Scalable Machine Learning in R and Python with H2O

R-Ladies & PyLadies Meetup

London, UK Dec 2016

Erin LeDell Ph.D.
Machine Learning Scientist
H2O.ai

Introduction

- Statistician & Machine Learning Scientist at H2O.ai, in Mountain View, California, USA
- Ph.D. in Biostatistics with Designated Emphasis in Computational Science and Engineering from UC Berkeley (focus on Machine Learning)
- Worked as a data scientist at several startups

- Co-organizer of R-Ladies San Francisco and co-founder of R-Ladies Global (rladies.org)
- Founder of "Women in Machine Learning & Data Science" (wimlds.org) Meetup organization

H2O.ai

Agenda

- Who/What is H2O?
- Machine Learning 101
- H2O Machine Learning Platform
- H2O in R & Python
- Tutorials: Intro, Grid, DL, Stacking
- New/active developments in H2O

H2O.ai

H2O.ai, the Company

- Founded in 2012
- Stanford & Purdue Math & Systems Engineers
- Headquarters: Mountain View, California, USA

H2O, the Platform

- Open Source Software (Apache 2.0 Licensed)
- R, Python, Scala, Java and Web Interfaces
- Distributed Algorithms that Scale to Big Data

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*

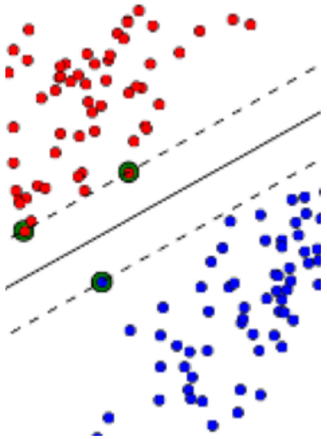
Dr. Robert Tibshirani

- 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*

What is Machine Learning?



"Field of study that gives computers the ability to learn without being explicitly programmed."

— Arthur Samuel, 1959

Unlike rules-based systems which require a human expert to hard-code domain knowledge directly into the system, a machine learning algorithm learns how to make decisions from the data alone.

Machine Learning Tasks

Regression

- Predict a real-valued response (e.g. price, weight)
- Response distribution: Gaussian, Gamma, Poisson, etc.
- Evaluate with MSE, MAE or R^2

Classification

- Multi-class or binary classification
- Ranking (e.g. Google Search results order)
- Evaluate with Classification Error or AUC

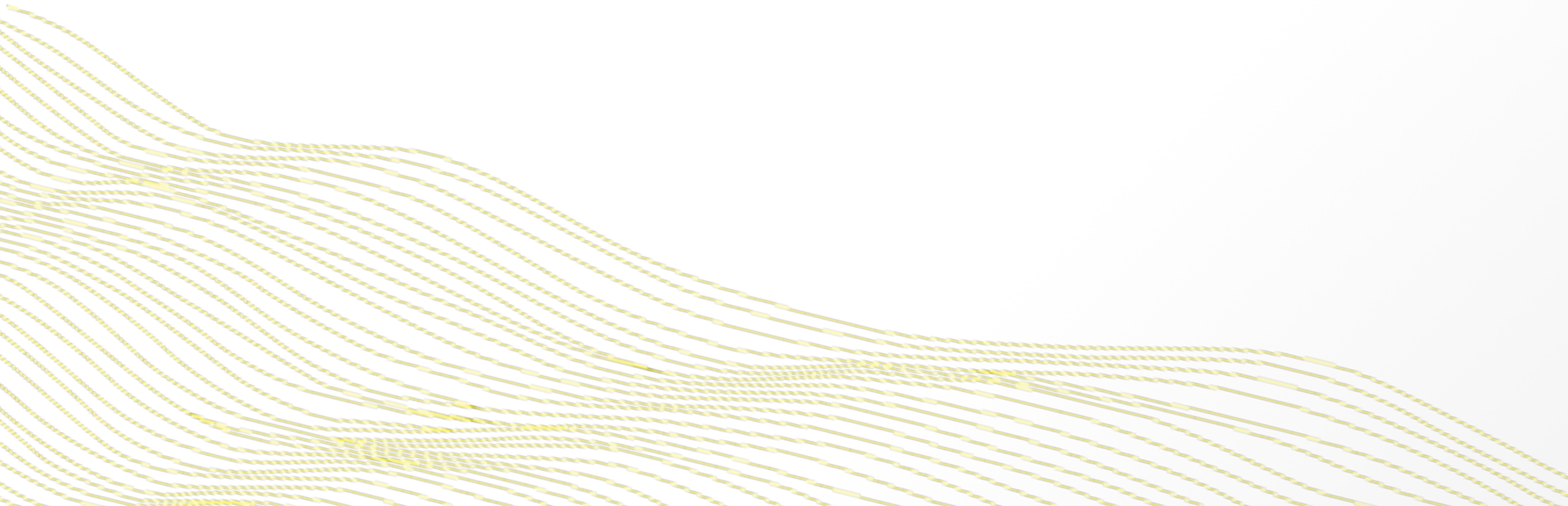
Clustering

- Unsupervised learning (no training labels)
- Partition the data; identify clusters or sub-populations
- Evaluate with AIC, BIC or Total Sum of Squares

Train, Validation and Test Set

- If you plan on doing any model tuning, you should split your dataset into three parts: Train, Validation and Test
- There is no general rule for how you should partition the data and it will depend on how strong the signal in your data is, but an example could be:
  50% Train, 25% Validation and 25% Test
- The validation set is used strictly for model tuning and the test set is used to make a final estimate of the generalization error.

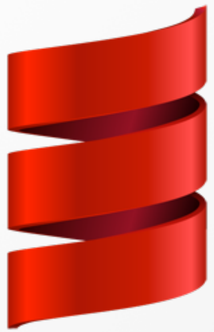
H2O Platform



H2O Platform Overview

- Distributed implementations of cutting edge ML algorithms.
- Core algorithms written in high performance Java.
- APIs available in R, Python, Scala, REST/JSON.
- Interactive Web GUI called H2O Flow.
- Easily deploy models to production with H2O Steam.



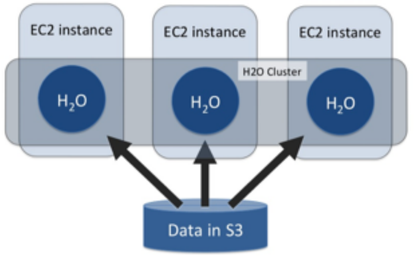




H2O Platform Overview

- Write code in high-level language like R (or use the web GUI) and output production-ready models in Java.
- To scale, just add nodes to your H2O cluster.
- Works with Hadoop, Spark and your laptop.

H2O Distributed Computing

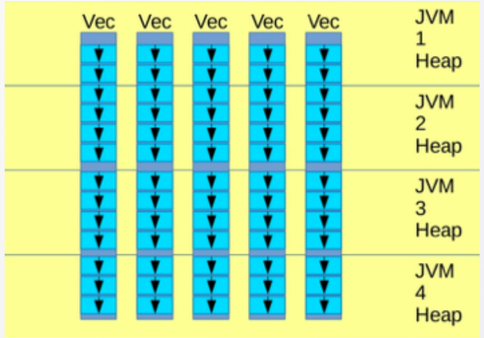
H2O Cluster



- Multi-node cluster with shared memory model.
- All computations in memory.
- Each node sees only some rows of the data.
- No limit on cluster size.

H2O Frame

- Distributed data frames (collection of vectors).
- Columns are distributed (across nodes) arrays.
- Works just like R's data.frame or Python Pandas DataFrame



Current Algorithm Overview

Statistical Analysis

- Linear Models (GLM)
- Naïve Bayes

Ensembles

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

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

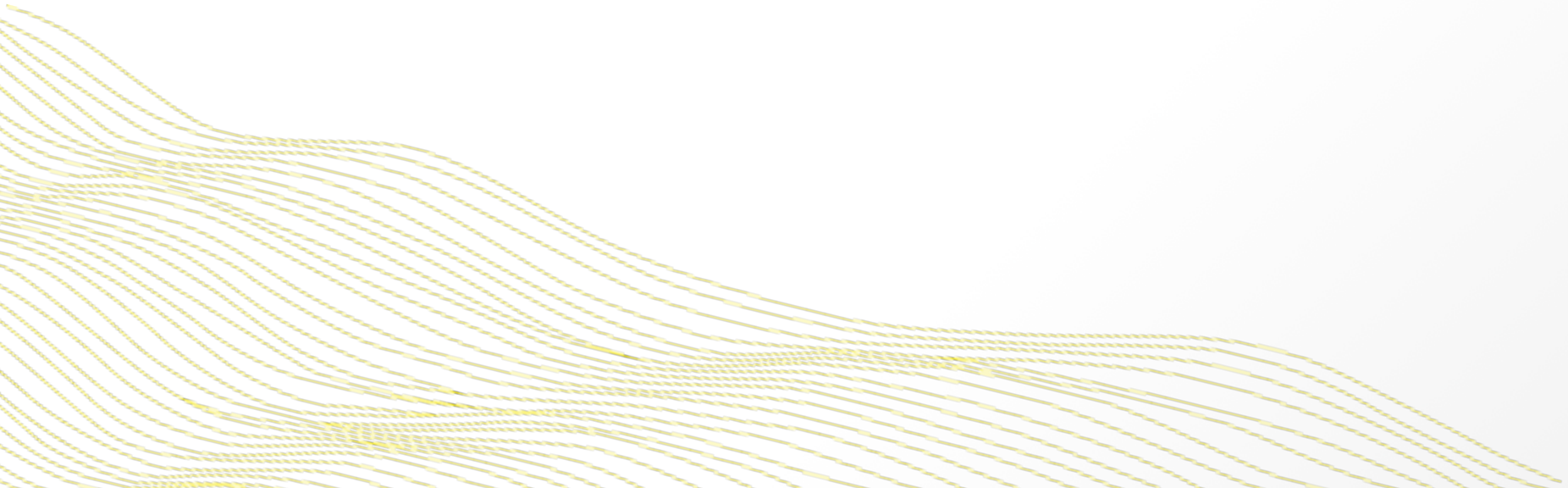
Solvers & Optimization

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

Data Munging

- Scalable Data Frames
- Sort, Slice, Log Transform

H2O in R



h2o R Package

Installation

- Java 7 or later; R 3.1 and above; Linux, Mac, Windows
- The easiest way to install the h2o R package is CRAN.
- Latest version: http://www.h2o.ai/download/h2o/r

Design

All computations are performed in highly optimized Java code in the H2O cluster, initiated by REST calls from R.

h2o Python Module

Installation

- Java 7 or later; Python 2.7, 3.5; Linux, Mac, Windows
- The easiest way to install the h2o Python module is PyPI.
- Latest version: http://www.h2o.ai/download/h2o/python

Design

All computations are performed in highly optimized Java code in the H2O cluster, initiated by REST calls from Python.

H2O R & Python Tutorials

https://github.com/h2oai/h2o-tutorials

R & Py Tutorial: Intro to H2O Algorithms

The "Intro to H2O" tutorial introduces five popular supervised machine learning algorithms in the context of a binary classification problem.

The training module demonstrates how to train models and evaluate model performance on a test set.

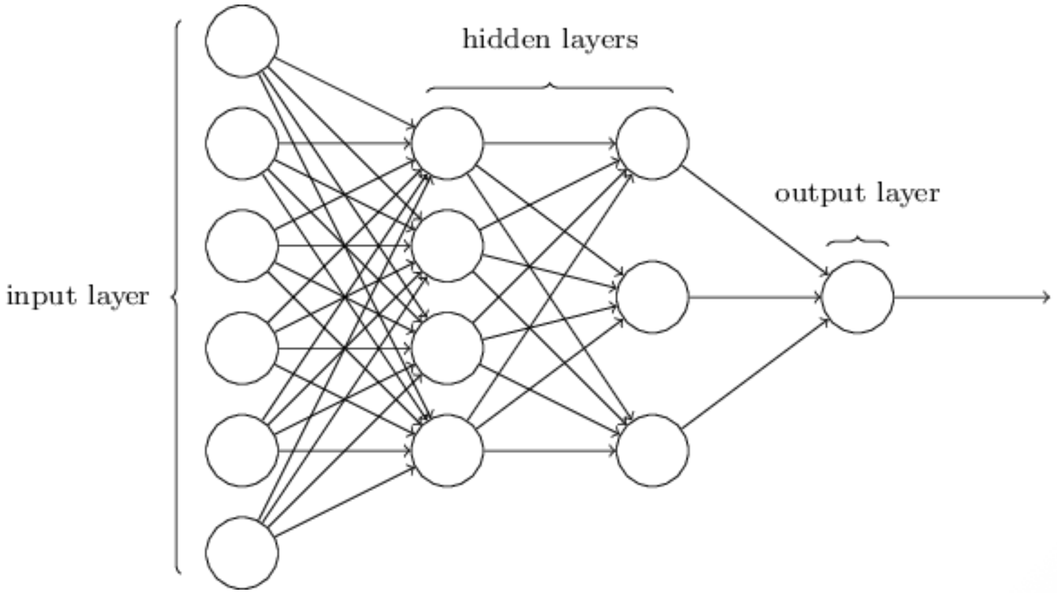
- Generalized Linear Model (GLM)
- Random Forest (RF)
- Gradient Boosting Machine (GBM)
- Deep Learning (DL)
- Naive Bayes (NB)

R & Py Tutorial: Grid Search for Model Selection

```
> print(gbm_gridperf)
H2O Grid Details
================

Grid ID: gbm_grid2
Used hyper parameters:
  -  sample_rate
  -  max_depth
  -  learn_rate
  -  col_sample_rate
Number of models: 72
Number of failed models: 0

Hyper-Parameter Search Summary: ordered by decreasing auc
  sample_rate max_depth learn_rate col_sample_rate          model_ids                auc
1           1         3       0.19                1 gbm_grid2_model_38 0.685166598389755
2         0.9         3       0.15                1 gbm_grid2_model_53 0.684956999713052
3         0.8         5       0.06                1 gbm_grid2_model_22 0.684843506375254
4         0.6         4       0.07                1  gbm_grid2_model_4 0.684327718715252
5        0.95         4       0.13                1 gbm_grid2_model_48 0.684042497773235
```

The second training module demonstrates how to find the best set of model parameters for each model using Grid Search.

R Tutorial: Deep Learning



The "Deep Learning in R" tutorial gives an overview of how to train H2O deep neural networks in R.

- Deep Learning via Multilayer Perceptrons (MLPs)
  - Early Stopping
  - Random Grid Search
- Deep Learning Autoencoders
  - Unsupervised Pretraining
  - Deep Features
  - Anomaly Detection

R Tutorial: Ensembles via Stacking

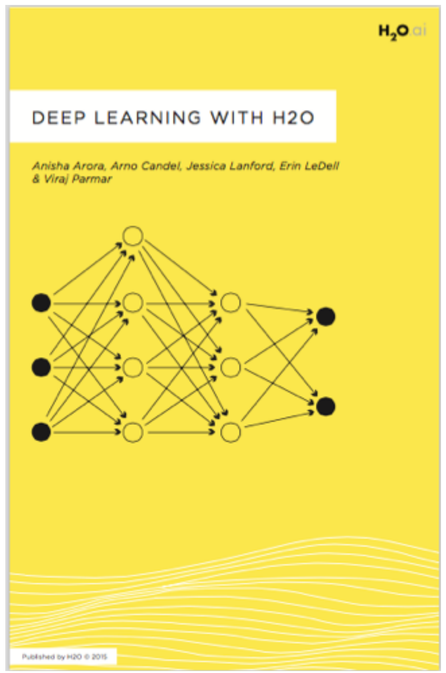
```
> perf <- h2o.ensemble_performance(fit, newdata = test)
  |======================================================================| 100%
  |======================================================================| 100%
  |======================================================================| 100%
  |======================================================================| 100%
>
> perf

Base learner performance, sorted by specified metric:
                  learner       AUC
1          h2o.glm.wrapper 0.6871177
4 h2o.deeplearning.wrapper 0.7172476
2 h2o.randomForest.wrapper 0.7653992
3          h2o.gbm.wrapper 0.7817096


H2O Ensemble Performance on <newdata>:
----------------
Family: binomial

Ensemble performance (AUC): 0.785726880397054

>
```

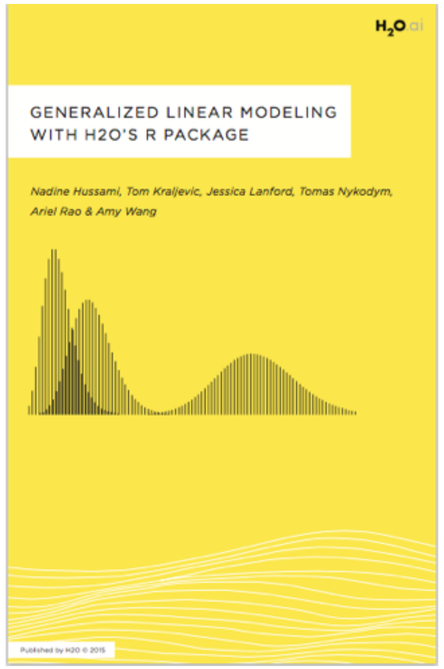
The "H2O Ensemble" Tutorial demonstrates how to use the h2oEnsemble R package which extends the h2o R API.

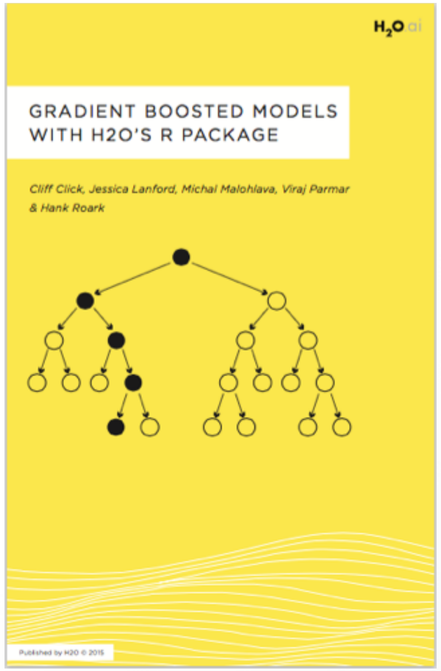
H2O Resources

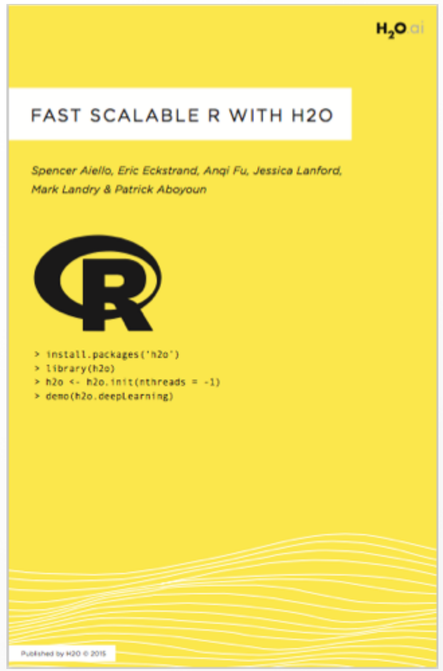
- Online Training: http://learn.h2o.ai
- Tutorials: https://github.com/h2oai/h2o-tutorials
- Slidedecks: https://github.com/h2oai/h2o-meetups
- Video Presentations: https://www.youtube.com/user/0xdata
- Events & Meetups: http://h2o.ai/events
- Community: https://community.h2o.ai

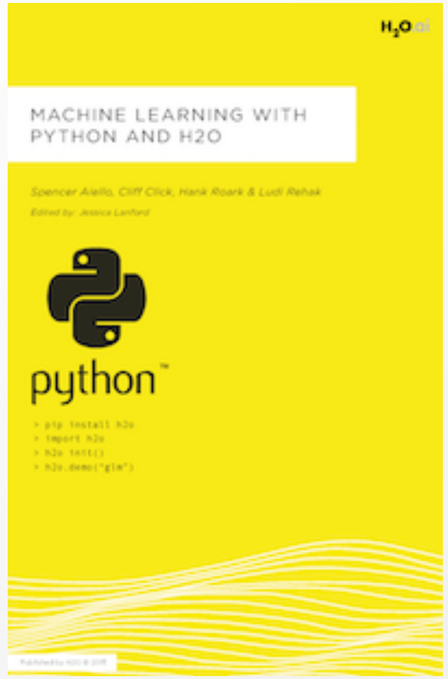
H2O Booklets











http://docs.h2o.ai/

Thank you!

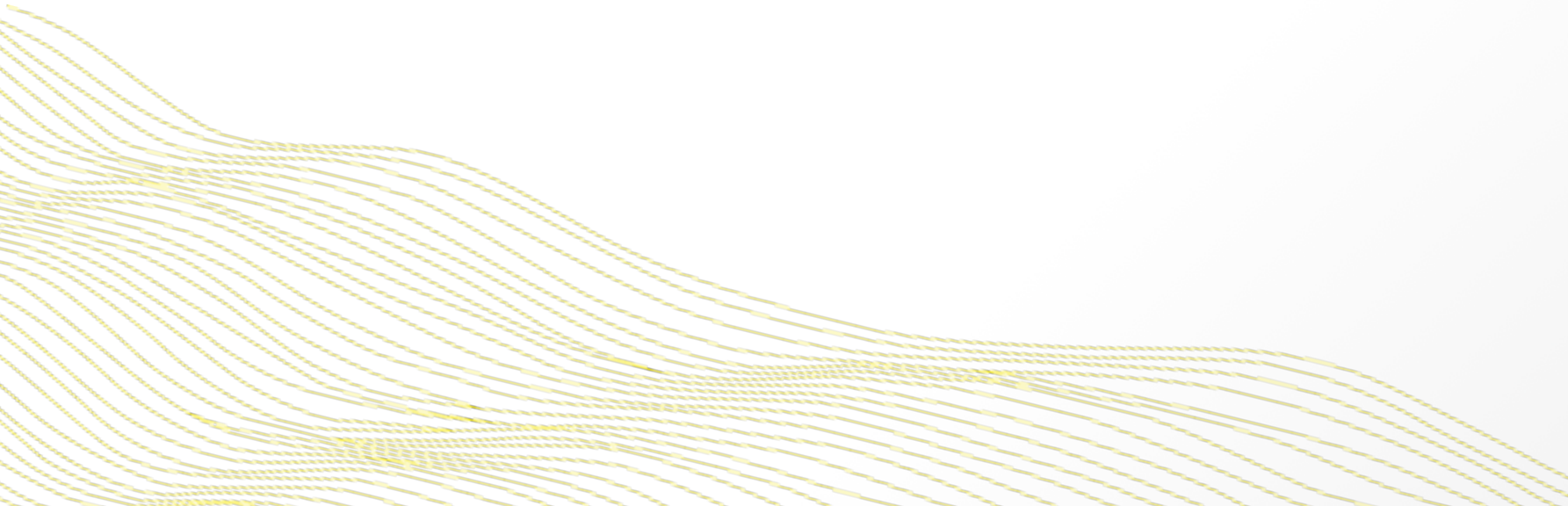
@ledell on Github, Twitter

erin@h2o.ai

http://www.stat.berkeley.edu/~ledell

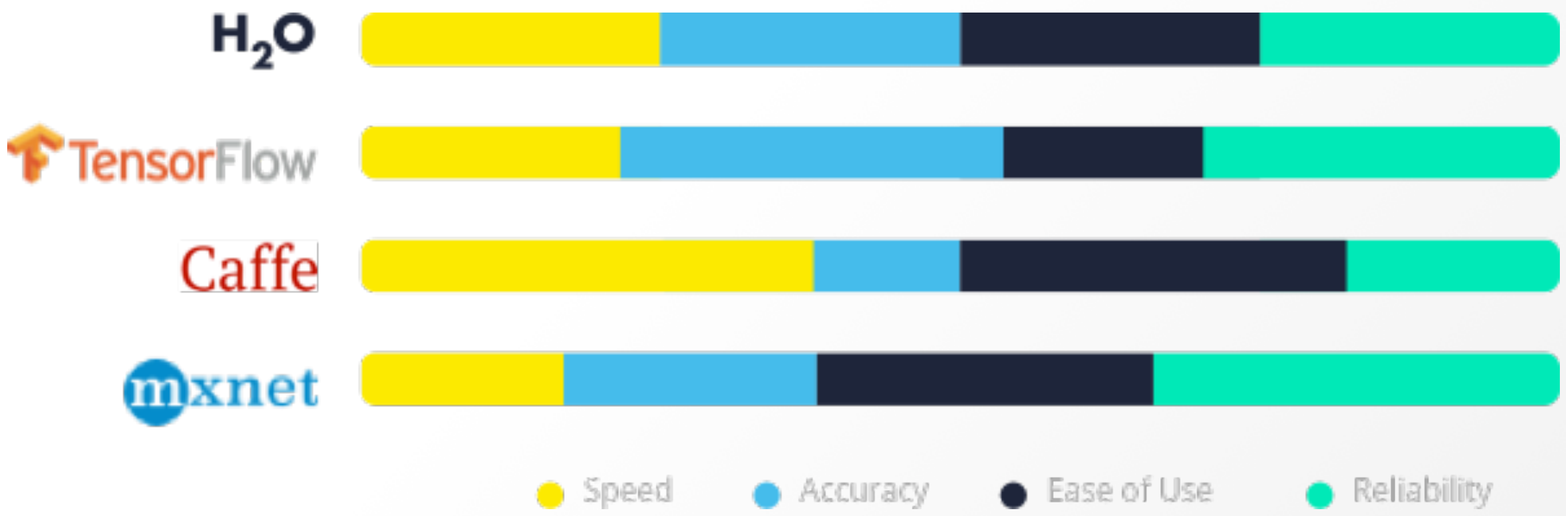
Appendix: New & Active Developments in H2O

Deep Water



Deep Water

Project “Deep Water” is a unification of the top open source libraries for Deep Learning.

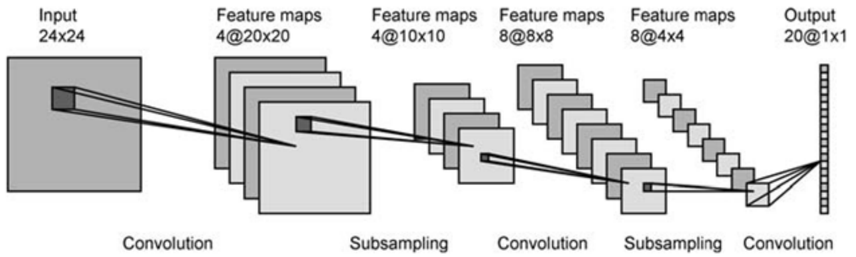


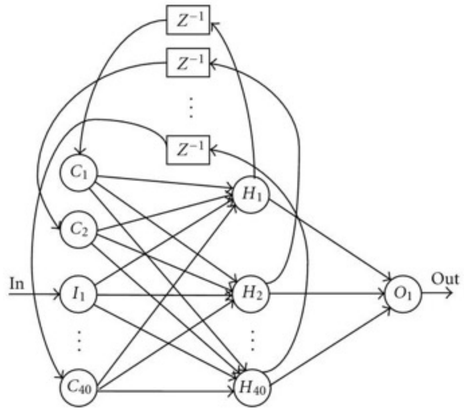
Deep Water

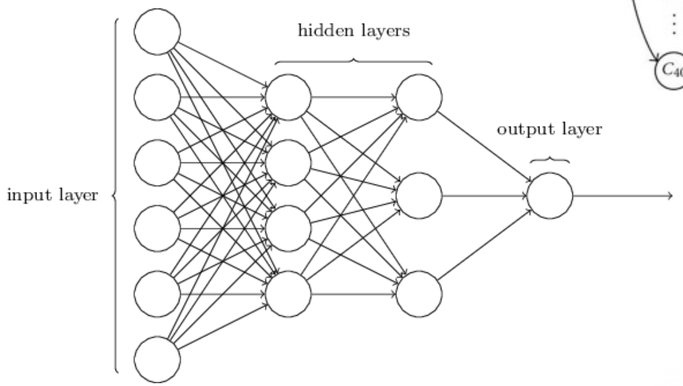
- Native implementation of Deep Learning models for GPU-optimized backends (mxnet, Caffe, TensorFlow, etc.)
- State-of-the-art Deep Learning models trained from the H2O Platform
- Provides an easy to use interface to any of the Deep Water backends.
- Extends the H2O platform to include Convolutional Neural Nets (CNNs) and Recurrent Neural Nets (RNNs) including LSTMs

https://github.com/h2oai/deepwater

Deep Water Neural Architectures





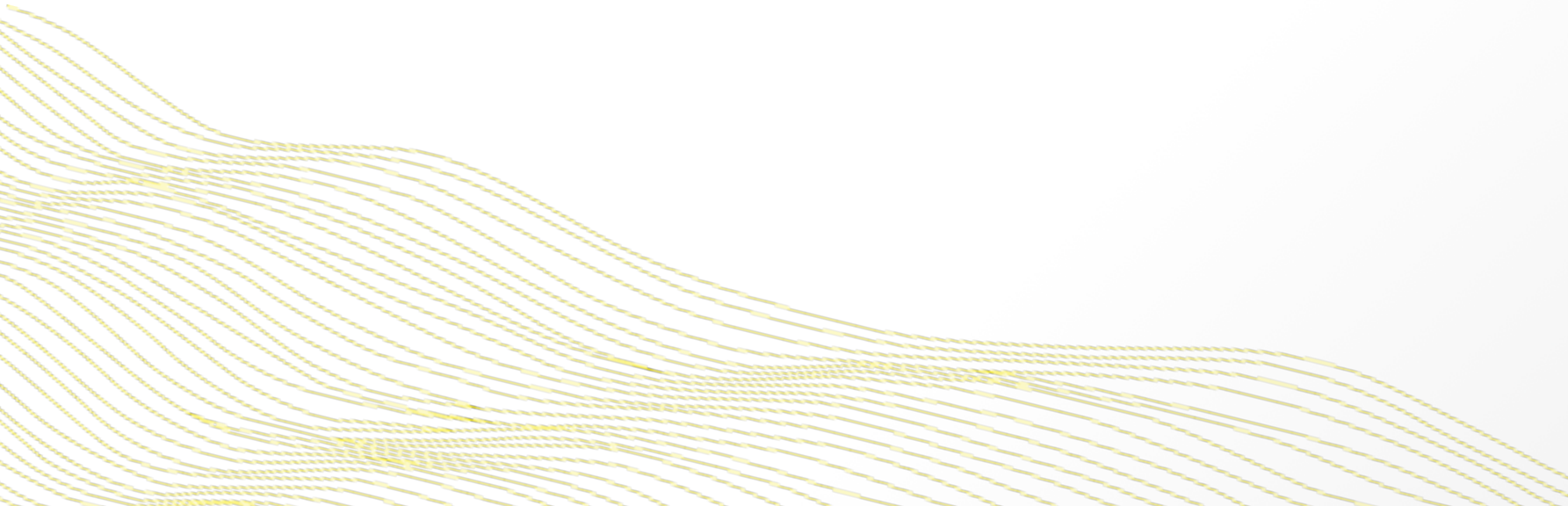


» Convolutional Neural Networks (CNNs), which are popular for image data.

» Recurrent Neural Networks (RNNs), including Long-Short-Term-Memory (LSTMs) for sequence learning including text, audio and video.

» Multilayer Perceptrons (MLPs), fully connected multilayer artificial neural networks, useful for numeric data.

RSparkling



H2O on Spark

Sparkling Water

- Sparkling Water is transparent integration of H2O into the Spark ecosystem.
- H2O runs inside the Spark Executor JVM.

Features

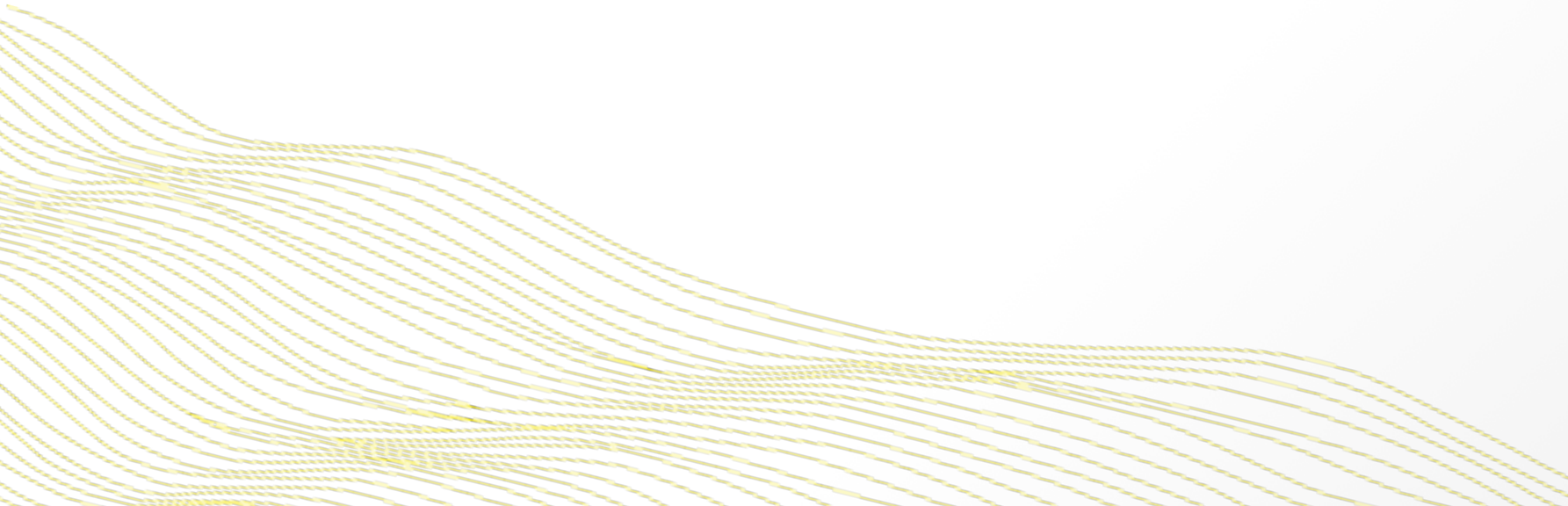
- Provides access to high performance, distributed machine learning algorithms to Spark workflows.
- Alternative to the MLlib/SparkML library in Spark.

rsparkling

- This provides an interface to H2O's machine learning algorithms on Spark, using R.
- This is an extension package for RStudio's sparklyr package that creates an R front-end for a Spark package (e.g. Sparking Water).
- This package implements only the most basic functionality (creating an H2OContext, showing the H2O Flow interface, and converting a Spark DataFrame to an H2OFrame or vice versa).

http://tinyurl.com/rsparkling-branch

AutoML



H2O 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 combination of grid search and Bayesian Optimization algorithms.
- If ensembles are permitted, then a Super Learner will be constructed.

Public code coming soon!