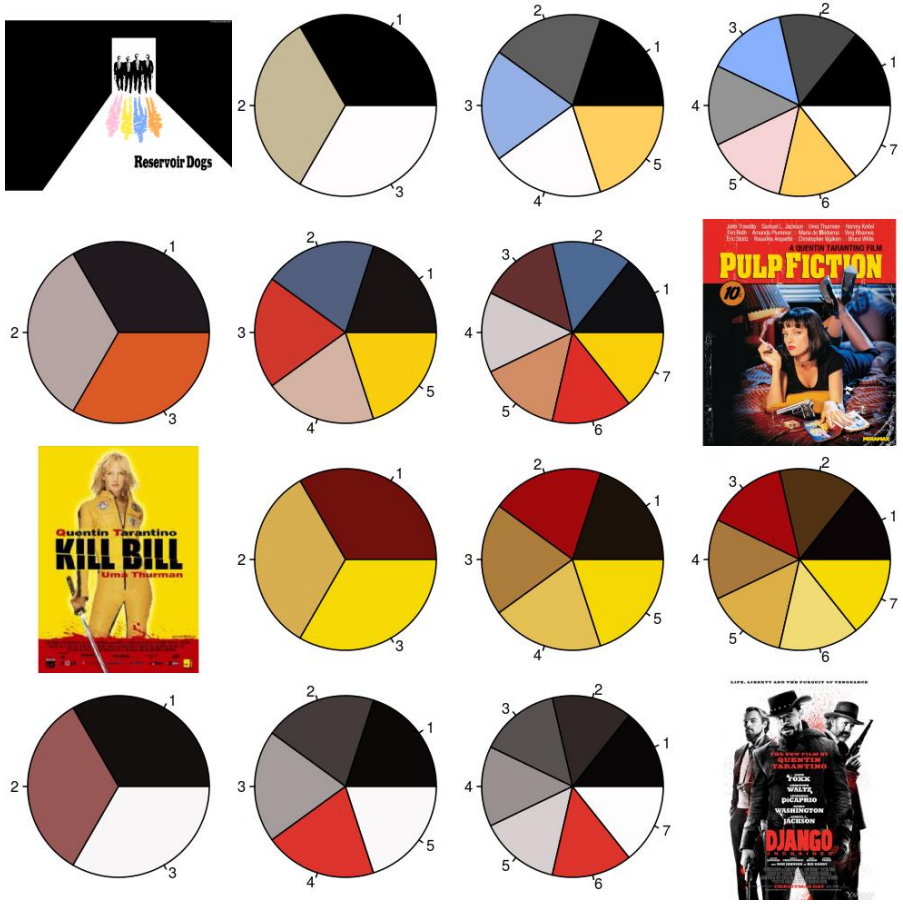
My First Data Viz & Shiny App Experience
[CrimeMap](#) (2013)

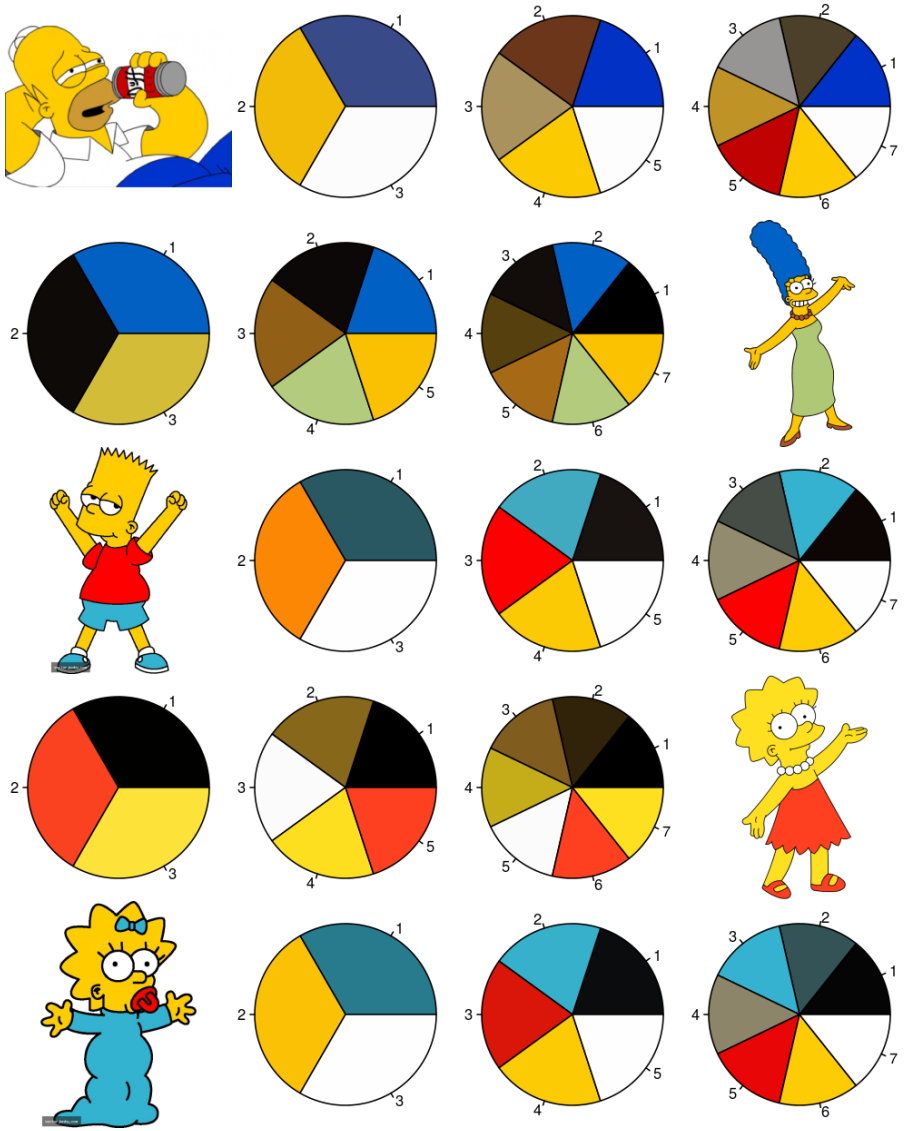


Revolution Analytics' Data Viz Contest
[RUGSMAPS](#) (2014)

H2O.ai

About Me – I love Colours

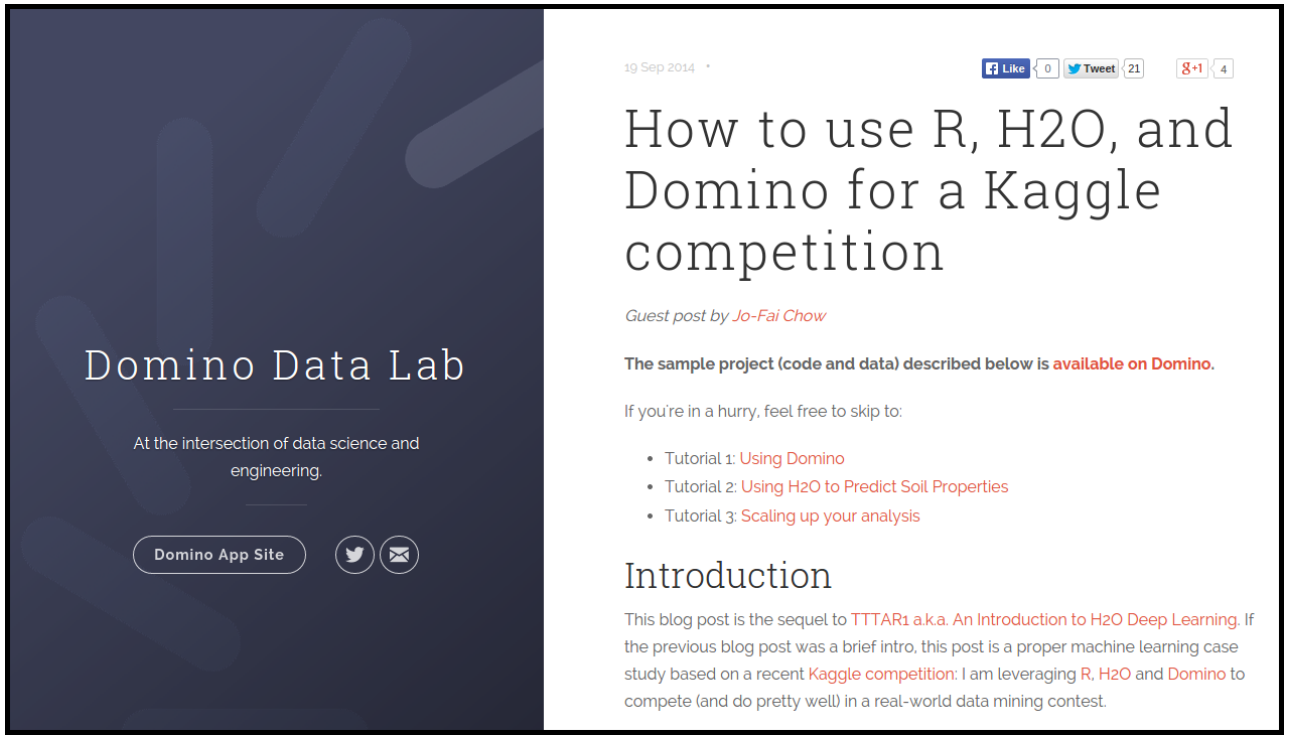




Developing R Packages for Fun
[rPlotter](#) (2014)

H2O.ai

About Me – From Kaggle to H_2O



R + H_2O + Domino for Kaggle

Guest Blog Post for Domino & H_2O (2014)

- The Long Story
  - bit.ly/joe_kaggle_story

H_2O.ai

My EngD Supervisors at Uni of Exeter, UK

- Prof Dragan Savić FREng

- Prof Zoran Kapelan


2012 – My First Trip to Belgrade
Urban Drainage Modelling Conference
Photo Credit: Jo-fai Chow (2012)

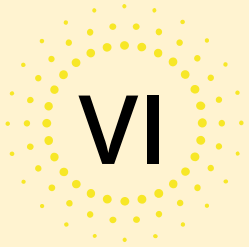
About H_2O.ai

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

H_2O.ai Offers AI Open Source Platform

Product Suite to Operationalize Data Science with Visual Intelligence

This Meetup

nework For Data Interpretation and
ul 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

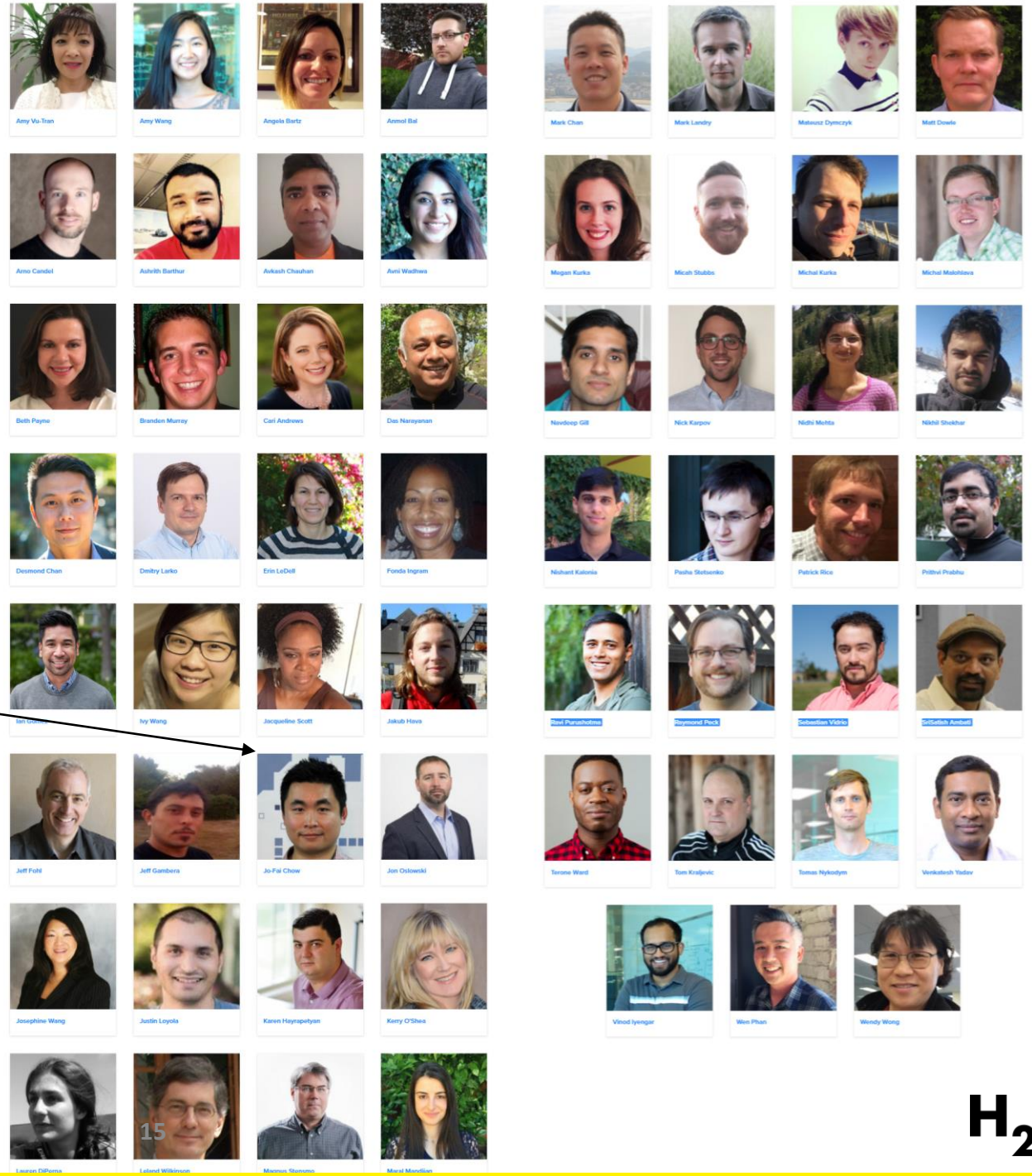
H2O Integration with Spark. Best Machine Learning on Spark.

Steam

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

Our Team

Joe



H2O.ai

Scientific Advisory Council

Dr. Trevor Hastie

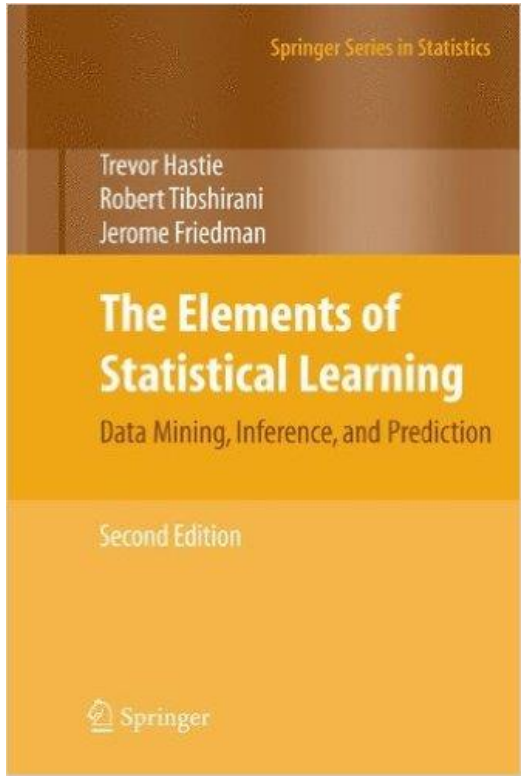
- John A. Overdeck Professor of Mathematics, Stanford University
- PhD in Statistics, Stanford University
- Co-author, *The Elements of Statistical Learning: Prediction, Inference and Data Mining*
- Co-author with John Chambers, *Statistical Models in S*
- Co-author, *Generalized Additive Models*

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

wenphan
@wenphan

Following

So much brain power in one place: @ArnoCandel and Stanford profs. Boyd, Tibs, and Hastie. Hacking algos at @h2oai HQ

Arno (CTO)

H2O.ai

Community

Search Inside and Read

Larger Cover

Practical Machine Learning with H2O

Powerful, Scalable Techniques for Deep Learning and AI

By Darren Cook

Publisher: O'Reilly Media
Final Release Date: December 2016
Pages: 300

0.0

Write a Review

Machine learning has finally come of age. With H2O software, you can perform machine learning and data analysis using a simple open source framework that's easy to use, has a wide range of OS and language support, and scales for big data. This hands-on guide teaches you how to use H20 with only minimal math and theory behind...

Full description

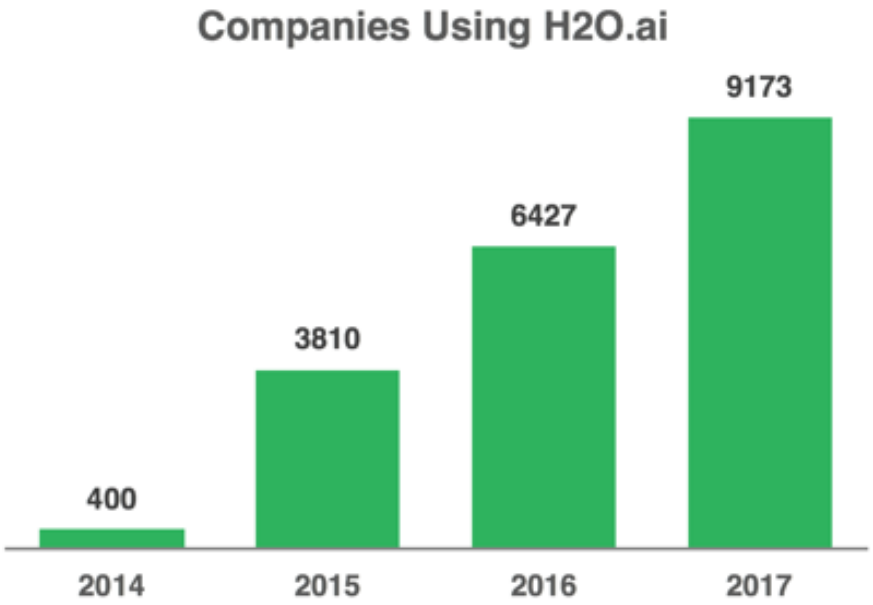
Table of Contents | Product Details | About the Author | Colophon

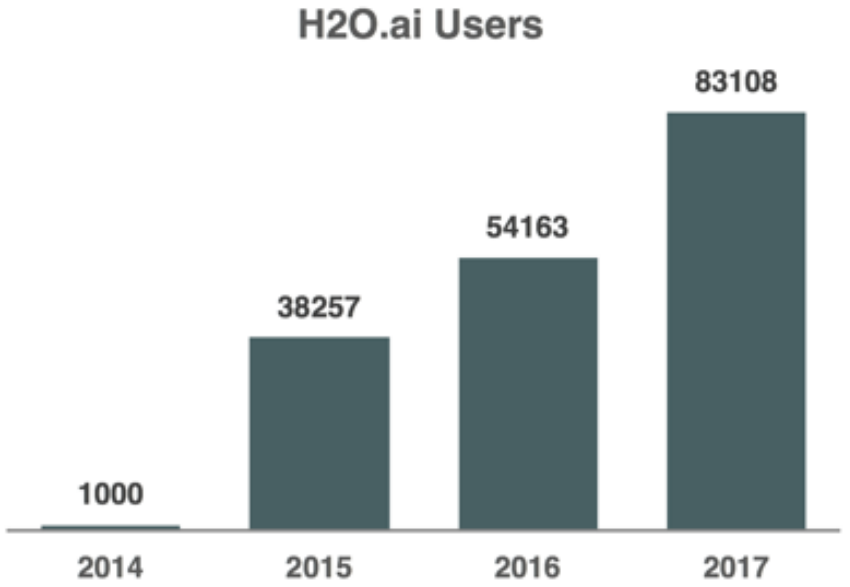
Darren Cook

Darren Cook has over 20 years of experience as a software developer, data analyst, and technical director, working on everything from financial trading systems to NLP, data visualization tools, and PR websites for some of the world's largest brands. He is skilled in a wide range of computer languages, including R, C++, PHP, JavaScript, and Python. He works at QQ Trend, a financial data analysis and data products company.

View Darren Cook's full profile page.

H_2O Community Growth





* Data from July of every year, except for 2017 when data from Feb 21st are used.

H2O.ai

#AroundTheWorldWithH2Oai

My first H_2O talk
March 2016

H2O.ai

A year ago I couldn't even imagine myself attending #StrataHadoop thx @h2oai for the oppor… ift.tt/1UjtrG7

First @h2oai #rstats #meetup in #Poznan! Many thanks to @mberesewicz @adolfoalvarez #PAZUR slideshare.net/JofaiChow/h2o- … Next stop: #London

Thanks #ViennaR #meetup for having us @h2oai see you next time with more 💦🍕 next stop #Amsterdam #PyData #AroundTheWorldWithH2Oai ✈️

Ciao #Barcelona Thanks @aleixrvr #RugBcn for having @h2oai we'll be back with #SparklingWater 💦 bit.ly/h2o_meetups next stop ✈️ #Madrid

Jo-fai (Joe) Chow
@matlabulous

My 2nd @h2oai talk at #skillsmatter #codenode bit.ly/joe_h2o_talk2 #twitter ift.tt/20nUQeu

#AroundTheWorldWithH2Oai @h2oai #paris bit.ly/h2o_meetups #museedulouvre #twitter bit.ly/2g6m9tb

Good evening #Cologne 🇩🇪 #AroundTheWorldWithH2Oai #CologneCathedral #Germany #twitter bit.ly/2nJTxJG

Helping #Refugees by teaching them basic #machinelearning skills @h2oai workshop @Restart_Network #Rotterdam 🇳🇱 #AroundTheWorldWithH2Oai

Thx @fishnets88 & everyone @GoDataDriven for hosting @pydataamsterdam @h2oai tutorial yesterday. Very cool office 😎 #AroundTheWorldWithH2Oai

"When one drinks @h2oai, one must not forget where it comes from." Thank you @Stanford for @h2oai & #useR2016 💦

@h2oai 's very first #meetup in #Warsaw. Thx @DominikBatorski & Wit Jakuczun for the connection & opportunities (bit.ly/h2o_warsaw_1)

Thx @DataScienceMi #datasciencemilan @h2oai bit.ly/h2o_milan_1 #AroundTheWorldWithH2Oai #t… ift.tt/2dTKWDh

@h2oai at #satRdays many thanks to @daroczig @SteffLocke & all volunteers slides & code at bit.ly/h2o_budapest_1

Thanks @datatons @quierodata @manuellamelas @lailaelqadi for organizing the first @h2oai #Madrid #meetup we will be back 👌 next stop #Vienna

Thanks everyone for coming to my @h2oai @BigData_LDN talk today, all our conf/meetup slides -> github.com/h2oai/h2o-meet… #fullhouse 🙏

H2O.ai

Users In Various Verticals Adore H_2O

H2O.ai

Figure 1. Magic Quadrant for Data Science Platforms

Source: Gartner (February 2017)

H2O.ai recognized for completeness of vision and ability to execute

We are thrilled to be named a Visionary among the 16 vendors included in Gartner's 2017 Magic Quadrant for Data Science Platforms. As a Visionary we believe we are positioned highest in Ability to Execute for companies of our size and scale.

Since 2011, our mission has been to democratize data science through open source AI and deep learning. Today, H2O.ai is focused on bringing AI to enterprises with a growing community of more than 8,500 organizations that depend on H2O for mission critical applications. H2O.ai was recently named CB Insights AI 100 and is used by 107 of the Fortune 500 companies.

Disclaimer: This graphic was published by Gartner, Inc. as part of a larger research document and should be evaluated in the context of the entire document. The Gartner document is available upon request from H2O.ai.

http://www.h2o.ai/gartner-magic-quadrant/

H2O.ai

Figure 1. Magic Quadrant for Data Science Platforms

Source: Gartner (February 2017)

Platforms with H_2O Integrations

http://www.h2o.ai/gartner-magic-quadrant/

H_2O.ai

H2O In Action

Check out our website h2o.ai

ADP

Various data leaders discuss the transformative impact of H2O AI for ADP.

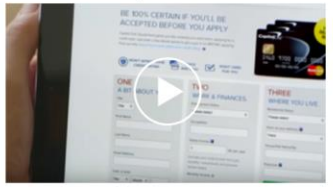
Insurance

What data products mean and why H2O keeps this industry leader relevant.

Progressive

See how Progressive uses H2O predictive analytics for User-based Insurance (UBI).

Capital One

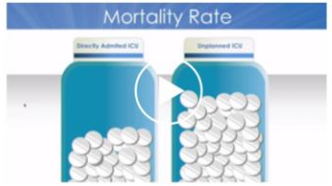


Capital One uses H2O machine learning for various use cases.

MarketShare

H2O predictive analytics helps boost the impact and results of digital marketing.

Kaiser



Kaiser uses H2O machine learning to save lives.

Zurich Insurance

Zurich turned to H2O as a strategic differentiator for commercial insurance.

Comcast

Comcast uses H2O to improve customer experience.

McKesson

McKesson discusses the adoption of artificial intelligence in healthcare.

Macy's

Macy's uses H2O for personalized site recommendations.

Transamerica

Transamerica turns to H2O to develop an insurance recommendation platform.

Paypal

Paypal turned to H2O Deep Learning for fraud detection and customer churn.

eBay

eBay chose H2O for open source machine learning.

Cisco

Cisco uses H2O to build a scalable model factory to improve sales and marketing.

Nielsen Catalina Solutions

H2O helps the country's largest TV behavior analytics company optimize ad performance.

H2O.ai

Why H_2O?

Szilard Pafka's ML Benchmark

Szilard Pafka
szilard

Follow

Block or report user

Santa Monica, California
https://www.linkedin.com/in/szi...

Organizations

Overview | Repositories 39 | Stars 27

Pinned repositories

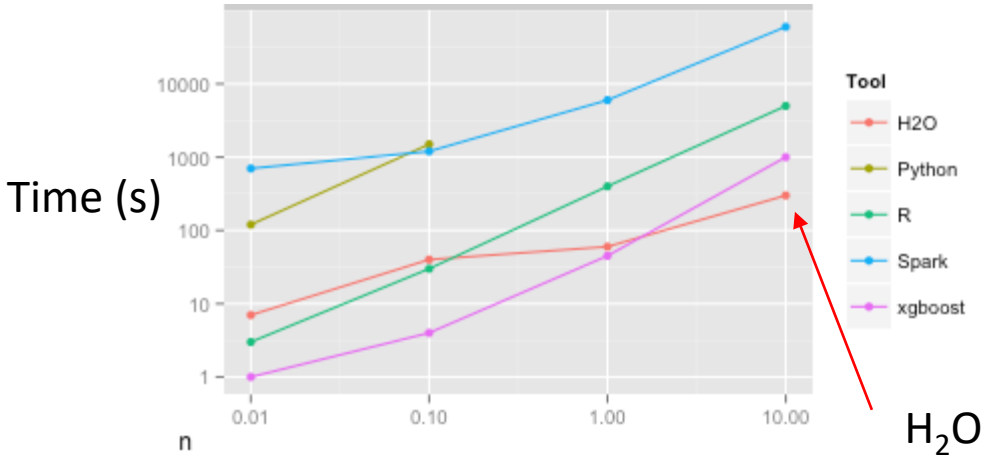
benchm-ml
A minimal benchmark for scalability, speed and accuracy of commonly used open source implementations (R packages, Python scikit-learn, H2O, xgboost, Spark MLlib etc.) of the top machine learning al...
R ★ 1.2k 198

teach-data-science-msc-analytics-ceu
Materials for a short introductory/intermediate Data Science course taught in the MSc in Business Analytics program at the Central European University
HTML ★ 21 11

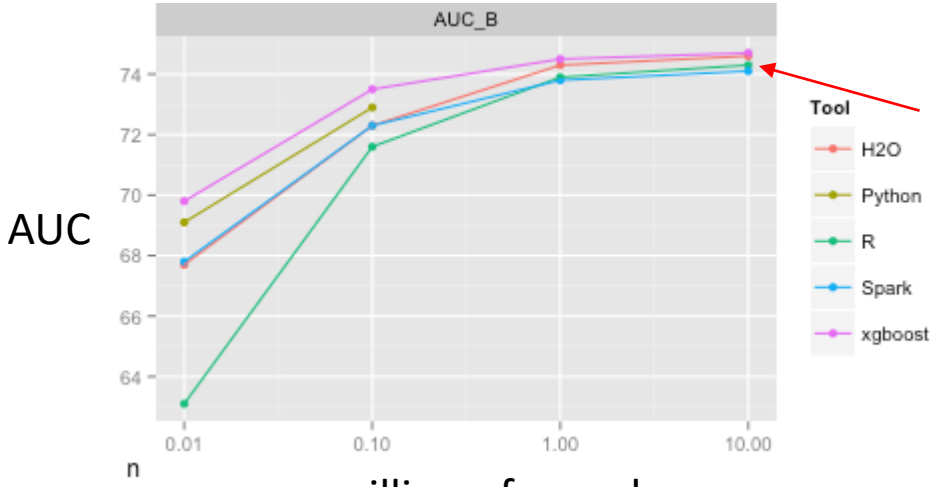
dataset-sizes-kdnuggets
Size of datasets used for analytics based on 10 years of surveys by KDnuggets.
HTML ★ 12 2

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

Gradient Boosting Machine Benchmark



H_2O is fastest at 10M samples

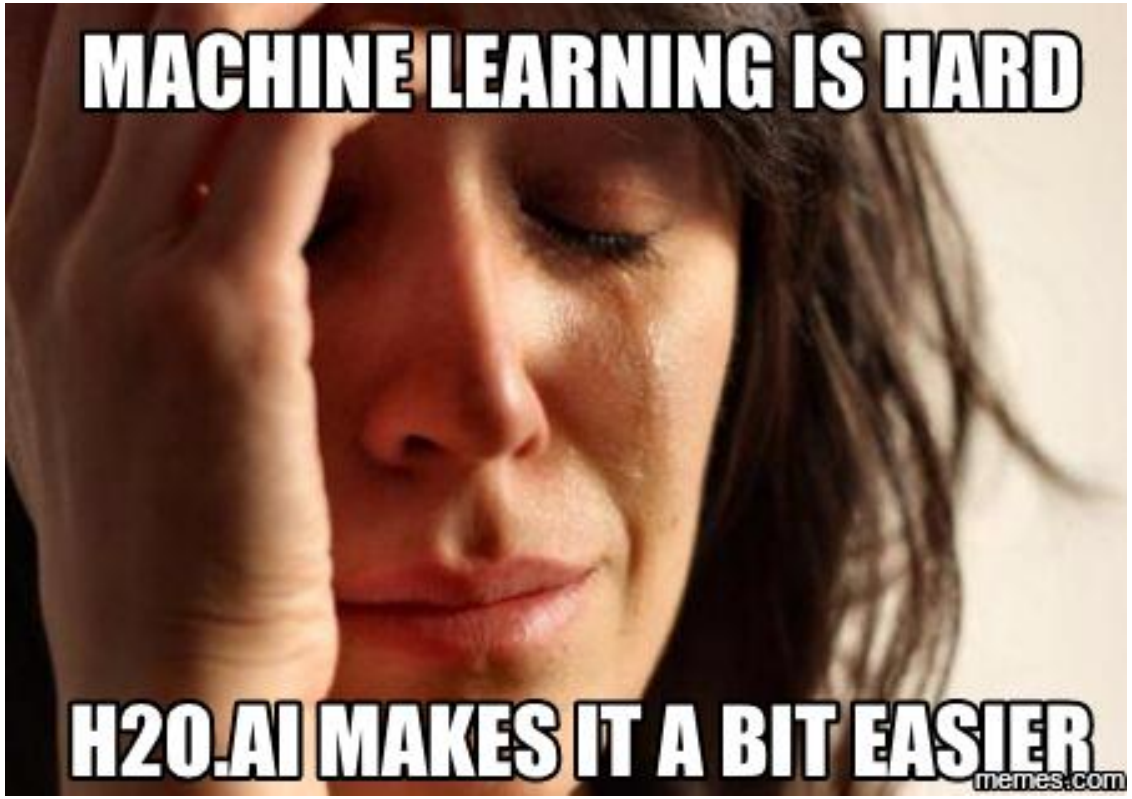


H_2O is as accurate as others at 10M samples

n = million of samples

H_2O.ai

Szilard Pafka's Comment



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

H2O.ai

H_2O for Kaggle Competitions

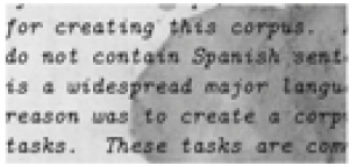
CIFAR-10 Competition Winners: Interviews with Dr. Ben Graham, Phil Culliton, & Zygmunt Zając

Triskelion | 01.02.2015

READ MORE

"I did really like H2O's deep learning implementation in R, though - the interface was great, the back end extremely easy to understand, and it was scalable and flexible. Definitely a tool I'll be going back to."

Kaggle challenge 2nd place winner Colin Priest



Completed • Knowledge • 161 teams

Denoising Dirty Documents

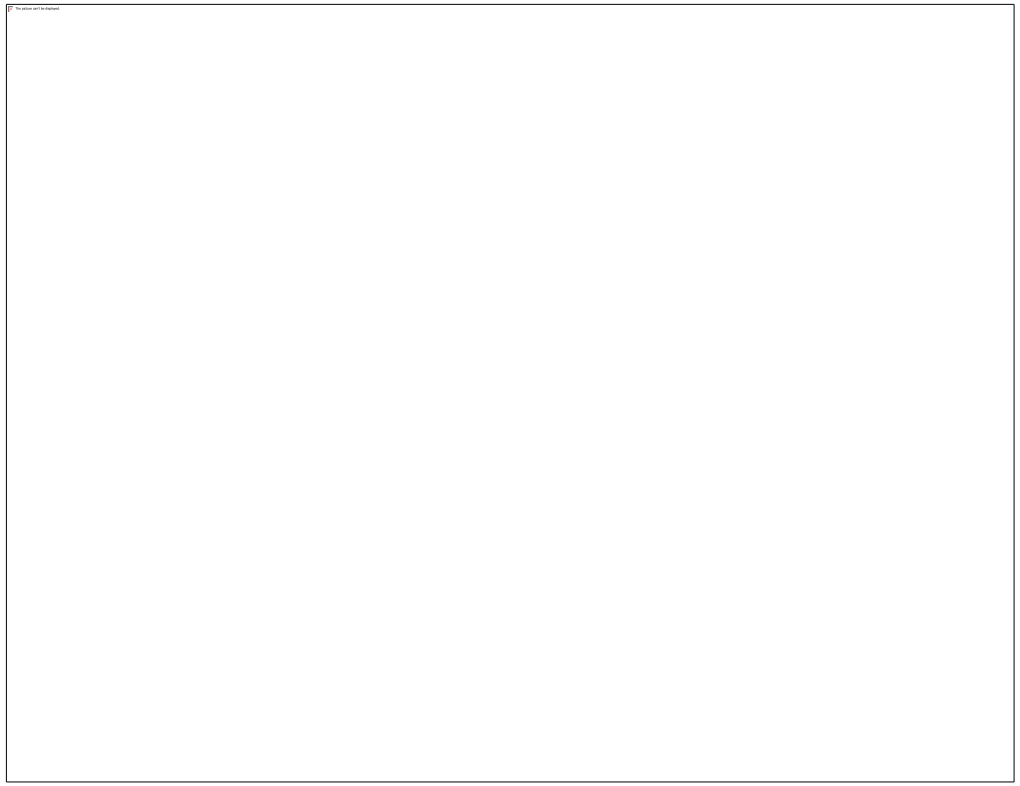
Mon 1 Jun 2015 – Mon 5 Oct 2015 (3 months ago)

READ MORE

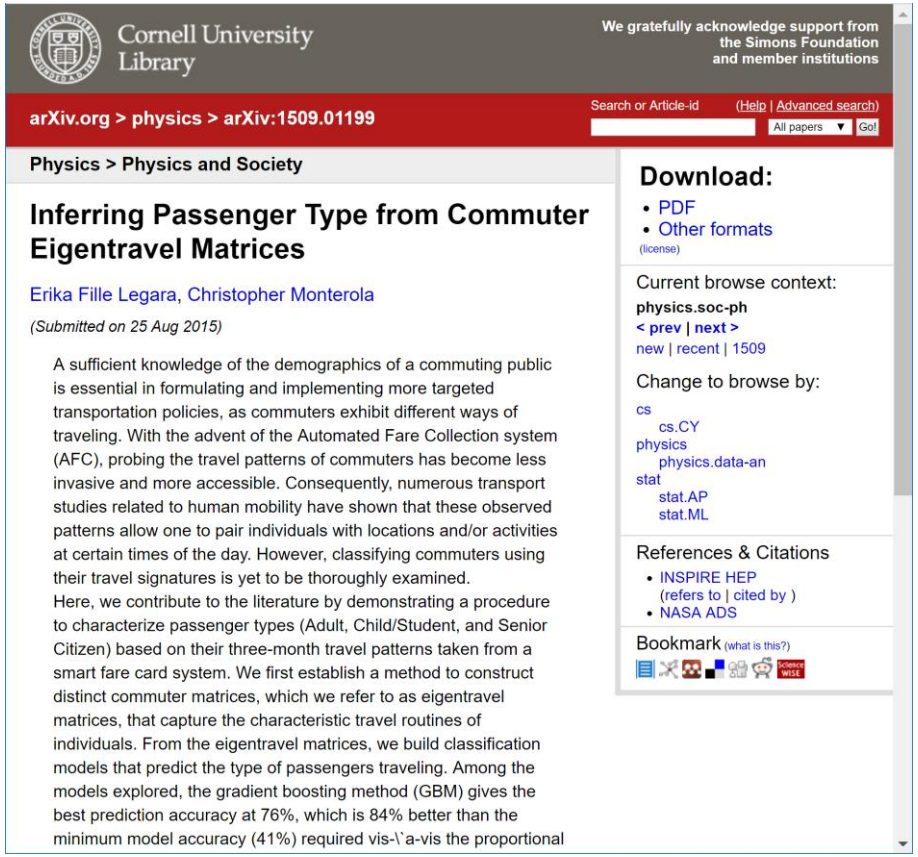
"For my final competition submission I used an ensemble of models, including 3 deep learning models built with R and h2o."

H_2O.ai

H_2O for Academic Research



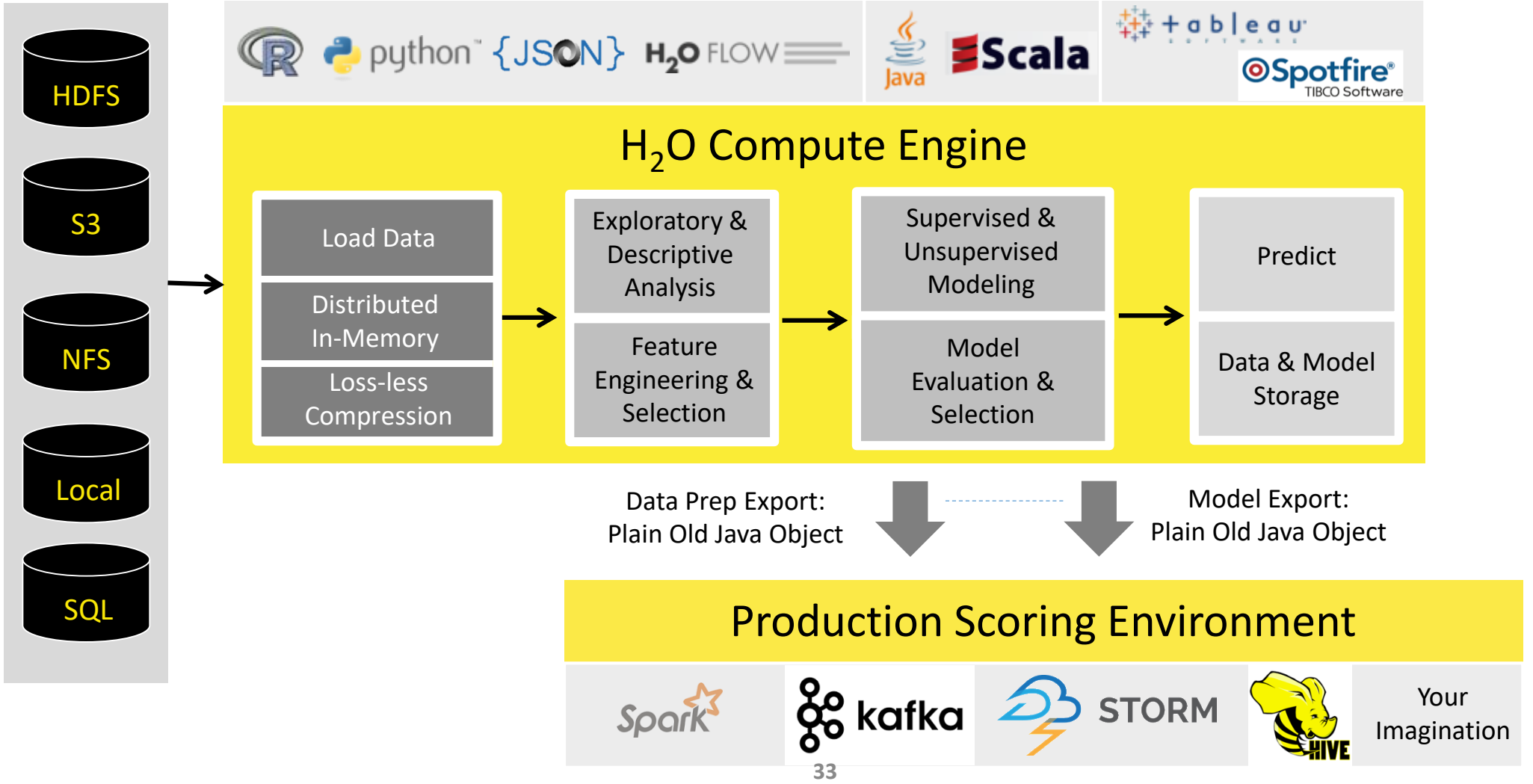
http://www.sciencedirect.com/science/article/pii/S0377221716308657



https://arxiv.org/abs/1509.01199

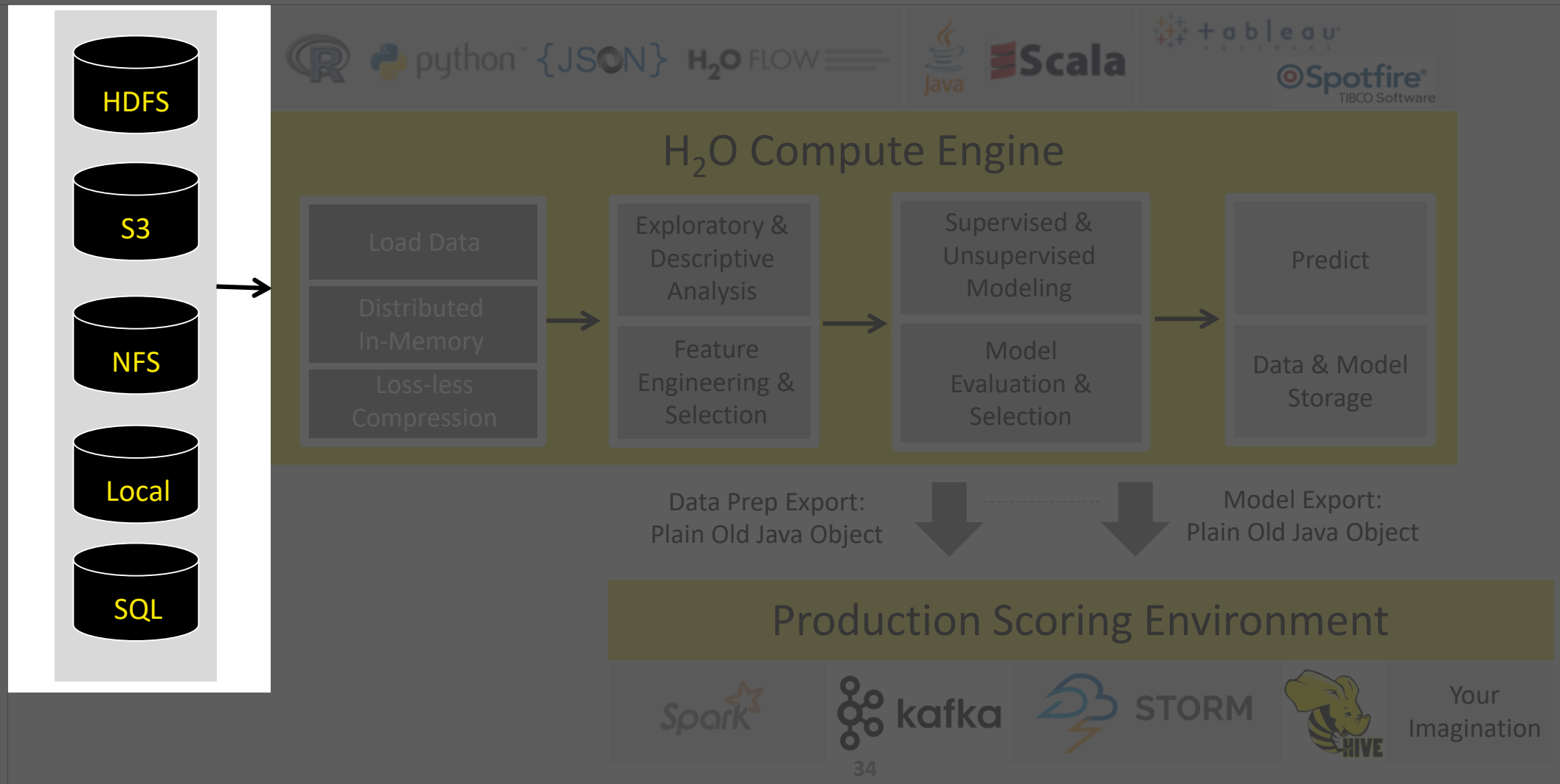
H_2O Machine Learning Platform

High Level Architecture



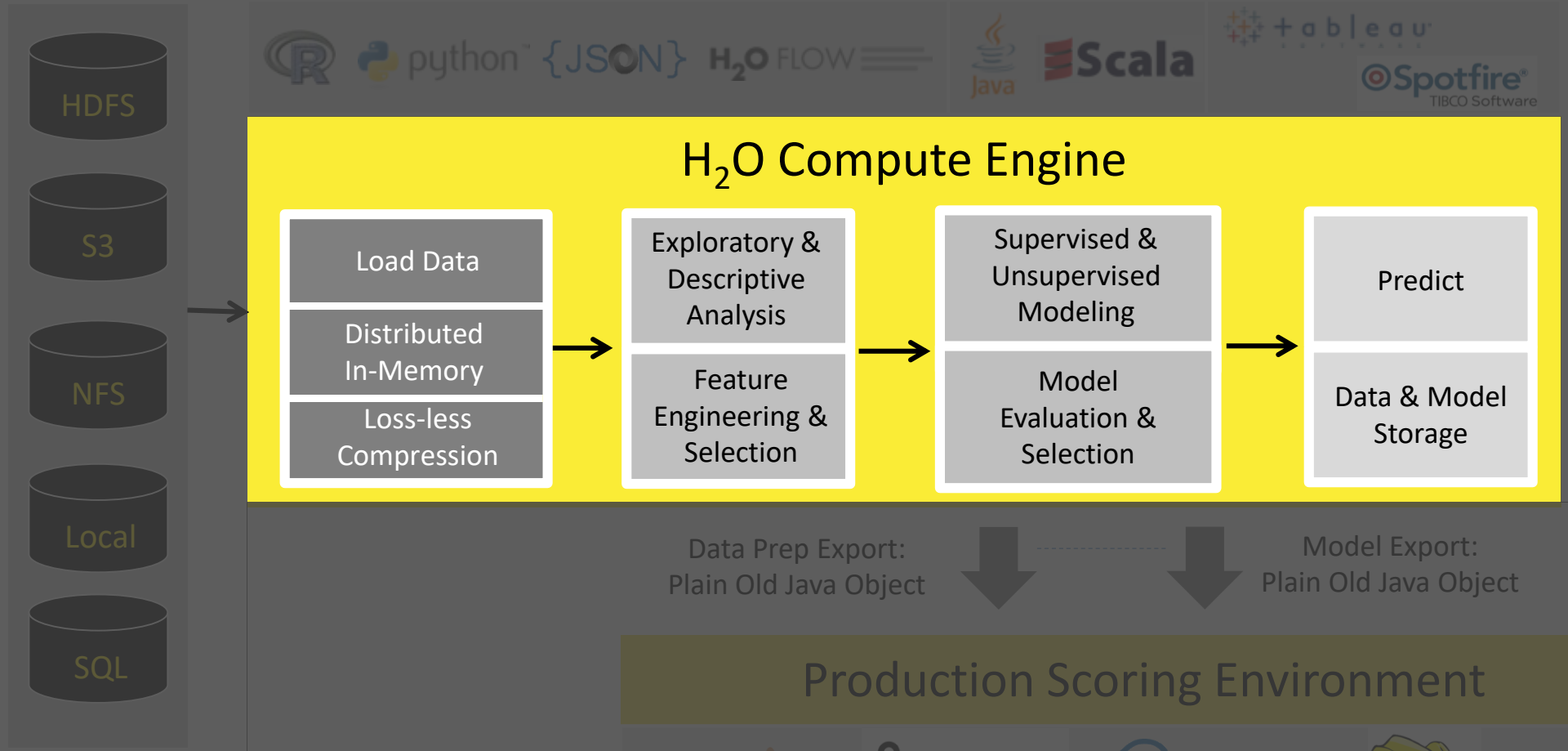
High Level Architecture

Import Data from Multiple Sources



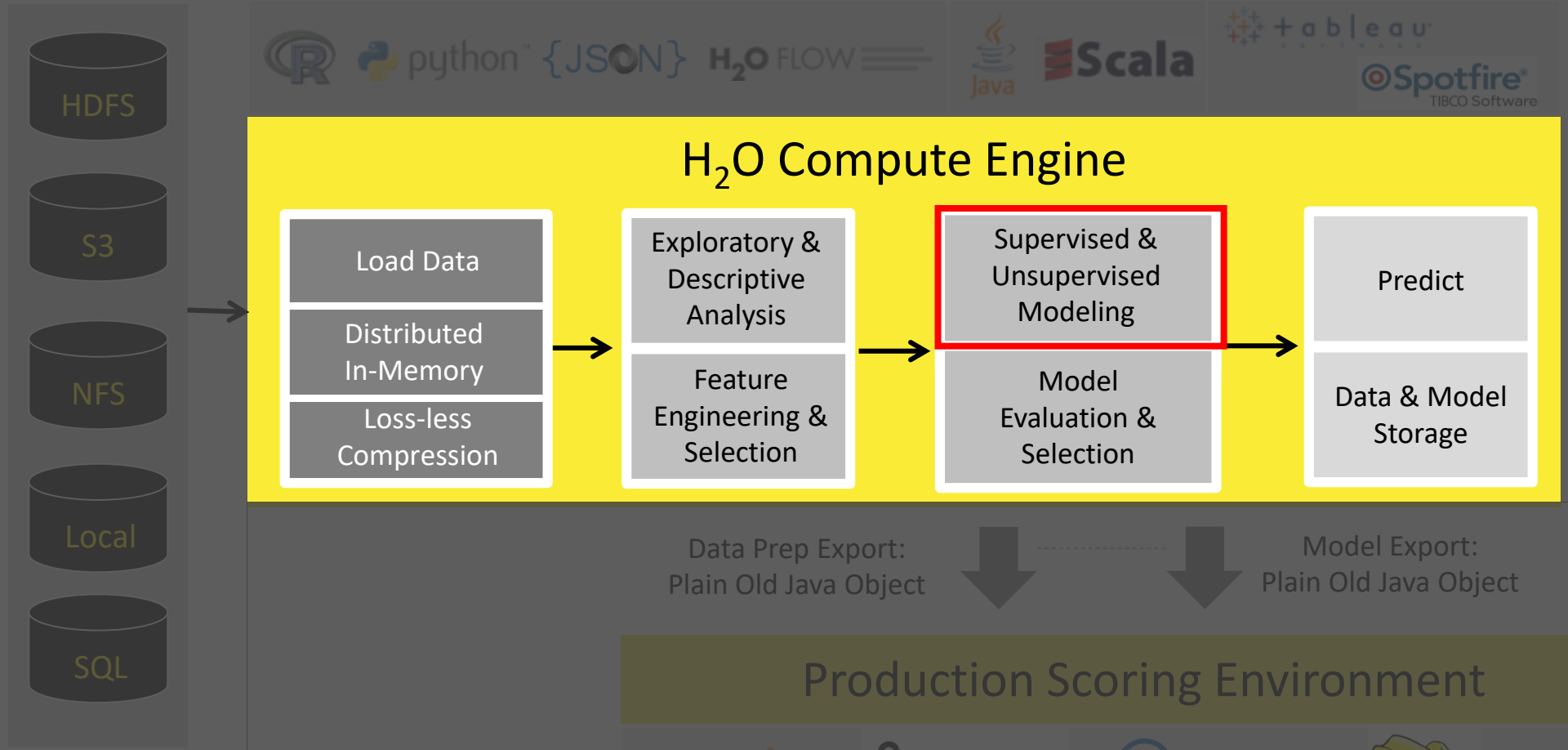
High Level Architecture

Fast, Scalable & Distributed Compute Engine Written in Java



High Level Architecture

Fast, Scalable & Distributed Compute Engine Written in Java



Algorithms Overview

Supervised Learning

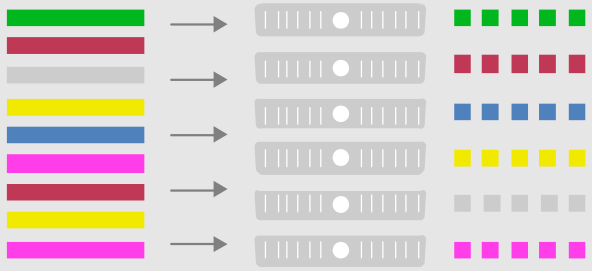
Statistical Analysis	• **Generalized Linear Models**: Binomial, Gaussian, Gamma, Poisson and Tweedie • **Naïve Bayes**
Ensembles	• **Distributed Random Forest**: Classification or regression models • **Gradient Boosting Machine**: Produces an ensemble of decision trees with increasing refined approximations
Deep Neural Networks	• **Deep learning**: Create multi-layer feed forward neural networks starting with an input layer followed by multiple layers of nonlinear transformations

Unsupervised Learning

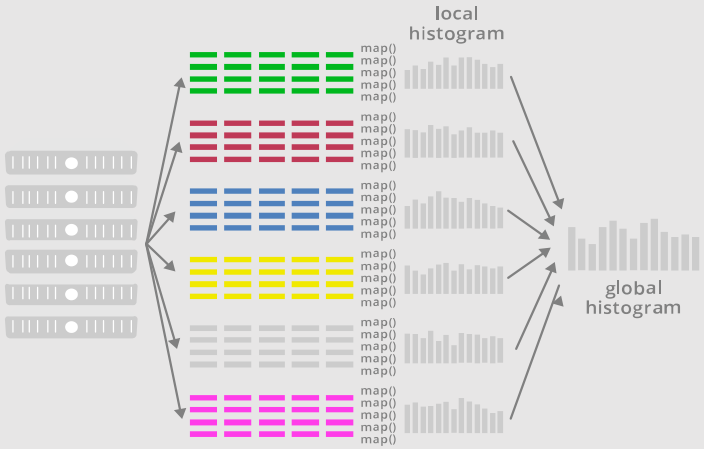
Clustering	• **K-means**: Partitions observations into k clusters/groups of the same spatial size. Automatically detect optimal k
Dimensionality Reduction	• **Principal Component Analysis**: Linearly transforms correlated variables to independent components • **Generalized Low Rank Models:** extend the idea of PCA to handle arbitrary data consisting of numerical, Boolean, categorical, and missing data
Anomaly Detection	• **Autoencoders**: Find outliers using a nonlinear dimensionality reduction using deep learning

Distributed Algorithms

Foundation for Distributed Algorithms



Parallel Parse into **Distributed Rows**



Fine Grain Map Reduce Illustration: Scalable Distributed Histogram Calculation for GBM

Advantageous Foundation

- Foundation for In-Memory Distributed Algorithm Calculation - **Distributed Data Frames** and **columnar compression**
- All algorithms are distributed in H_2O: GBM, GLM, DRF, Deep Learning and more. Fine-grained map-reduce iterations.
- **Only enterprise-grade, open-source distributed algorithms in the market**

User Benefits

- "Out-of-box" functionalities for all algorithms **(NO MORE SCRIPTING)** and uniform interface across all languages: R, Python, Java
- **Designed for all sizes of data sets, especially large data**
- **Highly optimized Java code for model exports**
- **In-house expertise for all algorithms**

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%

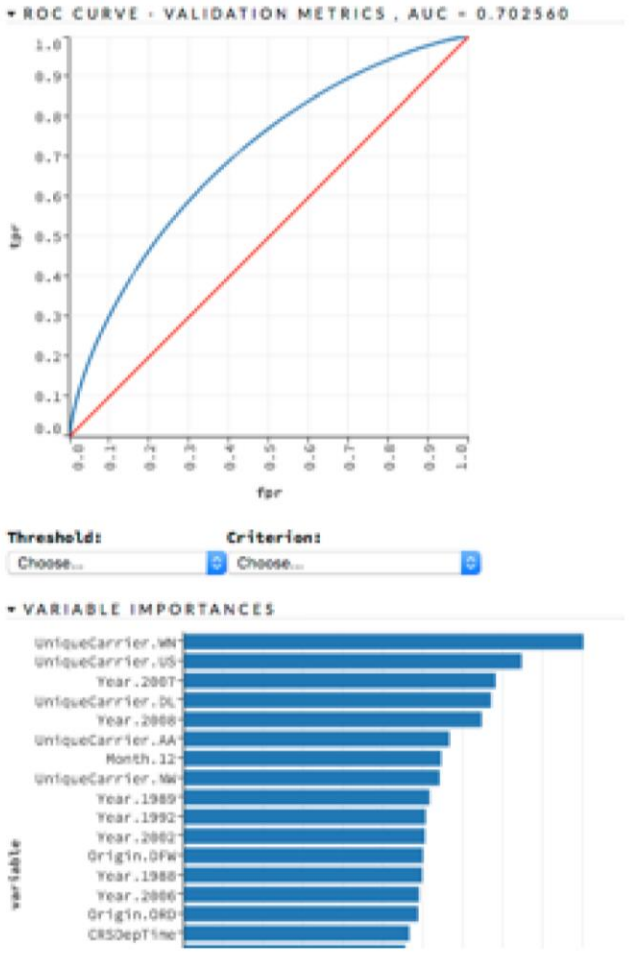
Iterations: 12. Epochs: 0.628821. Speed: 2,243,735 samples/sec. Estimated time left: 21.849 sec

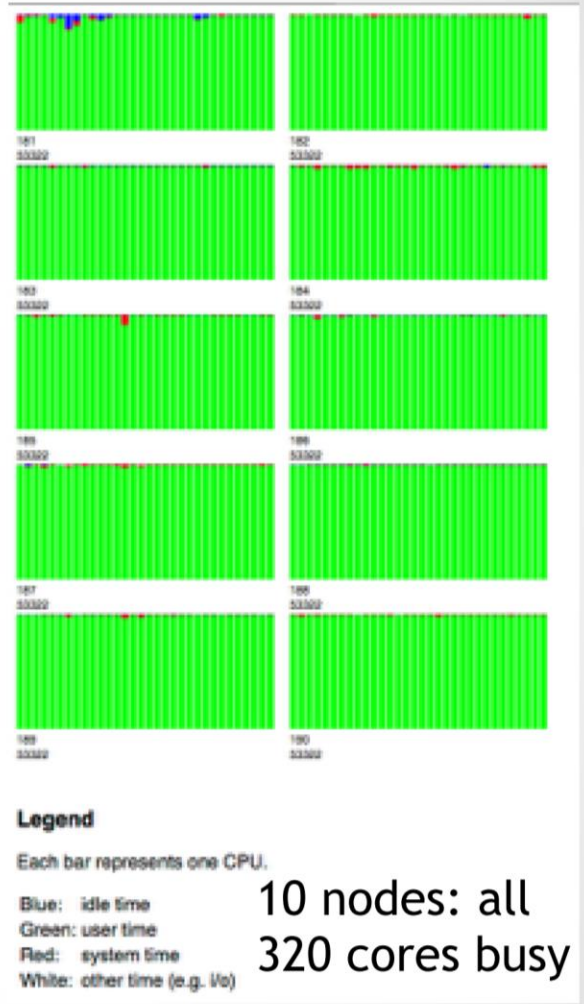
Actions: View | Cancel Job

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





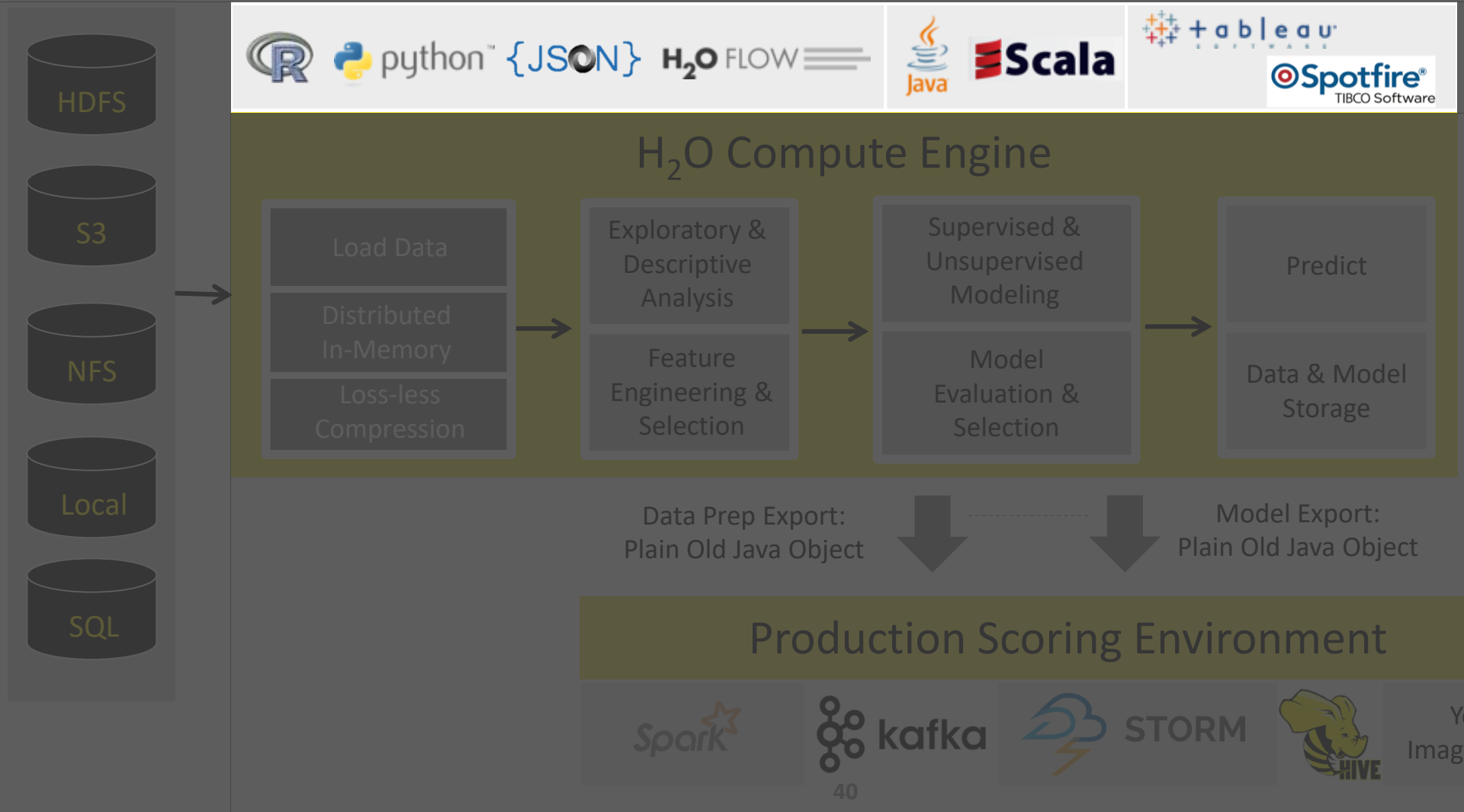
10 nodes: all
320 cores busy

real-time, interactive
model inspection in Flow

H_2O.ai **Deep Learning Model**

High Level Architecture

Multiple Interfaces



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)
```

H_2O + Python

Gradient Boosting Machines

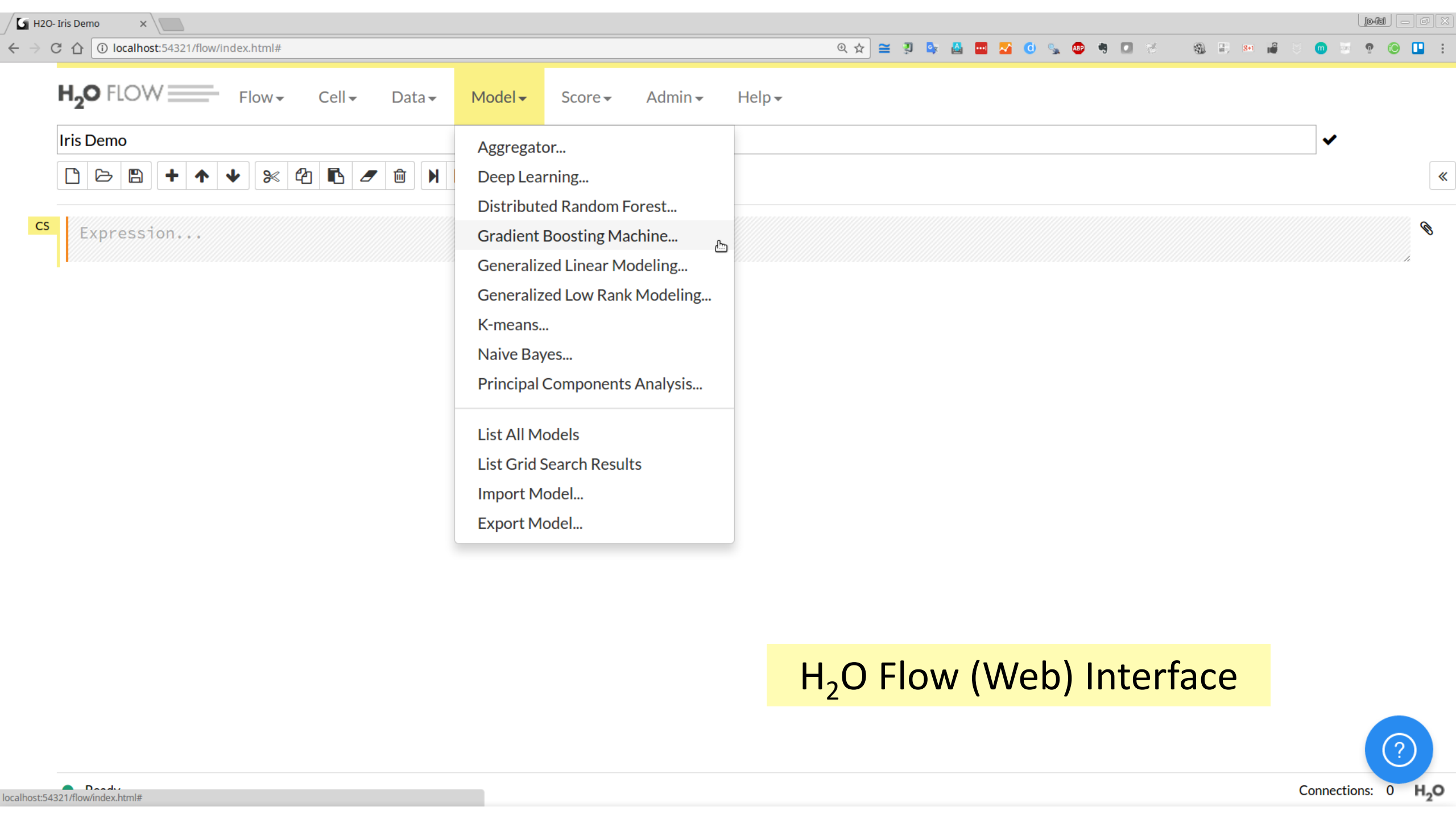
```
# Build a Gradient Boosting Machines (GBM) model with default settings

# 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)
```

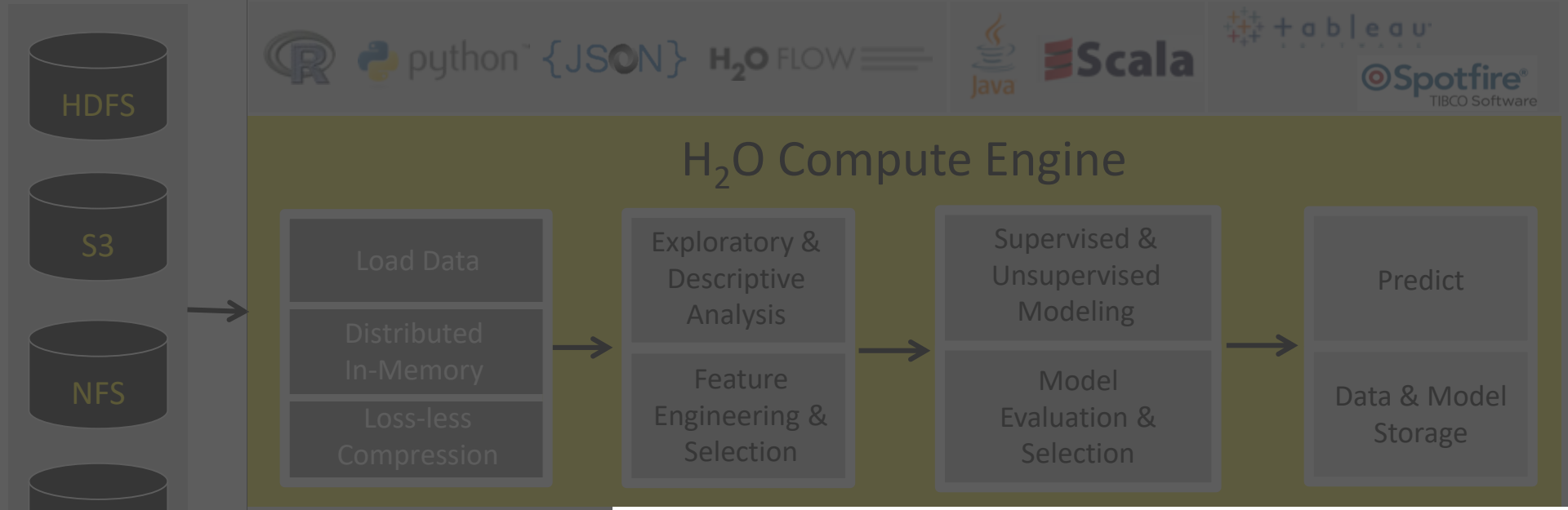
```
gbm Model Build progress: |████████████████████████████████████████████████| 100%
```



H_2O Flow (Web) Interface

High Level Architecture

Export Standalone Models for Production



Data Prep Export:
Plain Old Java Object

Model Export:
Plain Old Java Object

Production Scoring Environment

Spark | kafka | STORM | HIVE | Your Imagination

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

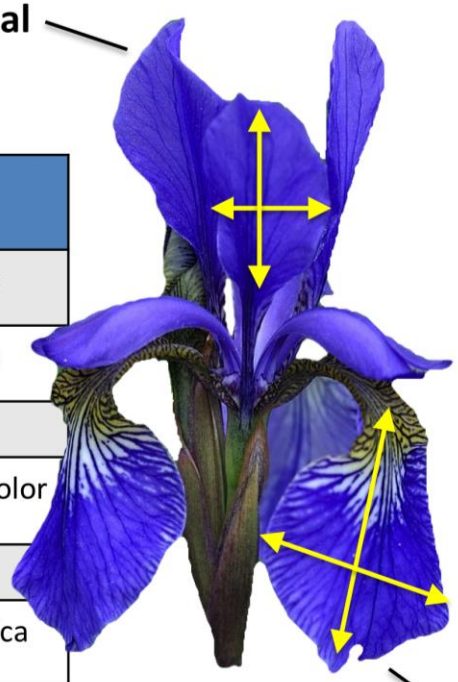
Demo Time

H_2O + R, Python and Flow (Web Interface)

Simple Demo – Iris

Samples
(instances, observations)

	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
			...		
50	6.4	3.5	4.5	1.2	Versicolor
			...		
150	5.9	3.0	5.0	1.8	Virginica



Class labels
(targets)

Features
(attributes, measurements, dimensions)

H_2O + R

```
# Start and connect to a local H2O cluster
library(h2o)
h2o.init(nthreads = -1)

# Import data from a R data frame
data(iris)
d_iris <- as.h2o(iris)

# Define Targets and Features
target <- "Species"
features <- setdiff(colnames(d_iris), c("Species"))
```

Package ‘h2o’ from CRAN or H_2O’s website

Start a local H_2O (Java Virtual Machine) cluster

Simple ‘iris’ example

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)
```

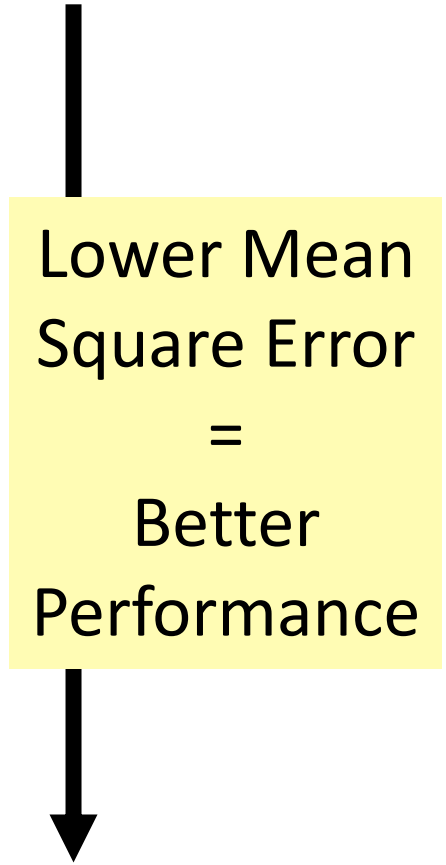
H_2O Tutorials

- Introduction to Machine Learning with H_2O
  - Basic Extract, Transform and Load (ETL)
  - Supervised Learning
  - Parameters Tuning
  - Model Stacking
- GitHub Repository
  - bit.ly/joe_h2o_tutorials
  - Both R & Python Code Examples Included

H2O.ai

Improving Model Performance (Step-by-Step)

Model Settings	MSE (CV)	MSE (Test)
GBM with default settings	N/A	0.4551
GBM with manual settings	N/A	0.4433
Manual settings + cross-validation	0.4502	0.4433
Manual + CV + early stopping	0.4429	0.4287
CV + early stopping + full grid search	0.4378	0.4196
CV + early stopping + random grid search	0.4227	0.4047
Stacking models from random grid search	N/A	0.3969



For More Details https://github.com/woobe/h2o_tutorials/tree/master/introduction_to_machine_learning

H2O.ai

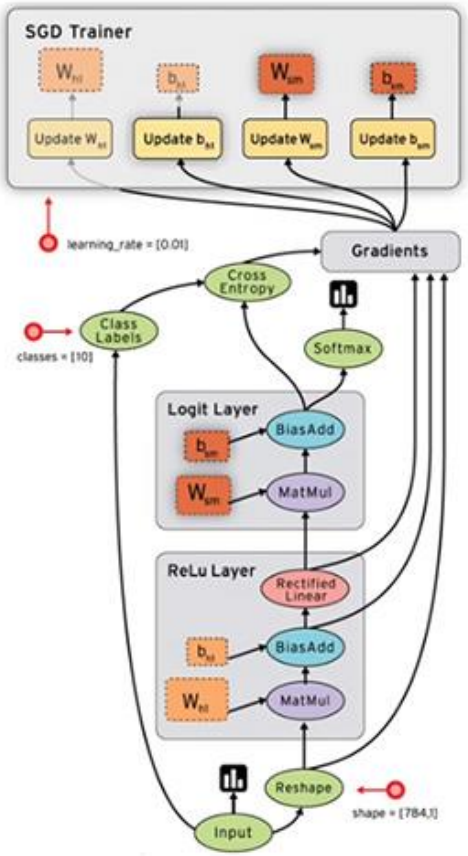
End of First Talk

Let’s have a break ☺


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 …



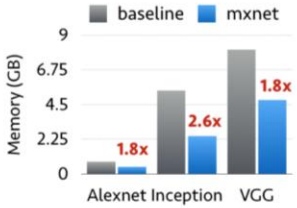
https://github.com/tensorflow/tensorflow

dmlc mxnet for Deep Learning

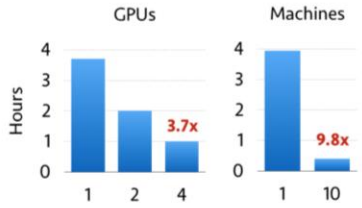
build passing | docs latest | license Apache 2.0

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%

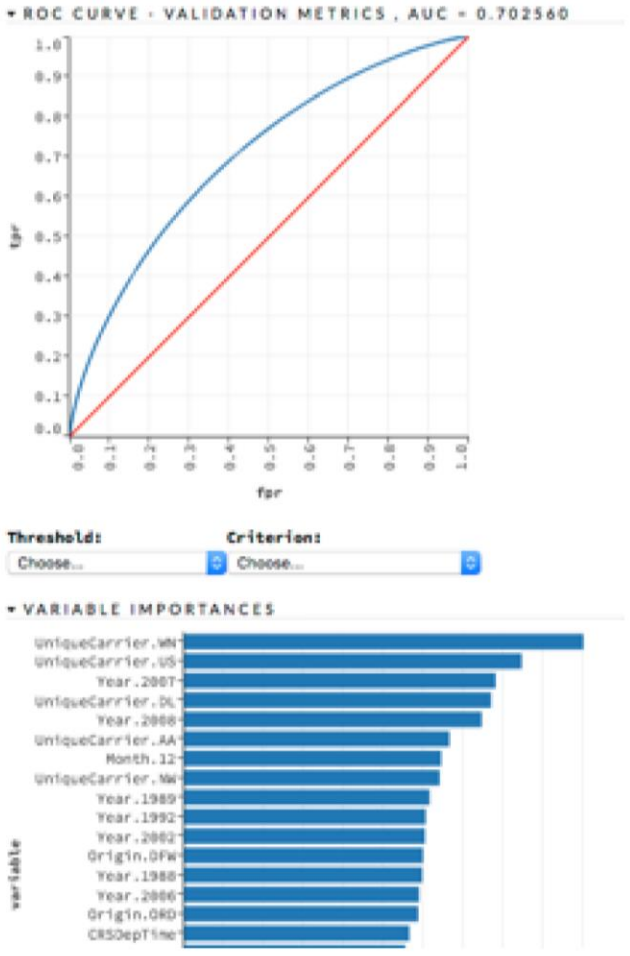
Iterations: 12. Epochs: 0.628821. Speed: 2,243,735 samples/sec. Estimated time left: 21.849 sec

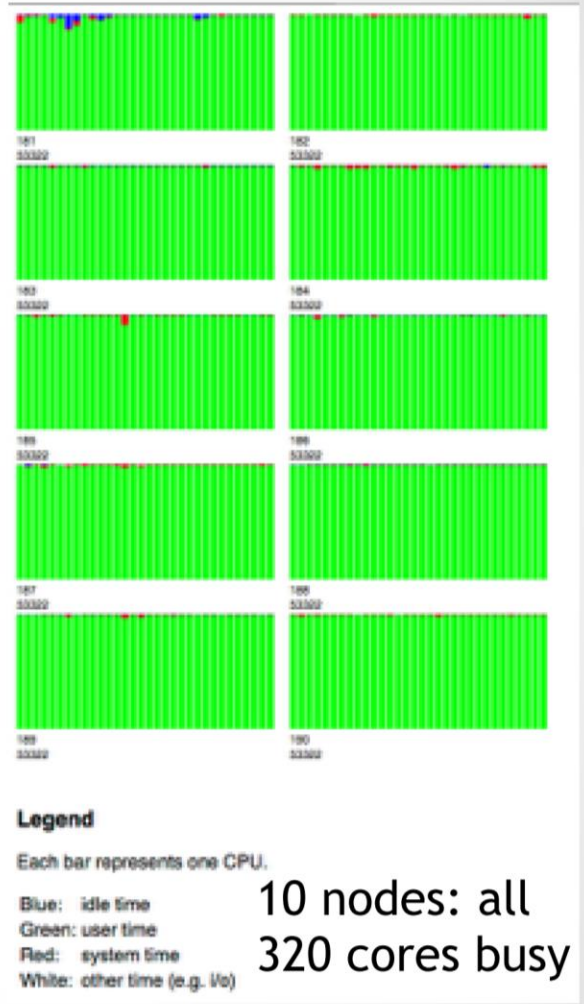
Actions: View | Cancel Job

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**

Both TensorFlow and H_2O are 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%
carch (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%

H_2O.ai

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

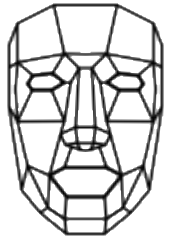
One Interface - GPU Enabled - Significant Performance Gains

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



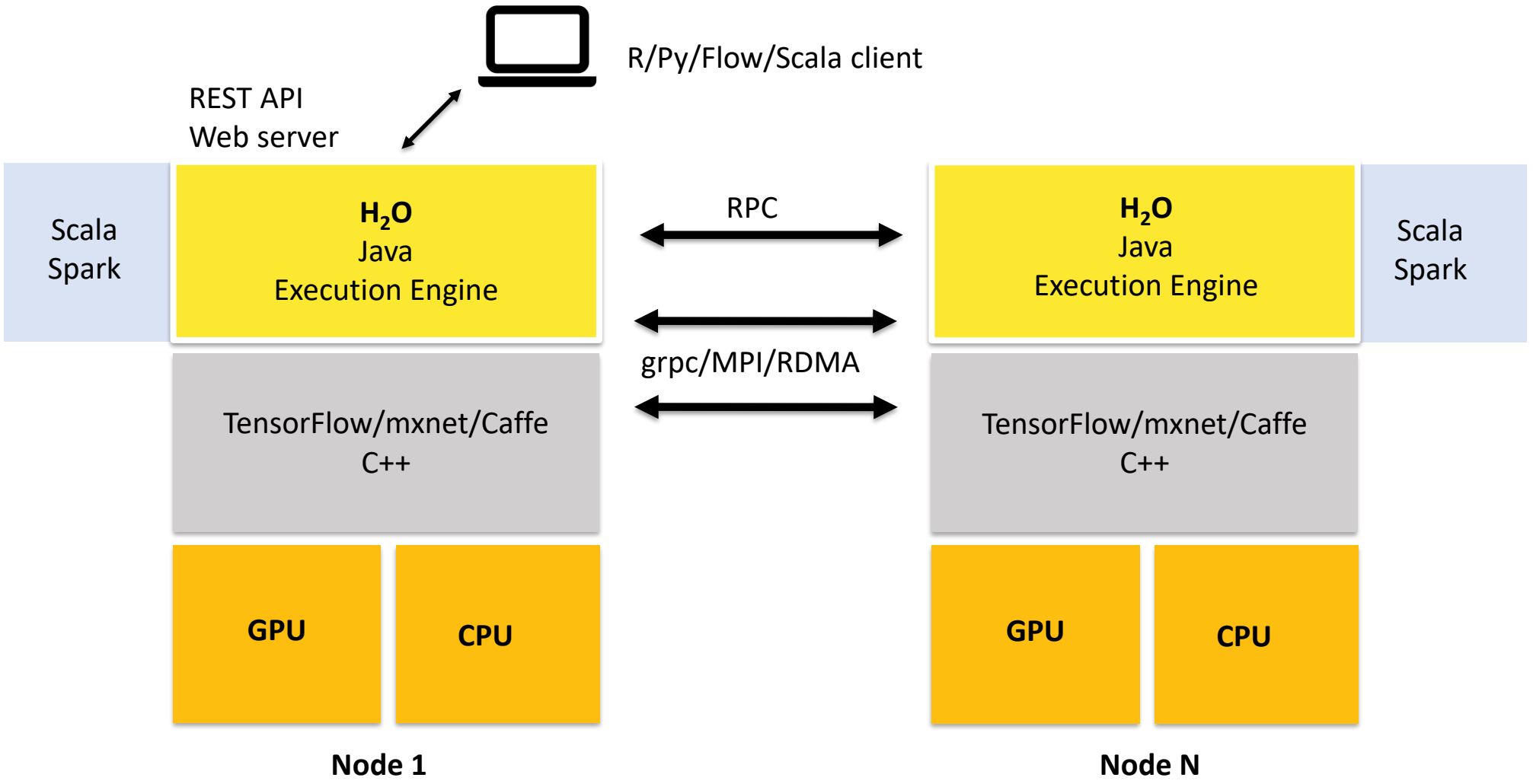
Convolutional Neural Networks enabling **Image, video, speech recognition**

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

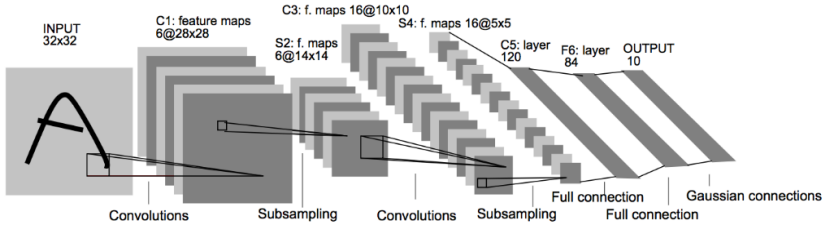
Deep Water Architecture



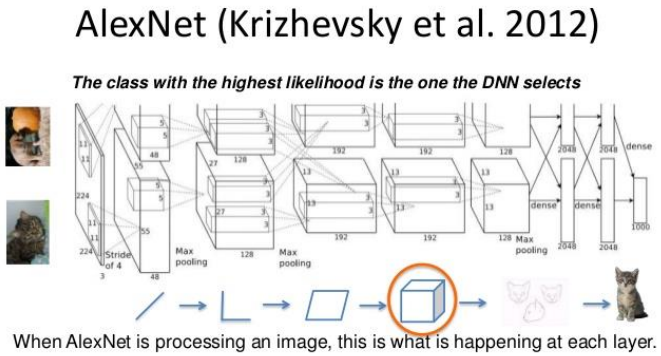
H2O.ai

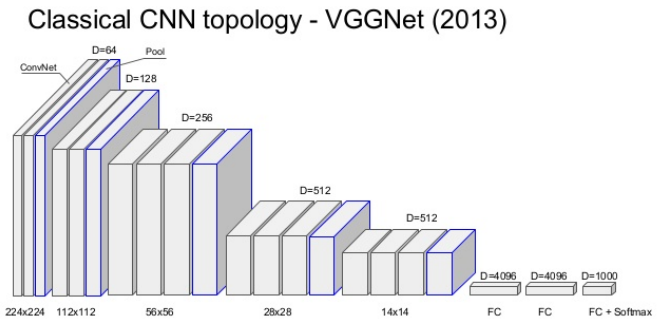
Available Networks in Deep Water

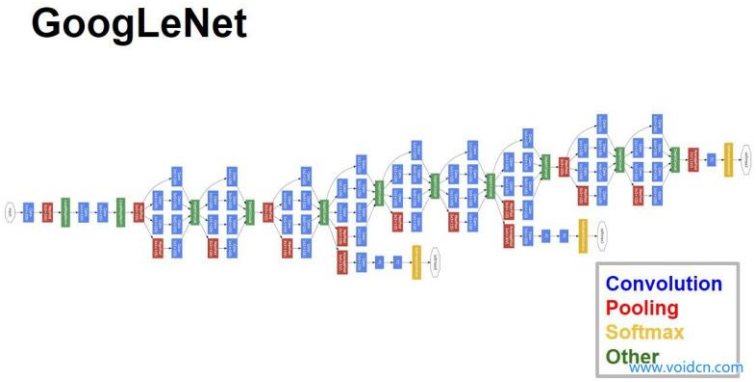
- LeNet
- AlexNet
- VGGNet
- Inception (GoogLeNet)
- ResNet (Deep Residual Learning)
- Build Your Own

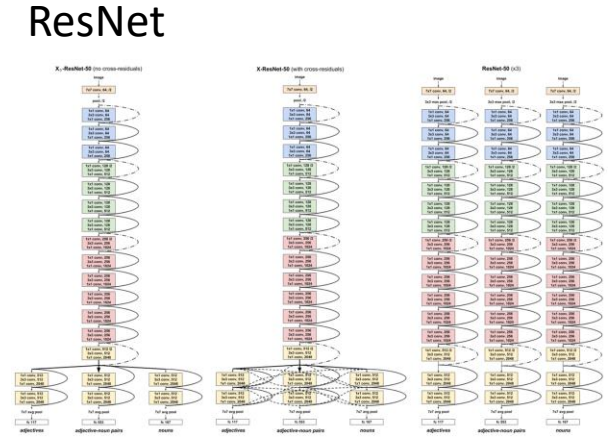


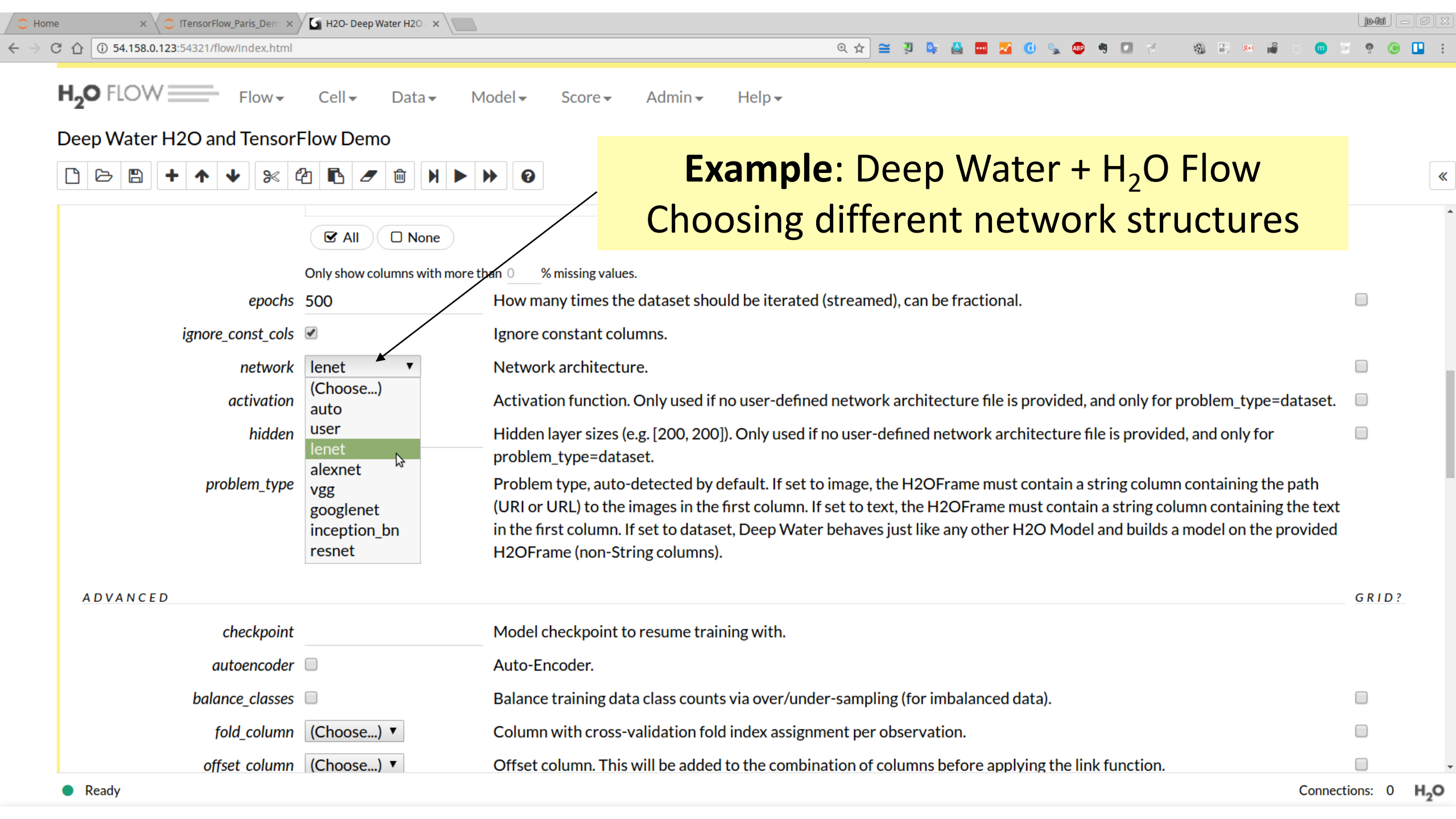
CNN called LeNet by Yann LeCun (1998)

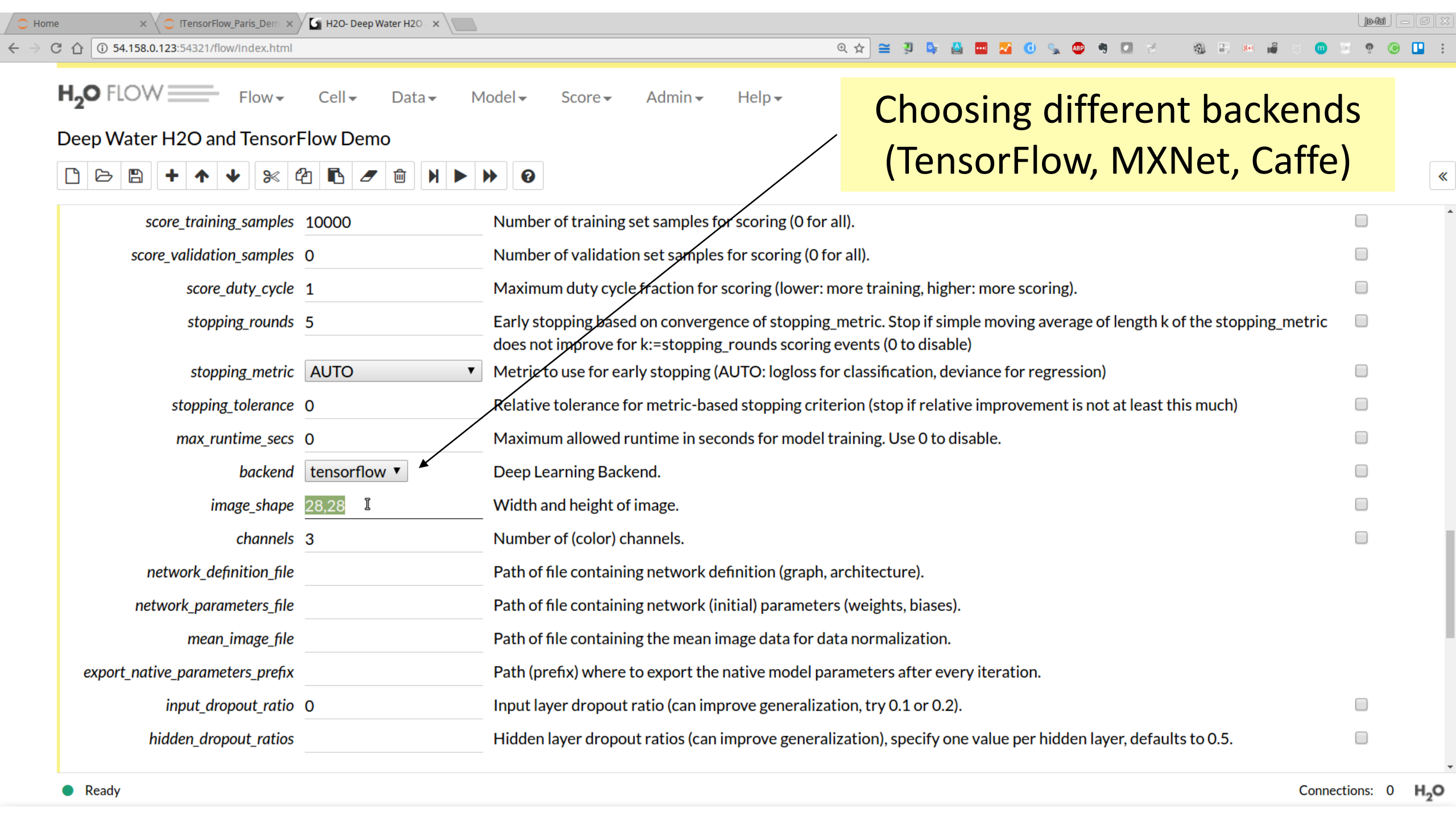









Home
ITensorFlow_Paris_Dem
H2O- Deep Water H2O
54.158.0.123:54321/flow/index.html
H2O FLOW
Flow
Cell
Data
Model
Score
Admin
Help
Deep Water H2O and TensorFlow Demo
Example: Deep Water + H2O Flow
Choosing different network structures
All
None
Only show columns with more than 0 % missing values.
epochs
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)

```
model <- h2o.deepwater(x=path, y=response,
                       training_frame=df, epochs=50,
                       learning_rate=1e-3, network = "lenet")
model
```

Choosing different network structures

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
```

H2O, Sparkling Water, Steam, & Deep Water Documentation

Getting Started | Data Science Algorithms | Languages | Tutorials, Examples, & Presentations | For Developers | For the Enterprise

docs.h2o.ai

Getting Started

H2O

What is H2O?
H2O User Guide
H2O Book (O'Reilly)
Recent Changes
Open Source License (Apache V2)

Quick Start Video - Flow Web UI
Quick Start Video - R
Quick Start Video - Python

Download H2O

Sparkling Water

What is Sparkling Water?
Sparkling Water Booklet
PySparkling Readme 2.0 | 1.6
RSparkling Readme
Open Source License (Apache V2)

Quick Start Video - Scala
Quick Start Video - Python

Download Sparkling Water

Steam

What is Steam?
Steam User Guide
Recent Changes
Open Source License (AGPL)

Download Steam

Deep Water (preview)

Deep Water Readme
Deep Water AMI Guide
Open Source License (Apache V2)

Launch Deep Water AMI
(choose g2.2xlarge)

Q & A

FAQ
Community Forum
h2ostream Google Group
Issue Tracking (JIRA)
Gitter
Stack Overflow
Cross Validated

For Supported Enterprise Customers

Enterprise Support Web | Email

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

Pre-Release Docker Image

We have a GPU-enabled Docker image on Docker Hub. To use it you need a Linux machine with at least one GPU, and with docker and nvidia-docker installed.

An **NVIDIA GPU** with a **Compute Capability of at least 3.5** is necessary. See https://developer.nvidia.com/cuda-gpus .

If you use **Amazon Web Services (AWS)**, a good machine type to use is the **P2** series. Note that G2 series machines have GPUs that are too old.

1. Install **Docker**, see http://www.docker.com
    - *Optional Step*. Make docker run without sudo. Instructions for Ubuntu 16.04:
        - `sudo groupadd docker`
        - `sudo gpasswd -a ${USER} docker`
        - `sudo service docker restart`
        - log out then log in, or `newgrp docker`
2. Install **nvidia-docker**, see https://github.com/NVIDIA/nvidia-docker . Note that you can only use Linux machines with one or more NVIDIA GPUs:
    - GNU/Linux x86_64 with kernel version > 3.10
    - Docker >= 1.9 (official docker-engine, docker-ce or docker-ee only)
    - NVIDIA GPU with Architecture > Fermi (2.1) and Compute Capability >= 3.5
    - NVIDIA drivers >= 340.29 with binary nvidia-modprobe
3. Download and run the H2O Docker image
    - `nvidia-docker run -it --net host -v $PWD:/host opsh2oai/h2o-deepwater`
    - You now get a prompt in the image: `#` . The directory you started from is avaiable as `/host`
    - Start H2O with `java -jar /opt/h2o.jar`
    - Python, R and Jupyter Notebooks are available
    - `exit` or `ctrl-d` closes the image

Docker Image

https://github.com/h2oai/deepwater

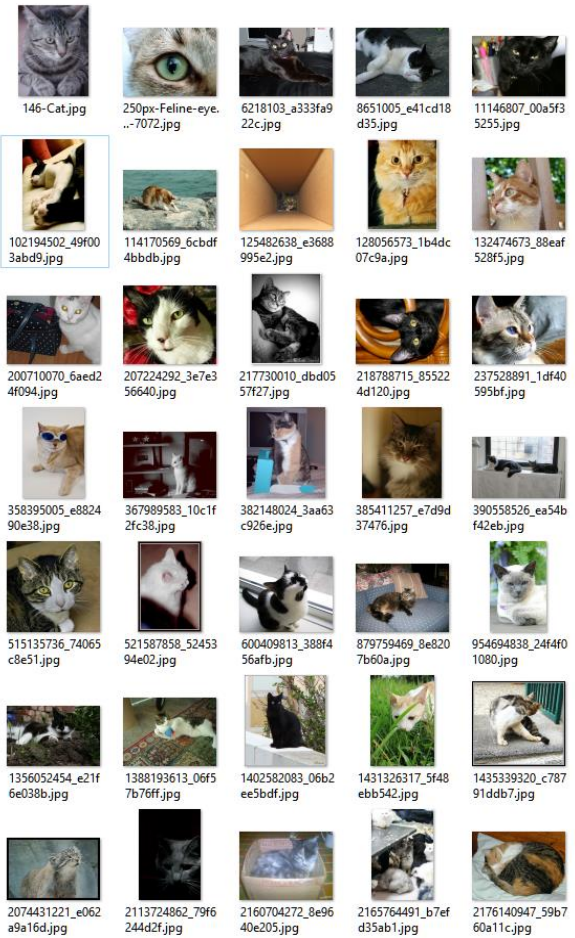
Deep Water Cat/Dog/Mouse Demo

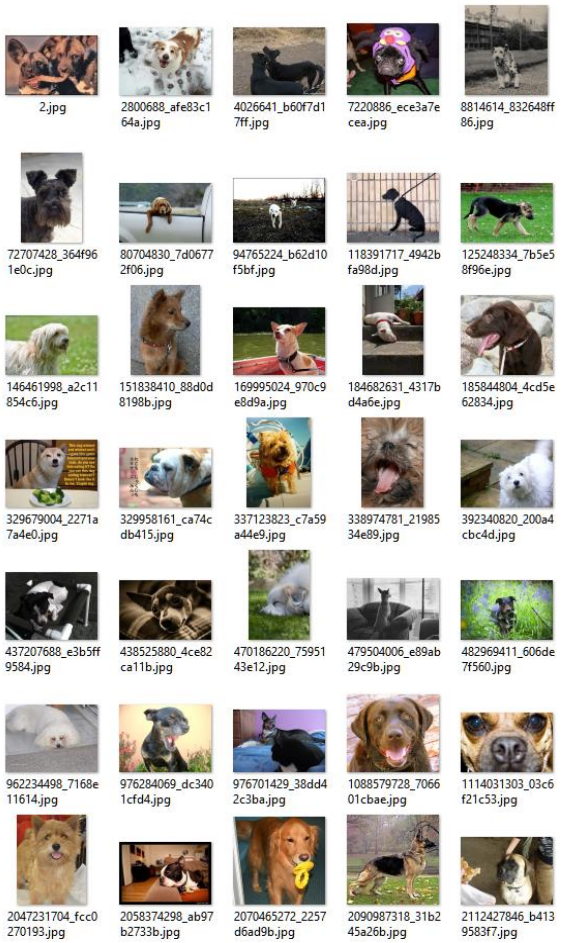
Deep Water R Demo

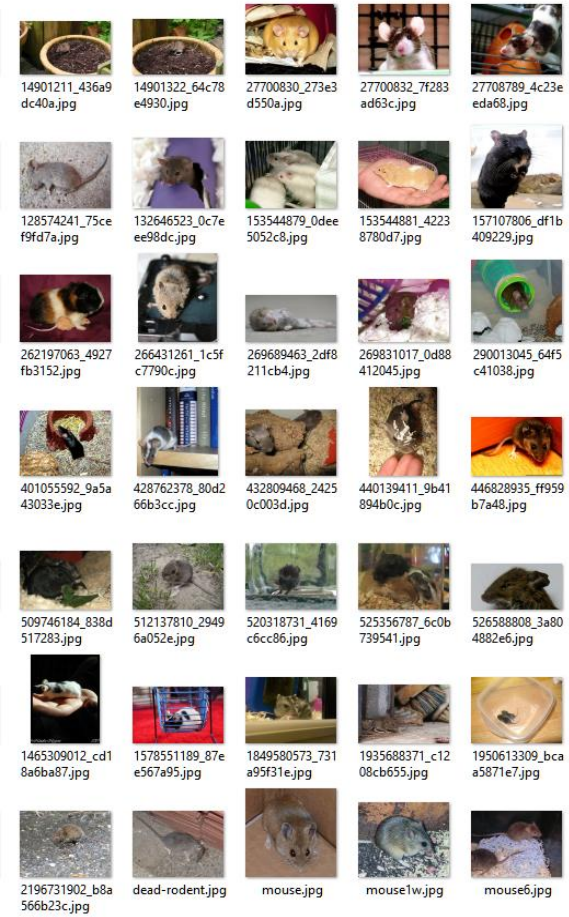
- H_2O + MXNet + TensorFlow
  - Dataset – Cat/Dog/Mouse
  - MXNet & TensorFlow as GPU backend
  - Train LeNet (CNN) models
  - R Demo

- 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

Start and Connect to H_2O Deep Water Cluster

- Download Latest Nightly Build
  - https://s3.amazonaws.com/h2o-deepwater/public/nightly/latest/h2o.jar

- In Terminal
  - cd to the folder containing h2o.jar
  - java –jar h2o.jar *(this is the default command)*
  - java –jar –Xmx16g h2o.jar *(this is the command to allocate 16GB of memory)*

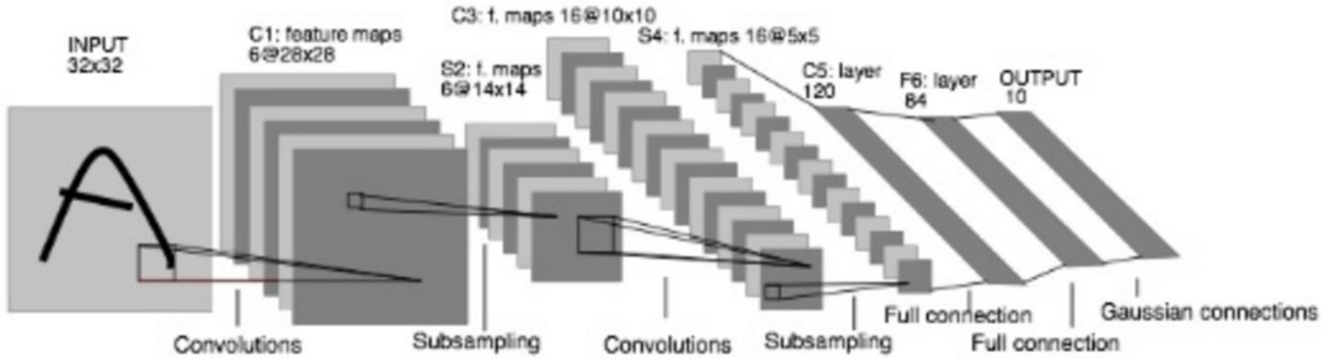
- In R
  - library(h2o) *(latest stable release from h2o.ai website or CRAN)*
  - h2o.connect(ip = “xxx.xxx.xxx.xxx”, strict_version_check = FALSE)

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



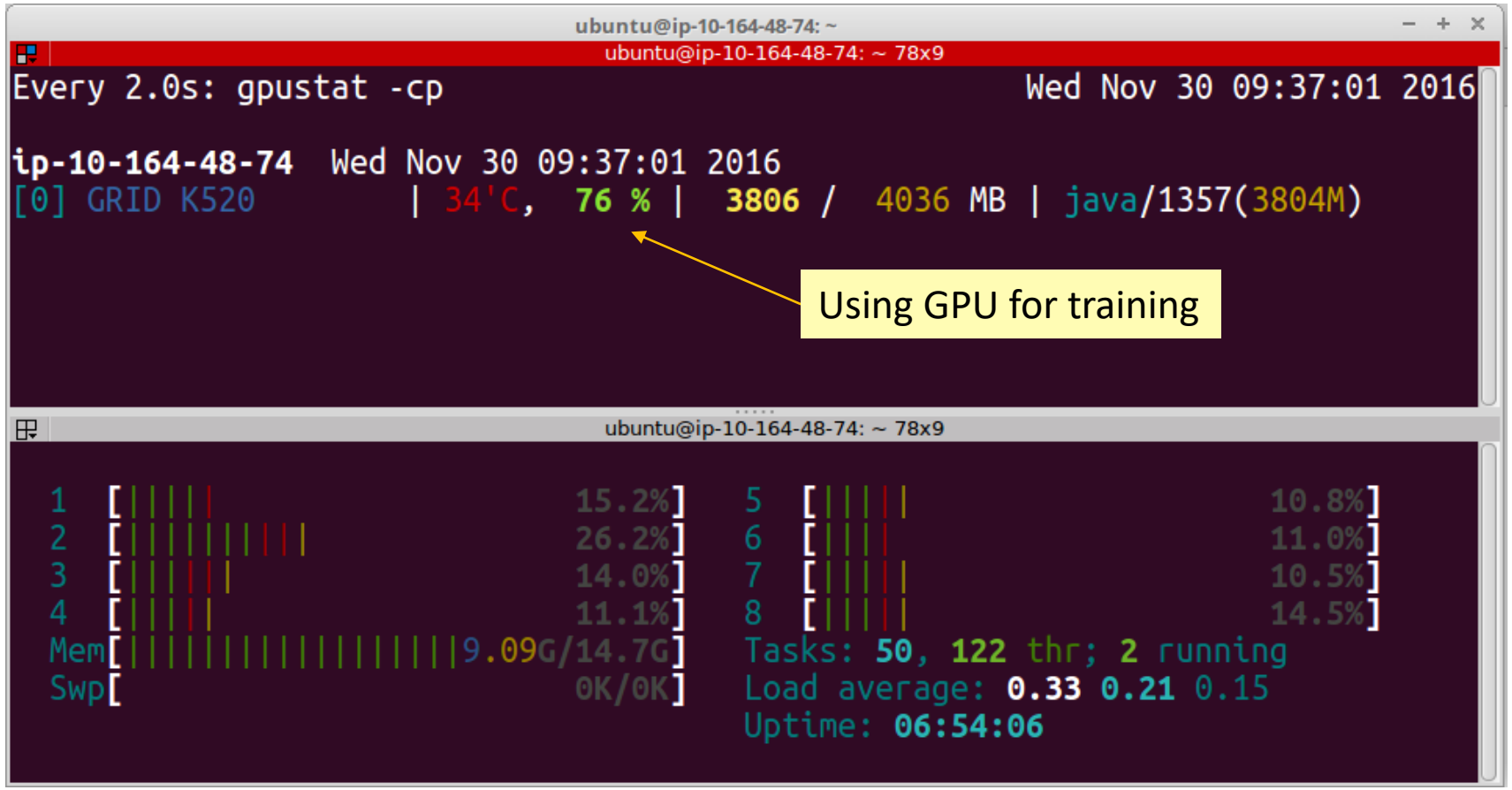
LeNet: a layered model composed of convolution and subsampling operations followed by a holistic representation and ultimately a classifier for handwritten digits. [Yann LeCun; LeNet]

We'll use a GPU to train such a LeNet model in seconds

To build a LeNet image classification model in H2O, simply specify network = "lenet":

```
model <- h2o.deepwater(x=path, y=response,
                       training_frame=df, epochs=50,
                       learning_rate=1e-3, network = "lenet")
```

Train a CNN (LeNet) Model on GPU



H2O.ai

Model

```
Model Details:
==============

H2OMultinomialModel: deepwater
Model ID:  DeepWater_model_R_1477378862430_2
Status of Deep Learning Model: lenet, 1.6 MB, predicting C2, 3-class classif
s, mini-batch size 32
  input_neurons     rate momentum
1          2352 0.000986 0.990000


H2OMultinomialMetrics: deepwater
** Reported on training data. **
** Metrics reported on full training frame **

Training Set Metrics:
=====================

Extract training frame with `h2o.getFrame("cat_dog_mouse.hex_sid_95f8_1")`
MSE: (Extract with `h2o.mse`) 0.131072
RMSE: (Extract with `h2o.rmse`) 0.3620386
Logloss: (Extract with `h2o.logloss`) 0.4176429
Mean Per-Class Error: 0.1165104
Confusion Matrix: Extract with `h2o.confusionMatrix(<model>,train = TRUE)`)
========================================================================
Confusion Matrix: vertical: actual; across: predicted
       cat dog mouse  Error       Rate
cat     75   4    11 0.1667 =  15 / 90
dog      4  75     6 0.1176 =  10 / 85
mouse    3   3    86 0.0652 =   6 / 92
Totals  82  82   103 0.1161 = 31 / 267
```

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

Model

```
Model Details:
==============

H2OMultinomialModel: deepwater
Model Key:  DeepWater_model_R_1477378862430_3
Status of Deep Learning Model: user, 1.6 MB, predicting C2, 3-class classifi
s, mini-batch size 32
  input_neurons     rate momentum
1          2352 0.004409 0.990000

H2OMultinomialMetrics: deepwater
** Reported on training data. **
** Metrics reported on full training frame **

Training Set Metrics:
=====================

Extract training frame with `h2o.getFrame("cat_dog_mouse.hex_sid_95f8_1")`
MSE: (Extract with `h2o.mse`) 0.03078524
RMSE: (Extract with `h2o.rmse`) 0.1754572
Logloss: (Extract with `h2o.logloss`) 0.1154222
Mean Per-Class Error: 0.03366487
Confusion Matrix: Extract with `h2o.confusionMatrix(<model>,train = TRUE)`)
========================================================================
Confusion Matrix: vertical: actual; across: predicted
       cat dog mouse  Error      Rate
cat     88   2     0 0.0222 =  2 / 90
dog      2  82     1 0.0353 =  3 / 85
mouse    1   3    88 0.0435 =  4 / 92
Totals  91  87    89 0.0337 = 9 / 267
```

Note: Overfitting is expected as we only use a very small datasets to demonstrate the APIs only

H2O.ai

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.

- Latest Nightly Build
  - https://s3.amazonaws.com/h2o-deepwater/public/nightly/latest/h2o.jar
- 100% Open Source
  - The party will get bigger!

H2O.ai

Latest H_2O Developments

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

H_2O + xgboost

Stacked Ensembles of H2O and XGBoost Models

Erin LeDell

https://github.com/h2oai/h2o-tutorials/tree/master/tutorials/ensembles-stacking

4/25/2017

This tutorial will demonstrate how to use the **h2o** R package to combine H2O models with XGBoost models into a Stacked Ensemble.

Install XGBoost-enabled H2O

Currently, XGBoost is available in a special development edition of H2O. The Mac OS X version (with XGBoost compiled for Mac) is available (temporarily) here. Download the file, unzip it, and install the R package: `R CMD install ./R/h2o_3.11.0.99999.tar.gz`

H2O ships with everything except the system library for multithreading (openMP). On a Mac, you will need to install OpenMP, which you get easily via Homebrew.

```
# Install OpenMP (required of xgboost-enabled h2o)
brew install gcc --without-multilib
```

Once the special edition **h2o** R package is installed, you are all set to start training H2O and XGBoost models from H2O.

Train Base Learners

Let's train and cross-validate a set of H2O and XGBoost models and then create a Stacked Ensemble using the **h2o** R package.

Start H2O Cluster & Load Data

```
library(h2o)
```

```
##
```

H_2O's wrapper for xgboost in R

```
# Train & Cross-validate a XGB-GBM
my_xgb1 <- h2o.xgboost(x = x,
                       y = y,
                       training_frame = train,
                       distribution = "bernoulli",
                       ntrees = 50,
                       max_depth = 3,
                       min_rows = 2,
                       learn_rate = 0.2,
                       nfolds = nfolds,
                       fold_assignment = "Modulo",
                       keep_cross_validation_predictions = TRUE,
                       seed = 1)
```

https://github.com/h2oai/h2o-tutorials/tree/master/tutorials/ensembles-stacking

Automatic Machine Learning

Model parameters tuning and stacking automated

H_2O AutoML

- AutoML stands for "Automatic Machine Learning"
- The idea here is to remove most (or all) of the parameters from the algorithm, as well as automatically generate derived features that will aid in learning.
- Single algorithms are tuned automatically using a carefully constructed random grid search (future: Bayesian Optimization algorithms).
- Optionally, a Stacked Ensemble can be constructed.

```r
# Import a sample binary outcome train/test set into H2O
train <- h2o.importFile("https://s3.amazonaws.com/erin-data/higgs/higgs_train_10k.csv")
test <- h2o.importFile("https://s3.amazonaws.com/erin-data/higgs/higgs_test_5k.csv")

# Identify predictors and response
y <- "response"
x <- setdiff(names(train), y)  #x is not required if you define it as everything except y

# For binary classification, response should be a factor
train[,y] <- as.factor(train[,y])
test[,y] <- as.factor(test[,y])
```

Train AutoML

Here we will let AutoML train for 360 seconds (6 minutes). The default is 600 seconds (10 minutes).

```r
aml <- h2o.automl(x = x,  #Note: x is not required if you define it as everything except y
                  y = y,
                  training_frame = train,
                  test_frame = test,
                  max_runtime_secs = 360)
```

H_2O AutoML

```
05-05 16:46:30.225 172.16.2.68:54321     16609  FJ-1-13   INFO: Leaderboard for project automl_RTMP_sid_b58c_124 (models sorted in order of auc, best first):
05-05 16:46:30.225 172.16.2.68:54321     16609  FJ-1-13   INFO: #                                                  model_id       auc   logloss
05-05 16:46:30.225 172.16.2.68:54321     16609  FJ-1-13   INFO: 0  GLM_grid__bb21707f30f51276607846fb290f4346_model_0  0.836905  0.483975
05-05 16:46:30.225 172.16.2.68:54321     16609  FJ-1-13   INFO: 1  GLM_grid__bb21707f30f51276607846fb290f4346_model_1  0.836905  0.483975
05-05 16:46:30.225 172.16.2.68:54321     16609  FJ-1-13   INFO: 2  StackedEnsemble_model_1494027949268_1131            0.828571  0.502921
05-05 16:46:30.225 172.16.2.68:54321     16609  FJ-1-13   INFO: 3  GBM_grid__bb21707f30f51276607846fb290f4346_model_1  0.789286  0.542307
05-05 16:46:30.225 172.16.2.68:54321     16609  FJ-1-13   INFO: 4  DRF_model_1494027949268_3                           0.770833  0.559774
05-05 16:46:30.225 172.16.2.68:54321     16609  FJ-1-13   INFO: 5  GBM_grid__bb21707f30f51276607846fb290f4346_model_0  0.759524  1.491769
05-05 16:46:30.225 172.16.2.68:54321     16609  FJ-1-13   INFO: 6  XRT_model_1494027949268_433                         0.694048  0.603735
```

Note: We are in final stages of testing and performance tuning. AutoML will be merged into H_2O master branch soon.

word2vec

For natural language processing

Word2Vec

- Learns vector representations of words by analyzing large text corpus
  - Word Embeddings = mapping of words to vectors from a high dimensional space (100-1000)
  - Text sources: Google News, Wikipedia, Tweets, …

- Embeddings capture meaning of the word
  - Semantically similar words are close to each other
  - Can be used to find synonyms & analogies
  - Famous example:
    - » man is to woman as king is to *???* (queen)
    - » Vec(king) – Vec(man) + Vec(woman) = Vec(queen)

car

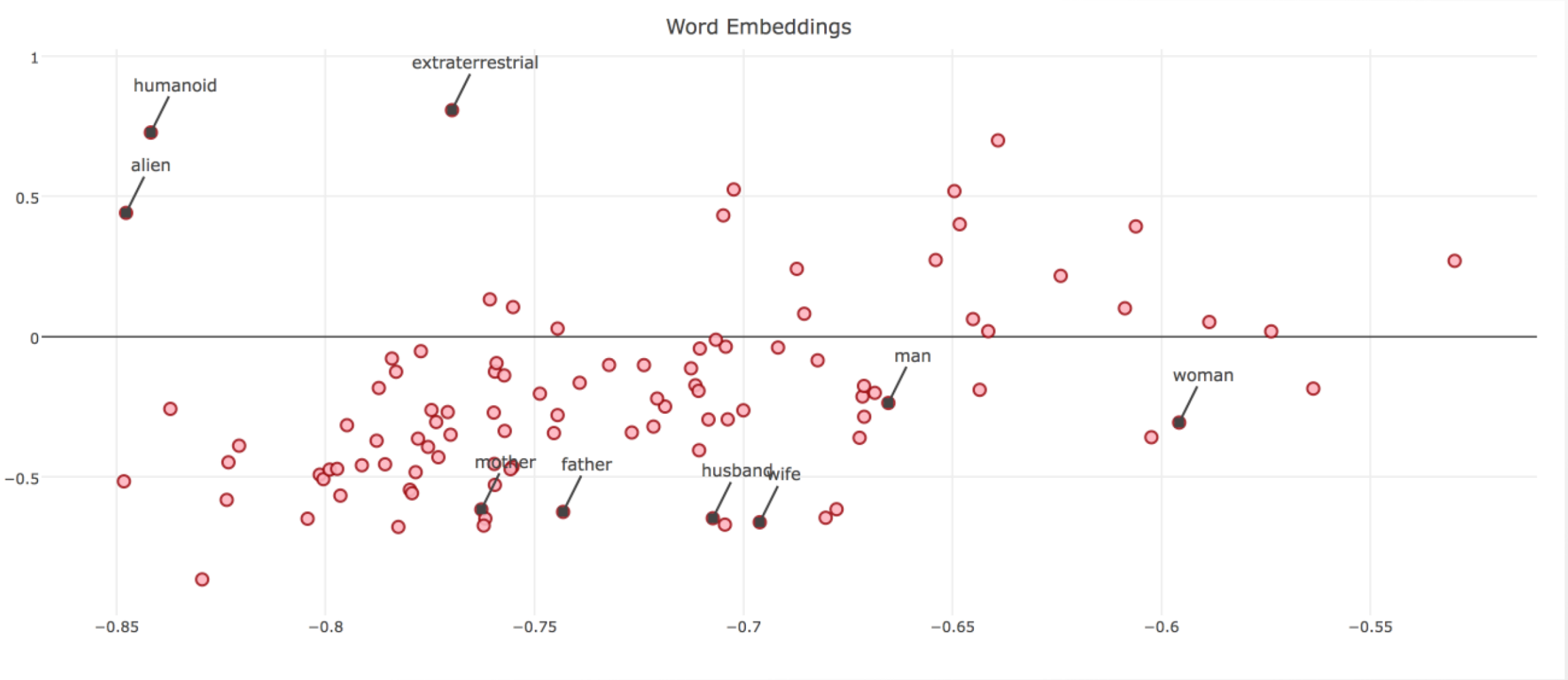
-0.09	-0.14	-0.12	-0.06	0.16

alien

-0.38	-0.11	0.10	-0.26	-0.24

- Different variations of word2vec
  - Skip-Gram, CBOW
  - Hierarchical Softmax, Negative Sampling

Word2Vec


Word Embeddings
1
0.5
0
−0.5
−0.85
−0.8
−0.75
−0.7
−0.65
−0.6
−0.55
humanoid
extraterrestrial
alien
man
woman
mother
father
husband
wife

In [8]:
```
# Download pre-trained Wikipedia 2014 + Gigaword 5 word vectors
# (6B tokens, 400K vocab, uncased, 50d, 100d, 200d, & 300d vectors, 822 MB)
# Reference: https://nlp.stanford.edu/projects/glove/
# Link: http://nlp.stanford.edu/data/glove.6B.zip
```

In [9]:
```
# Import pre-trained vectors as H2O frame
h_pretrained <- h2o.importFile("glove.6B.300d.txt")
```

In [10]:
```
# Convert pre-trained vectors to H2O word2vec model
n_vec <- ncol(h_pretrained) - 1
model_w2v <- h2o.word2vec(pre_trained = h_pretrained, vec_size = n_vec)
```

```
  |======================================================================| 100%
```

In [11]:
```
# Sanity Check: 'h2o'
print(h2o.findSynonyms(model_w2v, "h2o", count = 5))
```

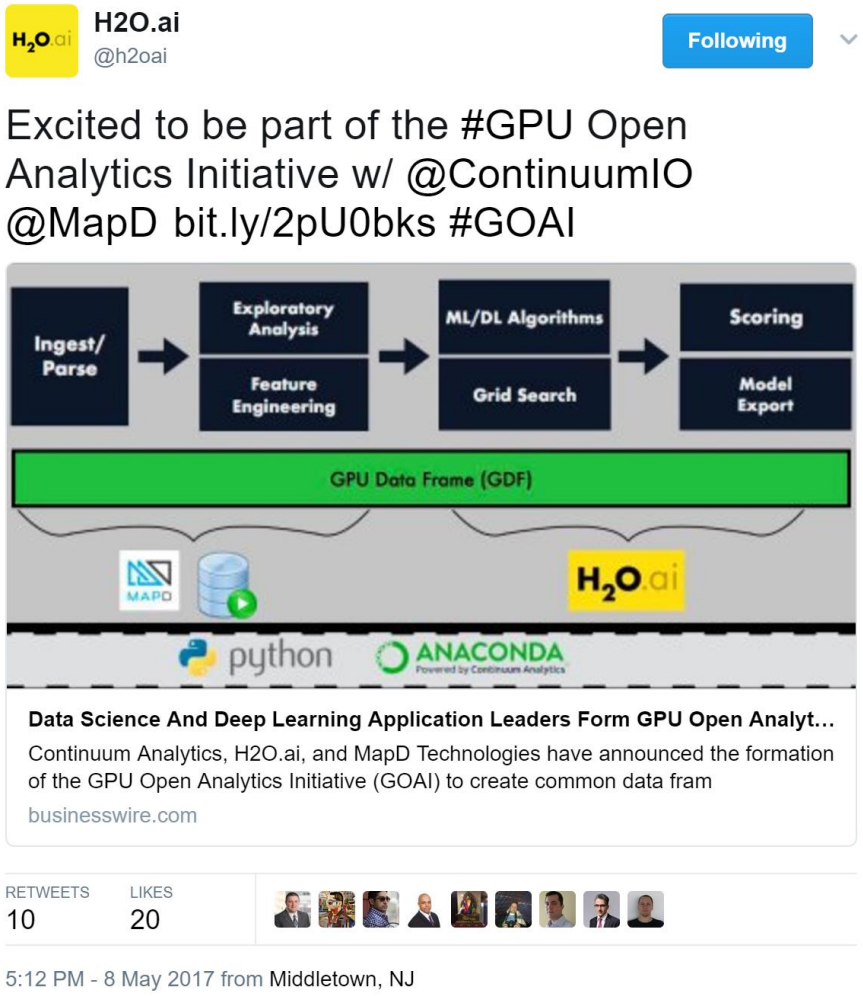
```
      synonym     score
1         nh3 0.6126952
2        h2o2 0.5957690
3        nadp 0.5485679
4  substrates 0.5450500
5 diphosphate 0.5159003
```

In [12]:
```
# Sanity Check: 'water'
print(h2o.findSynonyms(model_w2v, "water", count = 5))
```

```
     synonym     score
1   drinking 0.5925374
2     sewage 0.5703815
3 irrigation 0.5549443
4    potable 0.5404900
5     waters 0.5361896
```

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]
- Model Interpretation [Link]
- word2vec [Link]



H2O.ai

Хвала вам

- Organizers & Sponsors
  - Branko Kovač & BelgradeR
  - Seven Bridges

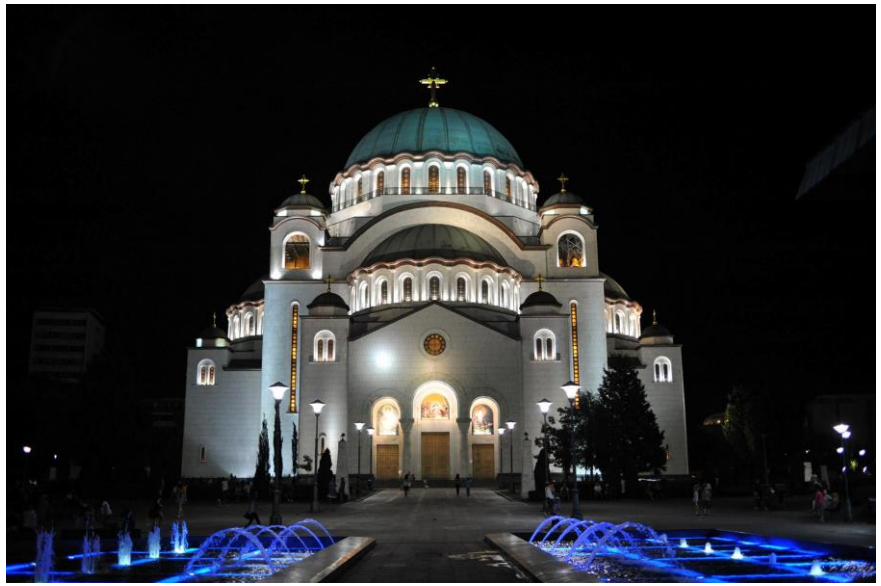


Photo Credit: Jo-fai Chow (2012)

- 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)