

Introduction to Machine Learning with H2O



Jo-fai (Joe) Chow
Data Scientist
joe@h2o.ai
[@matlabulous](https://twitter.com/matlabulous)

Data Science Milan
Politecnico di Milano
10th October, 2016

About Me: Civil Engineer → Data Scientist

- 2005 - 2015
 - Water Engineer
 - Consultant for Utilities
 - Industrial PhD
 - Water Engineering + Machine Learning
- 2015 - Present
 - Data Scientist
 - Virgin Media (UK)
 - Domino Data Lab (US)
 - H2O.ai (US)



Why? Long story – see bit.ly/joe_h2o_talk2

Agenda

- First Talk (25 mins)
 - About H2O.ai
 - Demo
 - A Simple Classification Task
 - H2O's Web Interface
 - Why H2O?
 - Our Community
 - Our Customers
 - What's Next?
 - New H2O Features
- Second Talk (25 mins)
 - H2O for IoT
 - Predictive Maintenance
 - Anomaly Detection
 - H2O's R Interface
- Third Talk (25 mins)
 - Deep Water
 - Demo
 - H2O + mxnet on GPU
 - H2O's Python Interface

About H2O.ai



About H2O.ai

- H2O.ai, the Company
 - Team: 80 (70 shown)
 - Founded in 2012
 - HQ: Mountain View, California
- H2O, the Platform
 - Open Source (Apache 2.0)
 - Algorithms written in Java
 - Fast, distributed and scalable
 - Multiple interfaces to suit different users
 - Web, R, Python, Java, Scala, REST/JSON
 - Works with desktop/laptop, cloud, Spark and Hadoop

Joe



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*



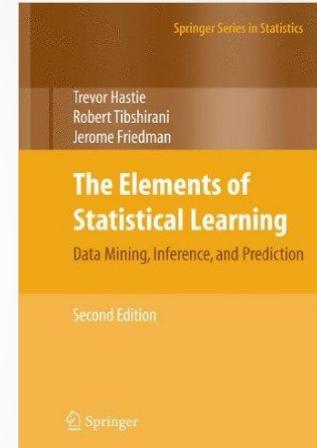
Dr. Robert Tibshirani

- 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*



Current Algorithm Overview

Statistical Analysis

- Linear Models (GLM)
- Naïve Bayes

Ensembles

- Random Forest
- Distributed Trees
- Gradient Boosting Machine
- R Package - Stacking / Super Learner

Today's Demos

Deep Neural Networks

- Multi-layer Feed-Forward Neural Network
- Auto-encoder
- Anomaly Detection
- Deep Features

Clustering

- K-Means

Dimension Reduction

- Principal Component Analysis
- Generalized Low Rank Models

Joe's Strata Hadoop London Talk
bit.ly/joe_h2o_talk4

Solvers & Optimization

- Generalized ADMM Solver
- L-BFGS (Quasi Newton Method)
- Ordinary least-Square Solver
- Stochastic Gradient Descent

Data Munging

- Scalable Data Frames
- Sort, Slice, Log Transform

Joe's LondonR Talk
bit.ly/joe_h2o_talk3

H2O Overview

Computer Science (CS)

Artificial Intelligence (A.I.)

Machine Learning (ML)

Deep Learning (DL)

hot hot hot hot hot



H2O's Mission

Making Machine Learning Accessible to Everyone



Photo credit: Virgin Media

H2O Web Interface Demo



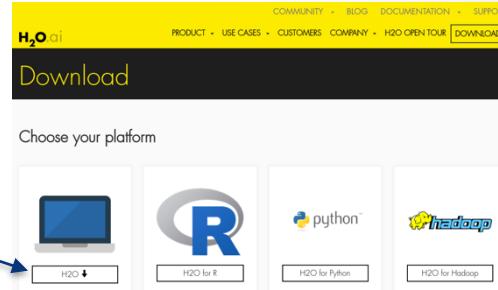
A Typical Machine Learning Task

- Demo
 - Dataset – MNIST
 - LeCun et al. (1999)
 - Hand-written Digits
 - Import & Explore Data
 - Build & Evaluate Models
 - Make Predictions



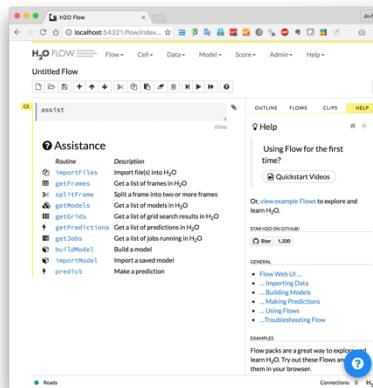
H2O Flow (Web Interface) Demo

- Download and unzip jar from www.h2o.ai



- In terminal:
 - java -jar h2o.jar
- Web browser:
 - localhost:54321

```
Jo-fais-MacBook-Pro-2:~ jofaichow$ cd h2o-3.10.0.6
Jo-fais-MacBook-Pro-2:h2o-3.10.0.6 jofaichow$ java -jar h2o.jar
09-18 13:16:13.620 192.168.0.6:54321 8620 main INFO: ----- H2O started -----
09-18 13:16:13.636 192.168.0.6:54321 8620 main INFO: Build git branch: rel-turing
09-18 13:16:13.636 192.168.0.6:54321 8620 main INFO: Build git hash: 3b286dea7b719b6ef2c2f5f7728648f2440a1502
09-18 13:16:13.636 192.168.0.6:54321 8620 main INFO: Build git describe: jenkins-rel-turing-6
09-18 13:16:13.636 192.168.0.6:54321 8620 main INFO: Build project version: 3.10.0.6 (latest version: 3.10.0.6)
```



H2O Live Demo



More H2O Flow Examples

The screenshot shows the H2O Flow web application running in a browser. The title bar reads "H2O Flow". The top navigation bar includes "Flow", "Cell", "Data", "Model", "Score", "Admin", and "Help". Below the navigation is a toolbar with various icons for file operations like import, export, and search. The main workspace is titled "Untitled Flow" and contains a search bar with the text "assist". A list of routines is displayed under the heading "Assistance":

Routine	Description
importFiles	Import file(s) into H ₂ O
getFrames	Get a list of frames in H ₂ O
splitFrame	Split a frame into two or more frames
getModels	Get a list of models in H ₂ O
getGrids	Get a list of grid search results in H ₂ O
getPredictions	Get a list of predictions in H ₂ O
getJobs	Get a list of jobs running in H ₂ O
buildModel	Build a model
importModel	Import a saved model
predict	Make a prediction

The status bar at the bottom left says "Ready". On the right side, there is a sidebar with tabs for "OUTLINE", "FLOWS", "CLIPS", and "HELP". The "HELP" tab is highlighted with a yellow background. The sidebar content lists various examples:

- PACK
- [examples](#)
- [GBM_Example.flow](#)
- [DeepLearning_MNIST.flow](#)
- [GLM_Example.flow](#)
- [DRF_Example.flow](#)
- [K-Means_Example.flow](#)
- [Million_Songs.flow](#)
- [KDDCup2009_Churn.flow](#)
- [QuickStartVideos.flow](#)
- [Airlines_Delav.flow](#)

A blue circular button with a question mark icon is located in the bottom right corner. The status bar at the bottom right shows "Connections: 0" and the H₂O logo.

Other H2O Interfaces

- R

```
1 # Load H2O R package
2 library(h2o)
3
4 # Initialize and Connect to H2O
5 h2o.init()
```

- Python

```
1 # Import H2O Python module
2 import h2o
3
4 # Initialize and Connect to H2O
5 h2o.init()
```

- **docs.h2o.ai**

Key Resources

H2O, Sparkling Water, and Steam Documentation

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

Getting Started

H2O

What is H2O?
H2O User Guide
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
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

Questions and Answers

FAQ
Community Forum
H2Ostream Google Group
Issue Tracking (JIRA)
Gitter
Stack Overflow
Cross Validated

For Supported Enterprise Customers
Enterprise Support via Web | Email

Data Science Algorithms

Supervised Learning

Generalized Linear Modeling (GLM)	Tutorial	Booklet	Reference	Tuning
Gradient Boosting Machine (GBM)	Tutorial	Booklet	Reference	Tuning
Deep Learning	Tutorial	Booklet	Reference	Tuning
Distributed Random Forest	Tutorial	Booklet	Reference	Tuning
Naive Bayes	Tutorial	Booklet	Reference	Tuning
Ensembles (Stacking)	Tutorial	Booklet	Reference	Tuning

Unsupervised Learning

Generalized Low Rank Models (GLRM)	Tutorial	Reference
K-Means Clustering	Tutorial	Reference
Principal Components Analysis (PCA)	Tutorial	Reference

Languages

R

Quick Start Video - R
R Package Docs
R Booklet
Examples and Demos
R FAQ

Python

Quick Start Video - Python
Python Module Docs
Python Booklet
Examples and Demos
Python FAQ

Java

POJO Model Javadoc
H2O Core Javadoc
H2O Algorithms Javadoc

Scala

Sparkling Water API
Sparkling Water Scaladoc
H2O Scaladoc

More Advanced Topics

- Advanced Features
 - Hyperparameters Tuning
 - Model Stacking
 - Saving/Loading Models
 - Export Plain Old Java Object (POJO)
- Key Resources
 - docs.h2o.ai
- Joe's Previous H2O Talks
 - bit.ly/joe_h2o_talk3
 - bit.ly/h2o_budapest_1
 - bit.ly/h2o_paris_1

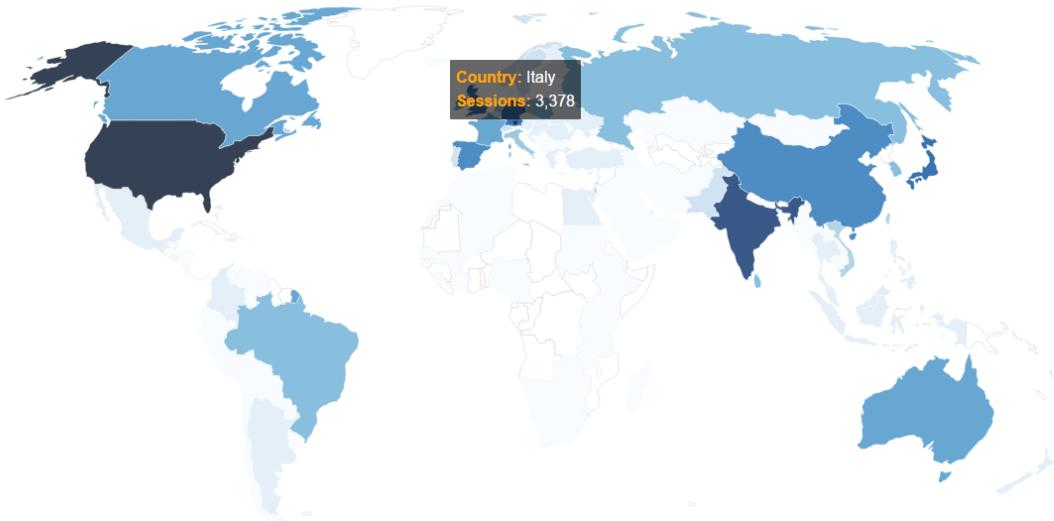
Why H2O?





H2O Usage in Italy

H2O Worldwide Usage



www.h2o.ai/community

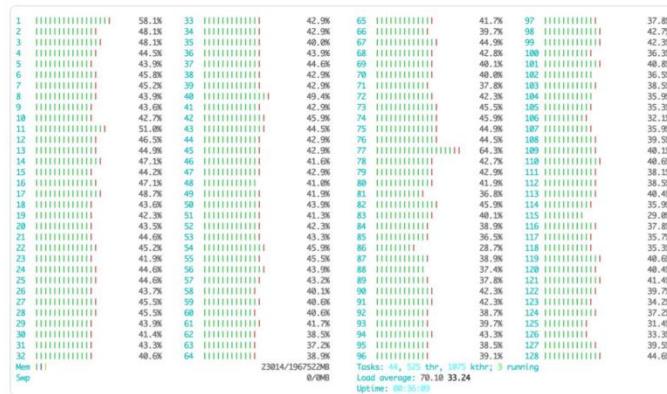
65,000 users, 7,000 organizations

Szilard Pafka – Chief Data Scientist at Epoch

- Sziland's talks / blog posts about H2O:
 - ML Benchmark
 - Intro to ML with H2O
 - H2O Scoring
 - Tweets



Also @h2oai on monster 2TB RAM 128 cores
EC2 X1 #bigdata #machinelearning
#datascience twitter.com/DataScienceLA/ ...



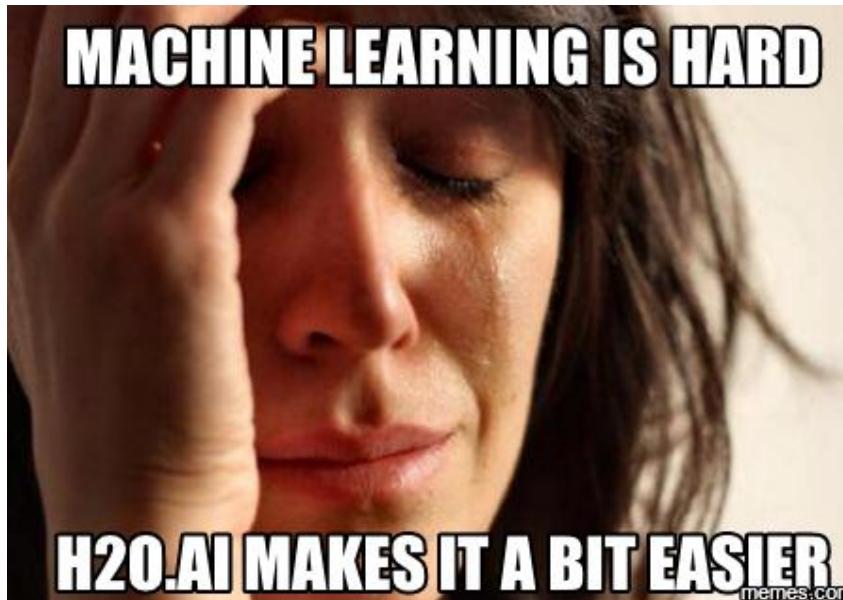
RETWEETS 10 LIKES 28

5:55 PM - 24 Jun 2016

21

Szilard Pafka – Why H2O?

- Szilard's Summary Slide



H2O for Kaggle

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

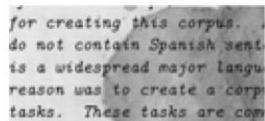
Triskelion | 01.02.2015

[READ MORE](#)

Kaggle challenge 2nd place winner Colin Priest

[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.”



Completed • Knowledge • 161 teams

Denoising Dirty Documents

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

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



H2O Community Support

Google forum – h2osteam

The screenshot shows a Google forum group named "h2osteam". The main post discusses the release of Steam and compatibility issues with Python 2.6. Below the post, there are two other recent posts: one about using the API to export models and another about decoding autoencoder outputs.

Groups

- My groups
- Home
- Starred
- Favourites
- Click on a group's star icon to add it to your favourites
- Recently viewed
- H2O Open Sour...
- packrat-discuss
- devtools
- Caffe Users
- ggplot2
- Recent searches
- spark streaming (i...)
- chord diagram gg...
- chord diagram (in ...)
- Pictaculous API (i...)
- pictaculous (in ma...
- Recently posted to
- H2O Open Sour...

Privacy - Terms of Service

community.h2o.ai

Please try

The screenshot shows the community.h2o.ai website. It features a sidebar with navigation links like "Ask a question", "Post an idea", and "Create an article". A central feed displays several posts from users like Avkash and windows, discussing topics such as Steam release, H2O Python Modules, H2O Installation, PySparkling launch problems, Predicted Values, and combining holdout predictions. A prominent announcement at the bottom right announces the "Sparkling Water Release 08/30".

All Posts

- Avkash answered When is Steam going to be released? 3 days ago in Steam
- windows commented H2O Python Modules 3 days ago in H2O
- windows commented H2O Installation 3 days ago in H2O
- Avkash Chauhan edited PySparkling launch problem with Python 2.6 or older 3 days ago in Python
- Avkash answered Predicted Values 3 days ago in H2O
- erin edited Combinig holdout predictions, while keep_cross_validation_predictions parameter is active in Python 0 Likes

Sparkling Water Release 08/30
We are happy to announce that Sparkling Water 2.0 release is almost here. On September 1, 2016 we will release Sparkling Water 2.0. Download info is coming soon.

Create

- Algorithms
- Announcements
- Artificial Intelligence
- Deep Water
- Demos
- H2O
- Java
- Machine Learning
- Python
- R
- Source Code
- Sparkling Water
- Steam
- Tools
- Troubleshooting

#AroundTheWorldWithH2Oai

London Kaggle Meetup



Strata Hadoop London



Chelsea FC



Big Data London



PyData Amsterdam



useR! 2016 Stanford



satRdays Budapest



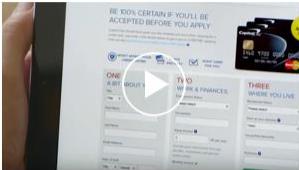
Paris ML Meetup

Stories of AI Transformation

H2O In Action

www.h2o.ai/customers

Capital One



Capital One uses H2O open source machine learning for various use cases.

MarketShare



Predicting Marketing Results Through Analytics

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.

Progressive



Progressive uses H2O predictive analytics for user-based insurance.

Comcast



Comcast uses H2O to improve customer experience.

Hospital Corporation of America



HCA uses H2O to predict patient outcomes in real-time.

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 a product recommendation platform for insurance.

Paypal



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

eBay



eBay chose H2O for open source machine learning.

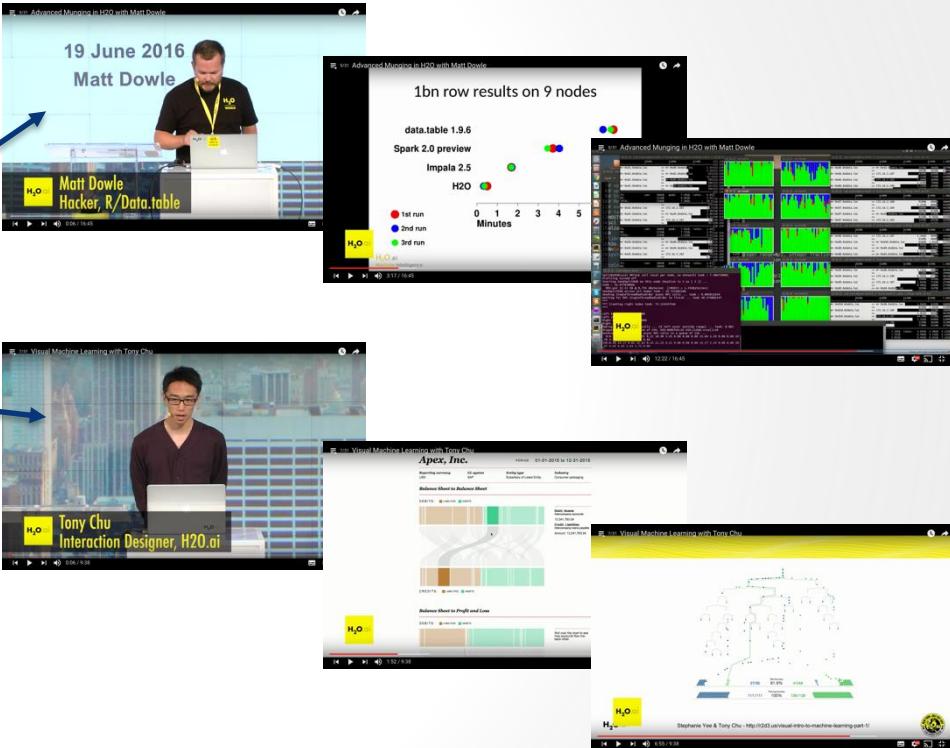
What's Next?



H2O is Evolving

- H2O Open Tour NYC YouTube Playlist
 - Advanced data munging
 - Visual ML
 - Deep Water (3rd talk)
 - Sparkling Water
 - PySparkling & RSparkling
 - Steam

Next time?



H2O's Mission

Making Machine Learning Accessible to Everyone

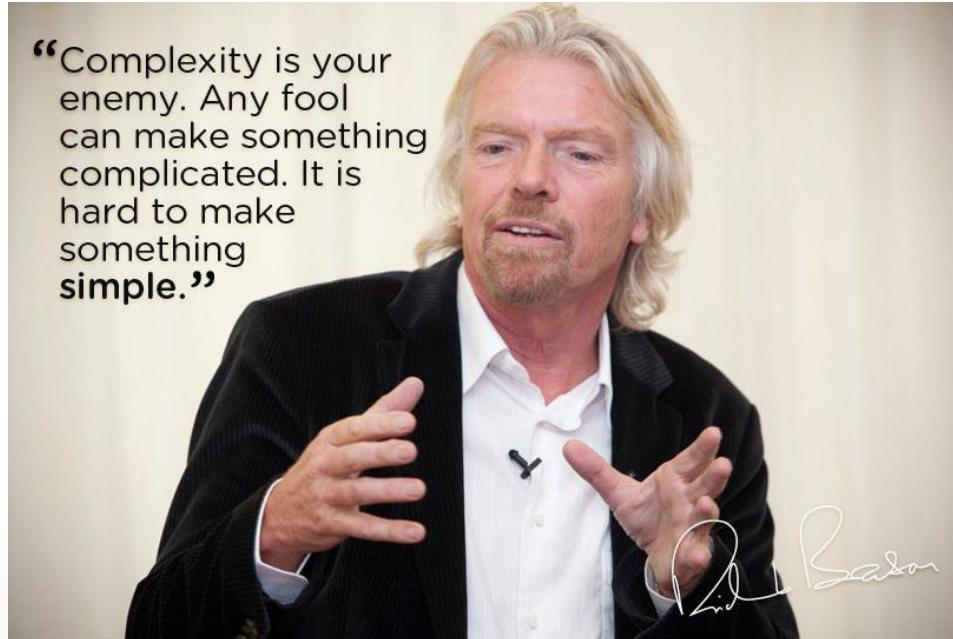


Photo credit: Virgin Media

End of First Talk – Thanks!

- Data Science Milan
- Gianmario Spacagna
- Politecnico di Milano
- Resources
 - bit.ly/h2o_milan_1
 - www.h2o.ai
 - docs.h2o.ai
- Contact
 - joe@h2o.ai
 - [@matlabulous](https://twitter.com/matlabulous)
 - github.com/woobe

Extra Slides

(H2O Flow Demo Screenshots – just in case)

H₂O FLOW

Flow ▾

Cell ▾

Data

Model

Score

Admin

Help

Untitled Flow



CS

assist

161ms

?

 Assistance

Routine	Description
importFiles	Import file(s) into H ₂ O
getFrames	Get a list of frames in H ₂ O
splitFrame	Split a frame into two or more frames
getModels	Get a list of models in H ₂ O
getGrids	Get a list of grid search results in H ₂ O
getPredictions	Get a list of predictions in H ₂ O
getJobs	Get a list of jobs running in H ₂ O
buildModel	Build a model
importModel	Import a saved model
predict	Make a prediction

Upload Dataset...

 Choose File MNIST_train.csv.gz

Cancel

Upload

Upload the file without decompressing it first

Untitled Flow



CS

setupParse source_frames: ["MNIST_train.csv.gz"]

747ms

Setup Parse

PARSE CONFIGURATION

Sources MNIST_train.csv.gz

ID Key_Frame_MNIST_train.hex

Parser CSV

Separator :';'44'

Column Headers Auto

- First row contains column names
- First row contains data

Options Enable single quotes as a field quotation character Delete on done

EDIT COLUMN NAMES AND TYPES

Search by column name...

781

Numeric

0 0 0 0 0 0 0 0 0 0 0

782

Numeric

0 0 0 0 0 0 0 0 0 0 0

783

Numeric

0 0 0 0 0 0 0 0 0 0 0

784

Numeric

0 0 0 0 0 0 0 0 0 0 0

785

Numeric

2 3 0 0 2 7 5 2 6 8

Change the data type of “label” from “Numeric” to “Enum” (categorical)

- Numeric
- Numeric
- Numeric
- Numeric
- Numeric
- Unknown
- Numeric
- Enum
- Time
- UUID
- String
- Invalid

Enum

Time

UUID

String

Invalid

Parse

Ready

Connections: 0 H2O



Untitled Flow



CS getFrameSummary "Key_Frame__MNIST_train.hex"

268ms

Key_Frame__MNIST_train.hexActions: [View Data](#) [Split...](#) [Build Model...](#) [Predict](#) [Download](#) [Export](#)[Delete](#)**Note: Size in Memory**Rows
60000
Columns
785
Compressed Size
21MB

▼ COLUMN SUMMARIES

label	type	Missing	Zeros	+Inf	-Inf	min	max	mean	sigma	cardinality	Actions
C781	int	0	60000	0	0	0	0	0	0	0	Convert to enum
C782	int	0	60000	0	0	0	0	0	0	0	Convert to enum
C783	int	0	60000	0	0	0	0	0	0	0	Convert to enum
C784	int	0	60000	0	0	0	0	0	0	0	Convert to enum
C785	enum	0	5923	0	0	0	9.0	.	.	10	Convert to numeric

[Previous 20 Columns](#) [Next 20 Columns](#)**Click on individual labels to explore data**[CHUNK COMPRESSION SUMMARY](#)[FRAME DISTRIBUTION SUMMARY](#)

● Ready

Connections: 0

**H₂O**

H2O Flow x jo-fai

localhost:54321/flow/index.html#

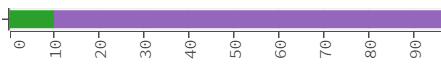
H2O FLOW Flow ▾ Cell ▾ Data ▾ Model ▾ Score ▾ Admin ▾ Help ▾

Untitled Flow

Actions Impute Inspect

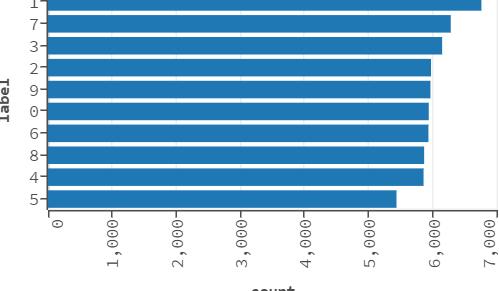
CHARACTERISTICS

All



Category	avg(percent)
All	~10

DOMAIN (MAX 1000 LEVELS)



label	count
1	~7000
7	~7000
3	~7000
2	~7000
9	~7000
0	~7000
6	~7000
8	~7000
4	~7000
5	~5000

Ready Connections: 0 H2O

H₂O FLOW

Flow ▾ Cell ▾ Data ▾ Model ▾ Score ▾ Admin ▾ Help ▾

Untitled Flow



splitFrame

21ms

Split Frame

Frame: Key_Frame__MNIST_train.hex ▾

Splits: Ratio

0.8

Key

train



0.20

valid

Add a new split

Seed: 1234

Split the full dataset into training (80% = 48k records) and validation (20% = 12k) – a common machine learning practice

>Create

splitFrame "Key_Frame__MNIST_train.hex", [0.8], ["train","valid"], 1234

815ms

Split Frames

Type Key

train

valid



?

Ratio
0.8
0.19999999999999996

Ready

Connections: 0

H₂O

H2O Flow x Jo-fai

localhost:54321/flow/index.html#

Apps Bookmarks hE Rs TP Py HM J G B JM JP S P gS k T TL H SF F H2OF T L D B DO E@ Y@ iL hL hConf hEC SL Hd kH Other bookmarks

H₂O FLOW Flow Cell Data Model Score Admin Help

Untitled Flow

Expression... buildModel "deeplearning"

104ms

Build a Model

Select an algorithm: Deep Learning

PARAMETERS

model_id deeplearning-flow Destination id for this model; auto-generated if not specified.

training_frame train Id of the training data frame (Not required, to allow initial validation of model parameters).

validation_frame valid Id of the validation data frame.

nfold 0 Number of folds for N-fold cross-validation (0 to disable or >= 2).

response_column (Choose...) Response variable column.

ignored_columns C766, C767, C768, C769, C770, C771, C772, C773, C774, C775, C776, C777, C778, C779, C780, C781, C782, C783, C784, C785 All None

Click and select parameters for model training

GRID?

Only show columns with more than 0 % missing values.

Previous 100 Next 100

Connections: 0 H₂O

H2O Flow x

localhost:54321/flow/index.html#

H2O FLOW Flow Cell Data Model Score Admin Help

Untitled Flow

Only show columns with more than 0 % missing values.

ignore_const_cols Ignore constant columns.

activation RectifierWithDropout Activation function.

hidden 200, 200 Hidden layer sizes (e.g. [100, 100]).

epochs 20 How many times the dataset should be iterated (streamed), can be fractional.

variable_importances Compute variable importances for input features (Gedeon method) - can be slow for large networks.

ADVANCED

fold_column (Choose...) Column with cross-validation fold index assignment per observation.

score_each_iteration Whether to score during each iteration of model training.

weights_column (Choose...) Column with observation weights. Giving some observation a weight of zero is equivalent to excluding it from the dataset; giving an observation a relative weight of 2 is equivalent to repeating that row twice. Negative weights are not allowed.

offset_column (Choose...) Offset column. This will be added to the combination of columns before applying the link function.

balance_classes Balance training data class counts via over/Under-sampling (for imbalanced data).

max_confusion_matrix_size 20 Maximum size (# classes) for confusion matrices to be printed in the Logs.

max_hit_ratio_k 0 Max. number (top K) of predictions to use for hit ratio computation (for multi-class only, 0 to disable).

checkpoint Model checkpoint to resume training with.

use_all_factor_levels Use all factor levels of categorical variables. Otherwise, the first factor level is omitted (without loss of accuracy). Useful for variable importances and auto-enabled for autoencoder.

standardize If enabled, automatically standardize the data. If disabled, the user must provide properly scaled input data.

train_samples_per_iteration -2 Number of training samples (globally) per MapReduce iteration. Special values are 0: one epoch, -1: all available data (e.g., replicated training data), -2: automatic.

adaptive_rate Adaptive learning rate.

input_dropout_ratio 0 Input layer dropout ratio (can improve generalization, try 0.1 or 0.2).

hidden_dropout_ratios I1 0 Hidden layer dropout ratios (can improve generalization), specify one value per hidden layer, defaults to 0.5.

I2 0 L1 regularization (can add stability and improve generalization, causes many weights to become 0).

loss Automatic Loss function.

distribution AUTO Distribution function

huber_alpha 0.9 Desired quantile for Huber/M-regression (threshold between quadratic and linear loss, must be between 0 and 1).

score_interval 5 Shortest time interval (in seconds) between model scoring.

Users have full access to all available parameters – fine-tune model training process

For example, I am using **rectifier with dropout** as the activation to train the model for **20 epochs** with **classes balancing**. Leaving other settings as default.

Ready

Connections: 0 H2O

39

H2O FLOW

Flow ▾ Cell ▾ Data ▾ Model ▾ Score ▾ Admin ▾ Help ▾

Untitled Flow



max_categorical_features 2147483647

Max. number of categorical features, enforced via hashing. #Experimental

reproducible

Force reproducibility on small data (will be slow - only uses 1 thread).

export_weights_and_biases

Whether to export Neural Network weights and biases to H2O Frames.

mini_batch_size 1

Mini-batch size (smaller leads to better fit, larger can speed up and generalize better).

elastic_averaging

Elastic averaging between compute nodes can improve distributed model convergence. #Experimental

Build Model

CS

```
buildModel 'deeplearning', {"model_id": "deeplearning-flow", "training_frame": "train", "validation_frame": "valid", "nfolds": 0, "response_column": "C785", "ignored_columns": [], "ignore_const_cols": true, "activation": "RectifierWithDropout", "hidden": [200, 200], "epochs": 20, "variable_importances": false, "score_each_iteration": false, "balance_classes": true, "max_confusion_matrix_size": 20, "max_hit_ratio_k": 0, "checkpoint": "", "use_all_factor_levels": true, "standardize": true, "train_samples_per_iteration": -2, "adaptive_rate": true, "input_dropout_ratio": 0, "hidden_dropout_ratios": [], "l1": 0, "l2": 0, "loss": "Automatic", "distribution": "AUTO", "huber_alpha": 0.9, "score_interval": 5, "score_training_samples": 10000, "score_validation_samples": 0, "score_duty_cycle": 0.1, "stopping_rounds": 5, "stopping_metric": "AUTO", "stopping_tolerance": 0, "max_runtime_secs": 0, "autoencoder": false, "categorical_encoding": "AUTO", "class_sampling_factors": [], "max_after_balance_size": 5, "pretrained_autoencoder": "", "overwrite_with_best_model": true, "target_ratio_comm_to_comp": 0.05, "seed": -1, "rho": 0.99, "epsilon": 1e-8, "nesterov_accelerated_gradient": true, "max_w2": "Infinity", "initial_weight_distribution": "UniformAdaptive", "classification_stop": 0, "score_validation_sampling": "Uniform", "diagnostics": true, "fast_mode": true, "force_load_balance": true, "single_node_mode": false, "shuffle_training_data": false, "missing_values_handling": "MeanImputation", "quiet_mode": false, "sparse": false, "col_major": false, "average_activation": 0, "sparsity_beta": 0, "max_categorical_features": 2147483647, "reproducible": false, "export_weights_and_biases": false, "mini_batch_size": 1, "elastic_averaging": false}
```

Started at 10:40:35 am

Job

Run Time 00:01:04.377

Remaining Time 00:02:45.486

Type Model

Key Q deeplearning-flow

Description DeepLearning

Status RUNNING

Progress 29%

Iterations: 5. Epochs: 5.00000. Speed: 5.553 samples/sec. Estimated time left: 3 min 10.601 sec

Actions View Cancel Job

Training the model with estimated remaining time
– users can stop the process early if they want to



Ready

Connections: 0 H2O



Untitled Flow



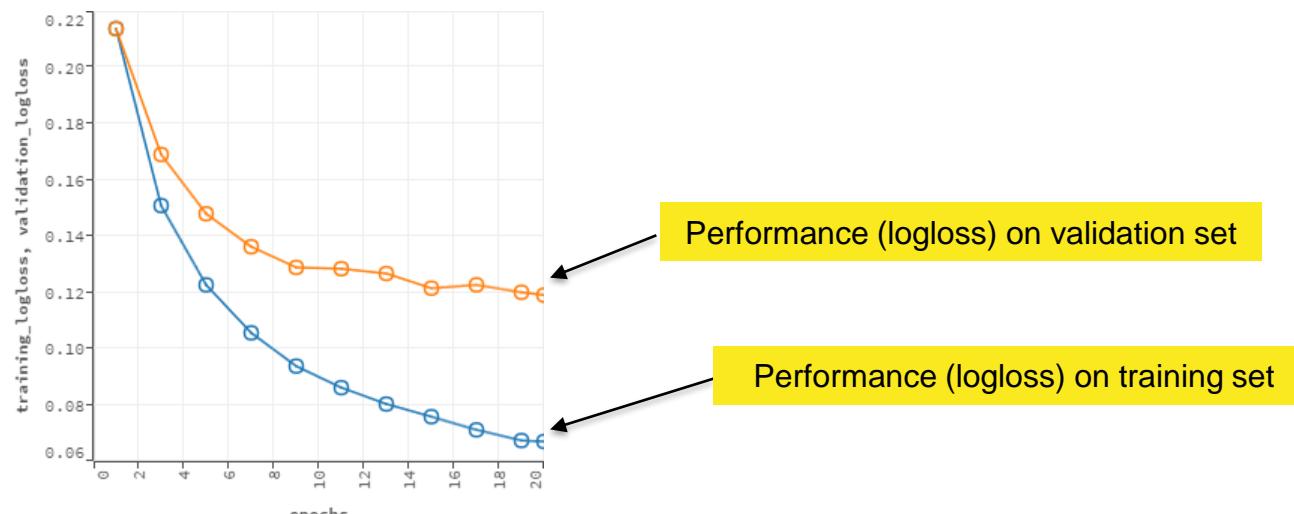
Model ID: deeplearning-flow

Algorithm: Deep Learning

Actions: Refresh Predict... Download POJO Export Inspect Delete

► MODEL PARAMETERS

▼ SCORING HISTORY - LOGLOSS



Performance (logloss) on validation set

Performance (logloss) on training set

H₂O FLOW

Flow ▾ Cell ▾ Data ▾ Model ▾ Score ▾ Admin ▾ Help ▾

Untitled Flow



▼ TRAINING METRICS - CONFUSION MATRIX VERTICAL: ACTUAL; ACROSS: PREDICTED

	0	1	2	3	4	5	6	7	8	9	Error	Rate
0	981	0	0	0	0	5	3	2	0	0	0.0101	10 / 991
1	0	966	4	2	0	0	4	2	1	0	0.0133	13 / 979
2	4	0	997	0	3	0	11	7	0	0	0.0245	25 / 1,022
3	0	0	10	928	0	4	7	12	3	2	0.0393	38 / 966
4	0	0	3	0	955	0	6	2	1	2	0.0144	14 / 969
5	0	0	2	1	0	919	19	1	1	2	0.0275	26 / 945
6	2	0	0	0	2	0	996	0	2	0	0.0068	6 / 1,002
7	0	1	2	1	0	1	6	1033	1	1	0.0124	13 / 1,046
8	1	4	6	4	0	8	9	3	984	3	0.0372	38 / 1,022
9	2	0	2	1	2	2	5	17	0	963	0.0312	▲ 994
Total	990	971	1026	937	962	934	1068	1080	995	973	0.0215	214 / 9,936

Confusion Matrix on Training Set (48k Records)
About 2% Error

▼ VALIDATION METRICS - CONFUSION MATRIX VERTICAL: ACTUAL; ACROSS: PREDICTED

	0	1	2	3	4	5	6	7	8	9	Error	Rate	
0	1175	0	5	4	0	2	14	2	2	0	0.0241	29 / 1,204	
1	0	1320	9	5	2	1	5	7	2	0	0.0229	31 / 1,351	
2	3	0	0	1107	5	1	1	26	9	4	1	0.0432	50 / 1,157
3	2	2	17	1179	0	25	8	17	8	4	0.0658	83 / 1,262	
4	2	3	5	0	1092	0	13	5	3	19	0.0438	50 / 1,142	
5	3	0	1	8	1	1073	17	3	3	4	0.0359	40 / 1,113	
6	7	0	3	0	2	3	1139	0	3	0	0.0156	18 / 1,157	
7	2	4	5	3	1	0	6	1247	1	5	0.0212	27 / 1,274	
8	1	9	8	11	3	10	7	6	1092	8	0.0545	63 / 1,155	
9	0	1	2	5	16	5	4	41	6	1096	0.0680	▲ 80 / 1,176	
Total	1195	1339	1162	1220	1118	1120	1239	1337	1124	1137	0.0393	471 / 11,991	

Confusion Matrix on Validation Set (12k Records)
About 4% Error



H2O FLOW

Flow ▾ Cell ▾ Data ▾ Model ▾ Score ▾ Admin ▾ Help ▾

Untitled Flow

**Predict**

Name: prediction-17ab8810-25d7-4bc4-9213-0ffec24b99ca

Model: deeplearning-flow ▾

Frame: Key_Frame__MNIST_test.hex ▾

Actions: Predict

Using the model for prediction on test set

CS

```
predict model: "deeplearning-flow", frame: "Key_Frame__MNIST_test.hex", predictions_frame: "prediction-17ab8810-25d7-4bc4-9213-0ffec24b99ca"
```

1.5s

Prediction

Actions: Inspect

PREDICTION

```
model deeplearning-flow
model_checksum -7748576812343879680
frame Key_Frame__MNIST_test.hex
frame_checksum 220716222683866
description .
model_category Multinomial
scoring_time 1475833596557
predictions prediction-17ab8810-25d7-4bc4-9213-0ffec24b99ca
    MSE 0.031250
    RMSE 0.176777
    nobs 10000
    r2 0.996273
    logloss 0.122813
    mean_per_class_error 0.036786
```

Combine predictions with frame

Ready

Connections: 0



Untitled Flow



▼ PREDICTION - CM - CONFUSION MATRIX

0	1	2	3	4	5	6	7	8	9	Error	Rate
962	0	1	2	0	2	10	1	1	1	0.0184	18 / 980
0	1121	6	1	0	0	3	1	3	0	0.0123	14 / 1,135
6	0	988	2	3	1	15	14	3	0	0.0426	44 / 1,032
0	0	10	971	0	6	7	11	3	2	0.0386	39 / 1,010
0	0	6	0	944	1	12	4	3	12	0.0387	38 / 982
3	0	2	9	2	849	18	1	5	3	0.0482	43 / 892
5	3	1	0	3	5	936	0	5	0	0.0230	22 / 958
1	4	14	1	1	0	2	1000	2	3	0.0272	28 / 1,028
5	1	4	6	7	9	9	7	922	4	0.0534	52 / 974
2	4	0	7	12	3	7	28	3	943	0.0654	66 / 1,009
984	1133	1032	999	972	876	1019	1067	950	968	0.0364	364 / 10,000

Confusion Matrix on Test Set (10k Records)
About 4% Error (similar to validation)



Untitled Flow



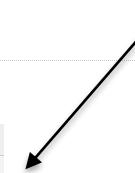
prediction-17ab8810-25d7-4bc4-9213-0ffec24b99ca

DATA

◀ Previous 20 Columns ▶ Next 20 Columns

Row	predict	p ₀	p ₁	p ₂	p ₃	p ₄	p ₅	p ₆	p ₇	p ₈	p ₉
1	8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
2	3	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
3	8	0.1102	0.0	0.0153	0.0039	0.0089	0.0461	0.1479	0.0	0.6660	0.0016
4	0	0.9990	0.0	0.0	0.0	0.0	0.0	0.0010	0.0	0.0	0.0
5	1	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	5	0.0	0.0	0.0	0.0	0.0	0.9594	0.0	0.0	0.0406	0.0
7	0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	1	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	5	0.0	0.0	0.0	0.0092	0.0	0.9873	0.0	0.0	0.0002	0.0033
10	2	0.0	0.0003	0.9851	0.0011	0.0	0.0	0.0	0.0011	0.0123	0.0
11	4	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
12	8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
13	1	0.0	0.9999	0.0	0.0	0.0	0.0	0.0	0.0	0.0001	0.0
14	4	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
15	3	0.0	0.0	0.0001	0.9983	0.0	0.0	0.0	0.0016	0.0	0.0
16	9	0.0	0.0	0.0	0.0002	0.0	0.0	0.0	0.0	0.9997	0.0
17	6	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0

Full prediction outputs including individual probabilities and predicted label



H2O Deep Learning beats MNIST

```
> install.packages("h2o")
> library(h2o)
> h2oServer <- h2o.init(ip="mr-0xd1", port=53322)
> train_hex <- h2o.importFile(h2oServer, "./train.csv.gz")
> test_hex <- h2o.importFile(h2oServer, "./test.csv.gz")
> record_model <- h2o.deeplearning(x = c(1:784), y = 785, data = train_hex, validation = test_hex,
  activation = "RectifierWithDropout", hidden = c(1024,1024,2048),
  epochs = 2000, l1 = 1e-5, input_dropout_ratio = 0.2,
  train_samples_per_iteration = -1, classification_stop = -1)
```

```
|-----| 100%
> record_model@model$confusion
```

Predicted											
Actual	0	1	2	3	4	5	6	7	8	9	Error
0	975	1	1	0	0	0	1	1	1	0	0.00510
1	0	1135	0	0	0	0	0	0	0	0	0.00000
2	0	1	1028	0	1	0	0	2	0	0	0.00388
3	0	0	1	1001	0	1	0	3	2	2	0.00891
4	0	0	2	0	971	0	3	1	0	5	0.01120
5	2	0	0	6	0	881	1	1	1	0	0.01233
6	2	3	0	1	1	1	950	0	0	0	0.00835
7	1	2	6	0	0	0	0	1017	0	2	0.01070
8	1	0	3	4	0	1	1	2	958	3	0.01540
9	1	2	0	1	3	3	0	3	0	988	0.01288
Totals	982	1145	1041	1013	976	887	956	1030	961	997	0.00870

Standard 60k/10k data

No distortions

No convolutions

No unsupervised training

No ensemble

4 hours on 4 16-core nodes

World-class result!

0.87% test set error