New Developments in H2O
Scalable Machine Learning

STATISTICAL LEARNING AND DATA MINING IV

Palo Alto, CA April 2017

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

H2O.ai

Agenda

- Who/What is H2O?
- H2O Machine Learning Platform
- New/active developments in H2O:
  - Stacked Ensembles
  - Deep Water (GPU Deep Learning)
  - RSparkling (R + Spark + H2O)
  - AutoML (Automatic Machine Learning)
- Tutorial: H2O Stacked Ensembles

H2O.ai

H2O.ai, the Company

- Team: 80; Founded in 2012
- Headquarters: Mountain View, California, USA
- Stanford & Purdue Math & Systems Engineers

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*

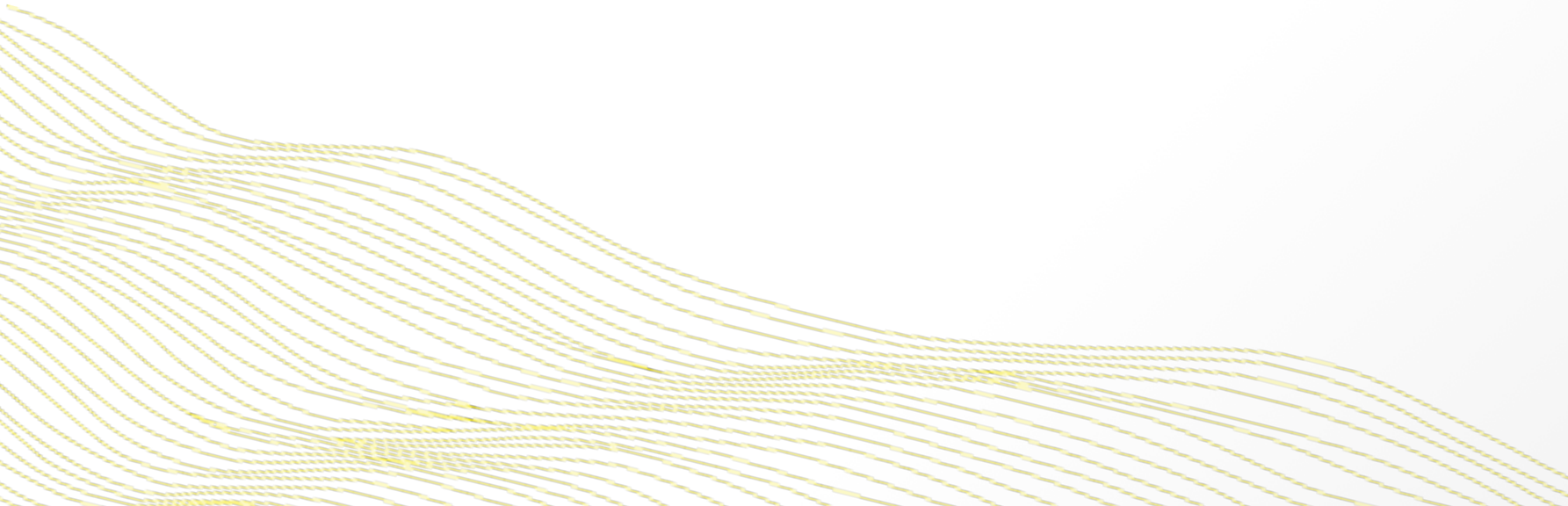
Dr. Robert Tibshirani

- 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*

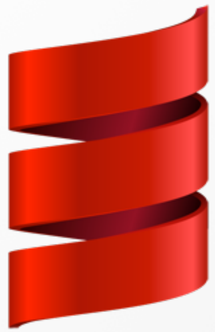
H2O Platform



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.



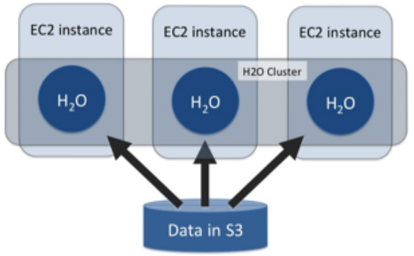




H2O Platform Overview

- 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

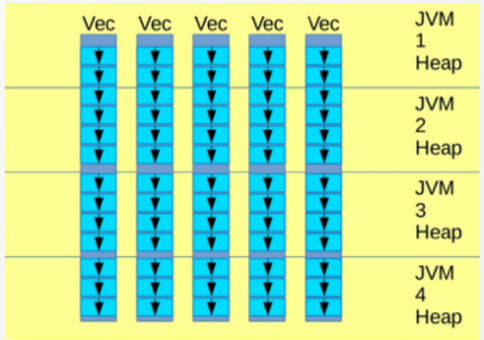
H2O Cluster



- 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
- Stacked Ensembles

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 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

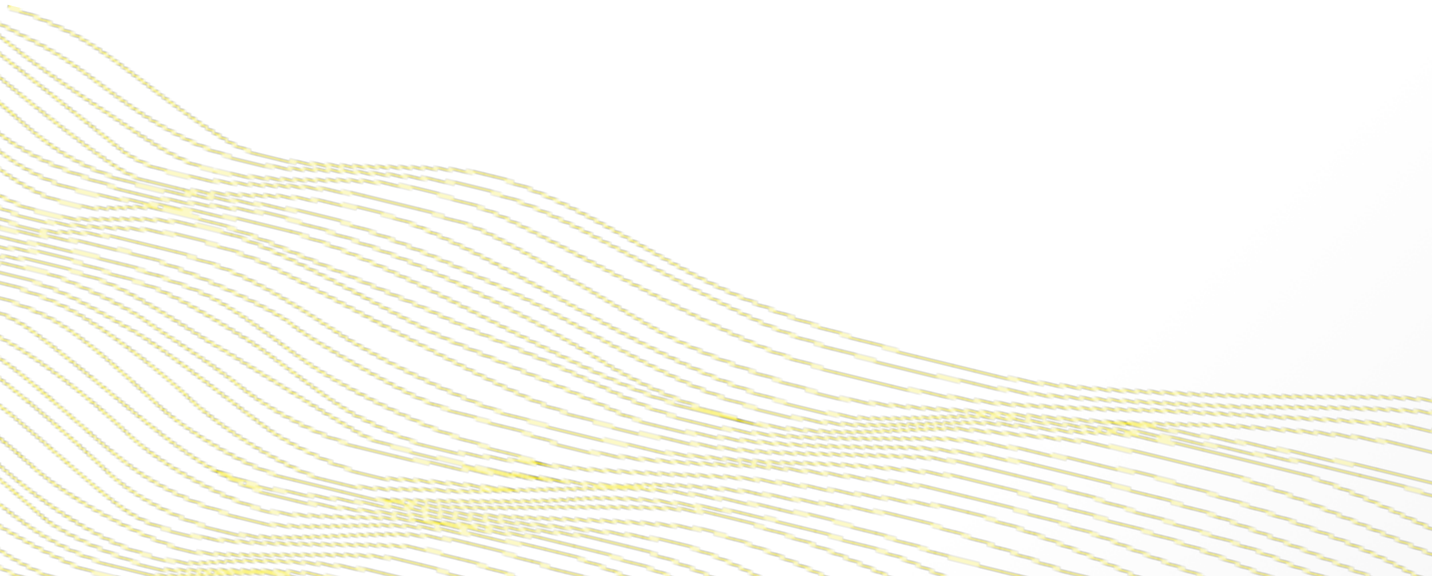
Installation

- 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

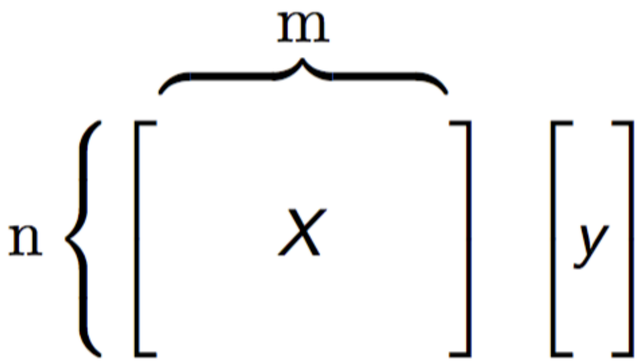
Design

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

Stacked Ensembles



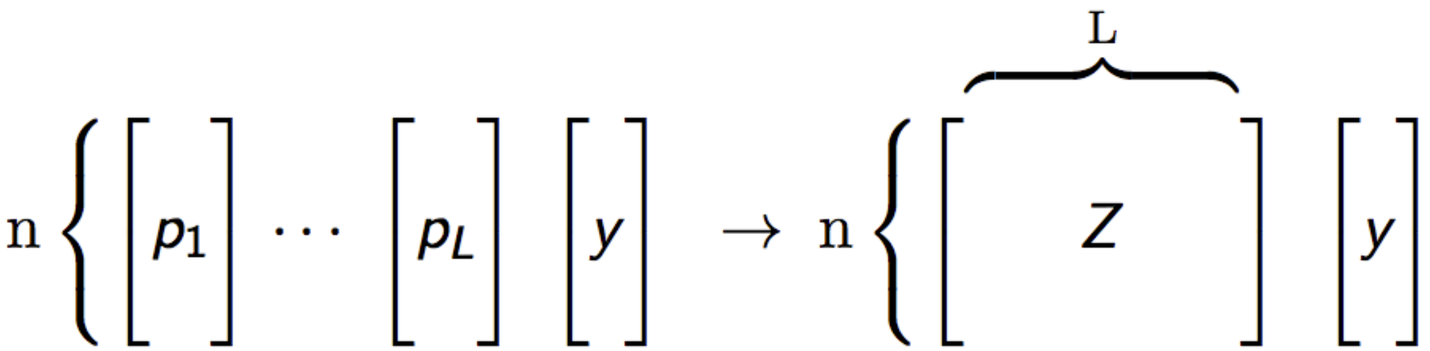
Stacking (aka Super Learner Algorithm)



"Level-zero" data

- Start with design matrix, X, and response, y
- Specify L base learners (with model params)
- Specify a metalearner (just another algorithm)
- Perform k-fold CV on each of the L learners

Stacking (aka Super Learner Algorithm)



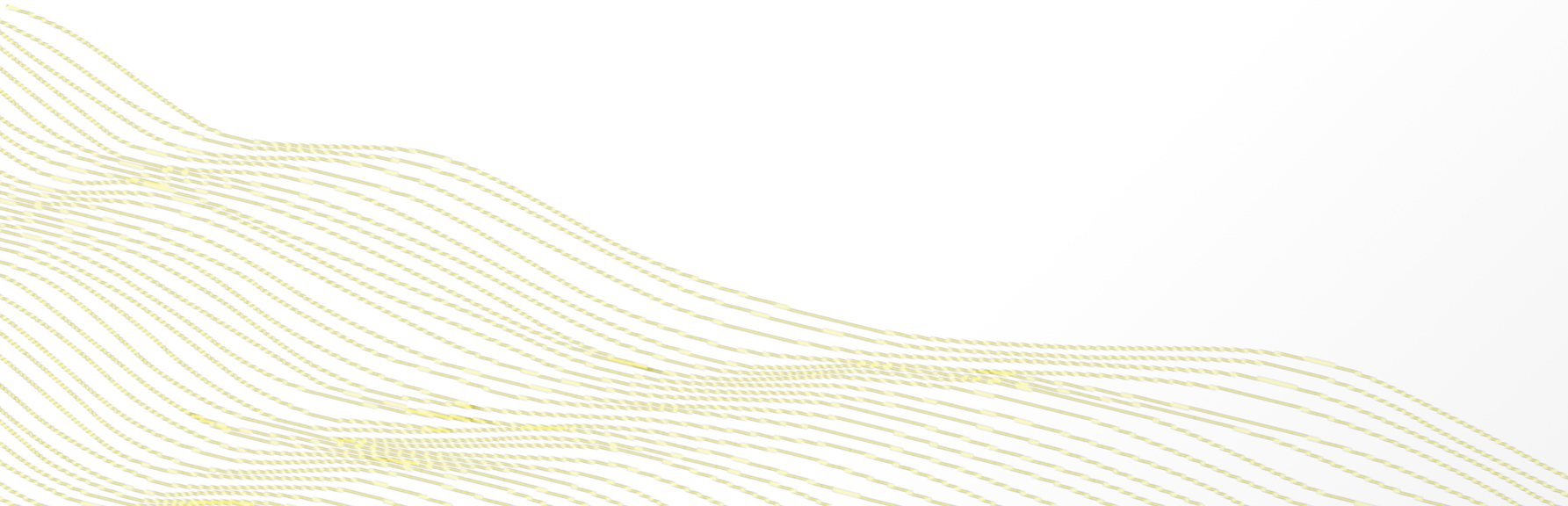
"Level-one" data

- Collect the predicted values from k-fold CV that was performed on each of the L base learners
- Column-bind these prediction vectors together to form a new design matrix, Z
- Train the metalearner using Z, y

Stacking vs. Parameter Tuning/Search

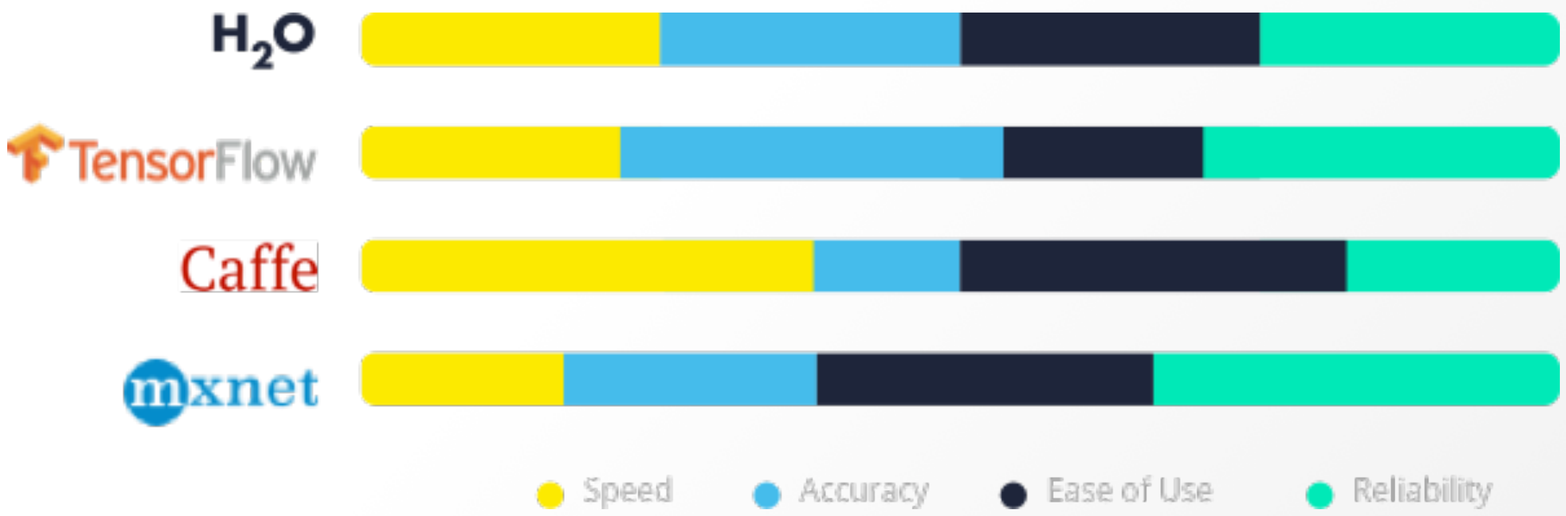
- A common task in machine learning is to perform model selection by specifying a number of models with different parameters.
- An example of this is Grid Search or Random Search.
- The first phase of the Super Learner algorithm is computationally equivalent to performing model selection via cross-validation.
- The latter phase of the Super Learner algorithm (the metalearning step) is just training another single model (no CV).
- With Stacking, your computation does not go to waste!

Deep Water



Deep Water

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

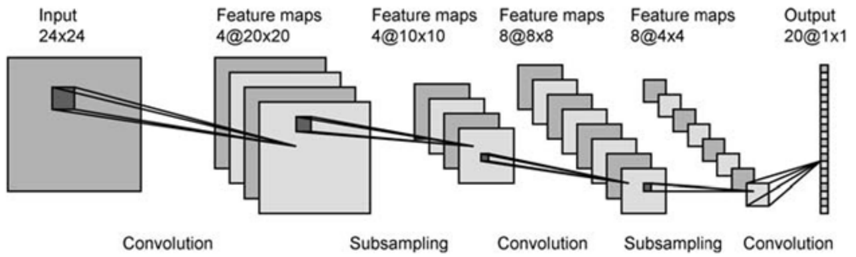


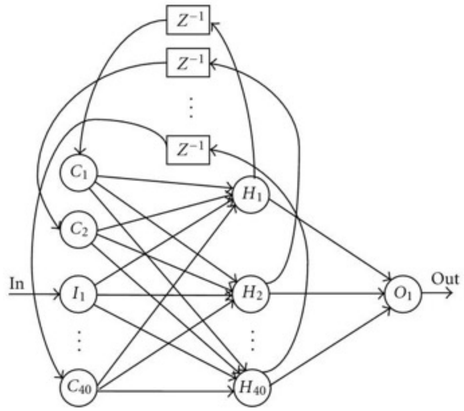
Deep Water

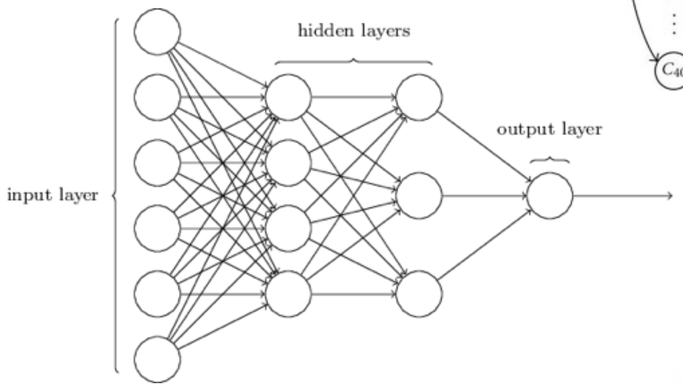
- Native implementation of Deep Learning models for GPU-optimized backends (mxnet, Caffe, TensorFlow, etc.)
- State-of-the-art Deep Learning models trained from the H2O Platform
- Provides an easy to use interface to any of the Deep Water backends.
- Extends the H2O platform to include Convolutional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

https://github.com/h2oai/deepwater

Deep Water Neural Architectures

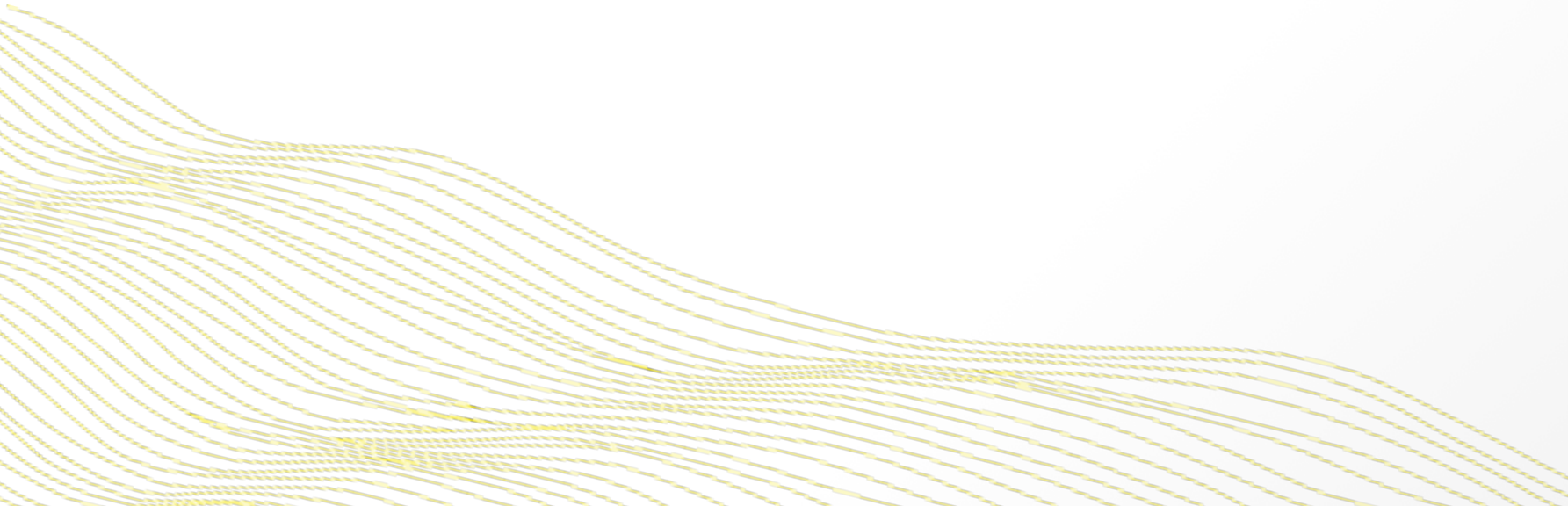






- Convolutional Neural Networks (CNNs), which are popular for image data.
- Recurrent Neural Networks (RNNs), including Long-Short-Term-Memory (LSTMs) for sequence learning including text, audio and video.
- Multilayer Perceptrons (MLPs), fully connected multilayer artificial neural networks, useful 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

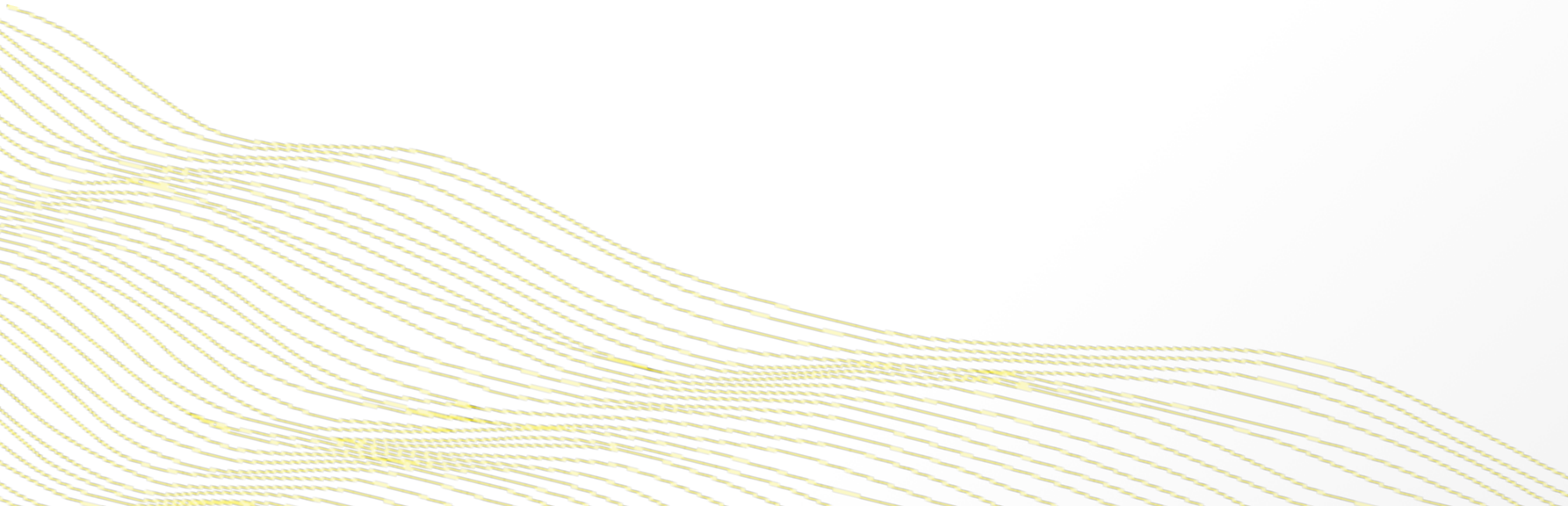
- Provides access to high performance, distributed machine learning algorithms to Spark workflows.
- Alternative to the default MLlib library in Spark.

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).

https://github.com/h2oai/rsparkling

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!

Tutorial: Stacking in H2O

http://tinyurl.com/h2o-stacking-docs

H2O Resources

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

Thank you!

@ledell on Github, Twitter

erin@h2o.ai

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