

New Developments in H2O Scalable Machine Learning



Berkeley, CA Sept 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

Agenda



- Who/What is H2O?
- H2O Machine Learning Platform
- New/active developments in H2O:
 - Super Learning with Random Grids
 - H2O + Hyperopt
 - Deep Water (GPU Deep Learning)
 - RSparkling (R + Spark + H2O)
 - AutoML

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

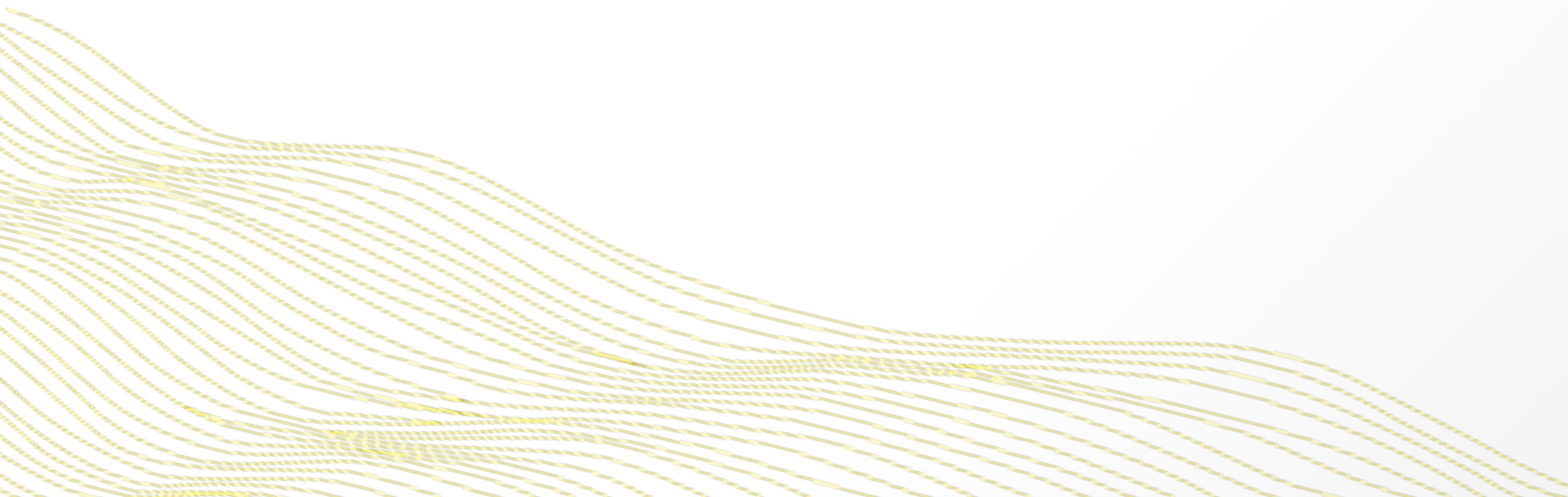
- Professor of Statistics and Health Research and Policy, Stanford University
- PhD in Statistics, Stanford University
- Co-author, *The Elements of Statistical Learning: Prediction, Inference and Data Mining*
- Author, *Regression Shrinkage and Selection via the Lasso*
- Co-author, *An Introduction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



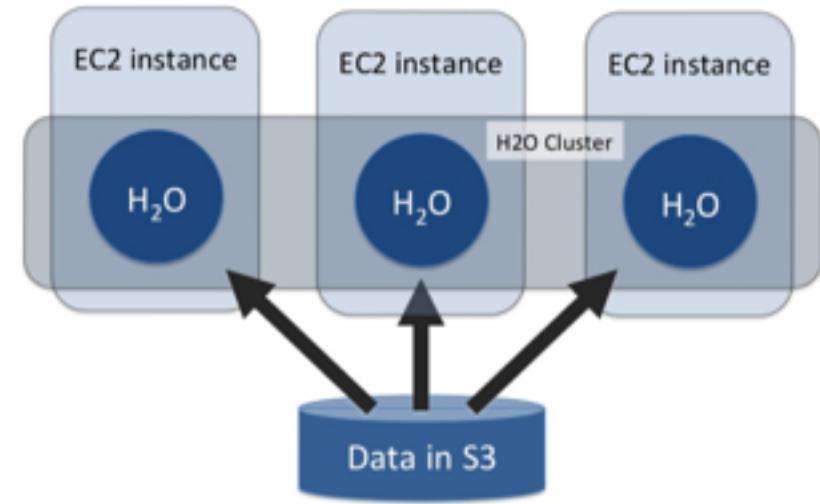
H2O Platform Overview

- 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

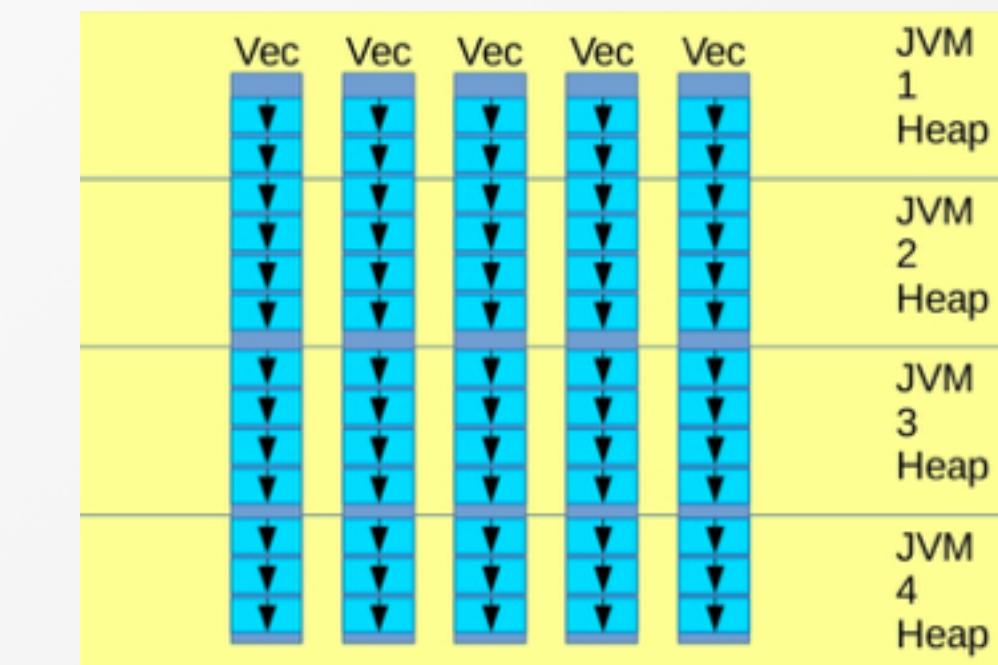
H2O Cluster



- 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
- 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

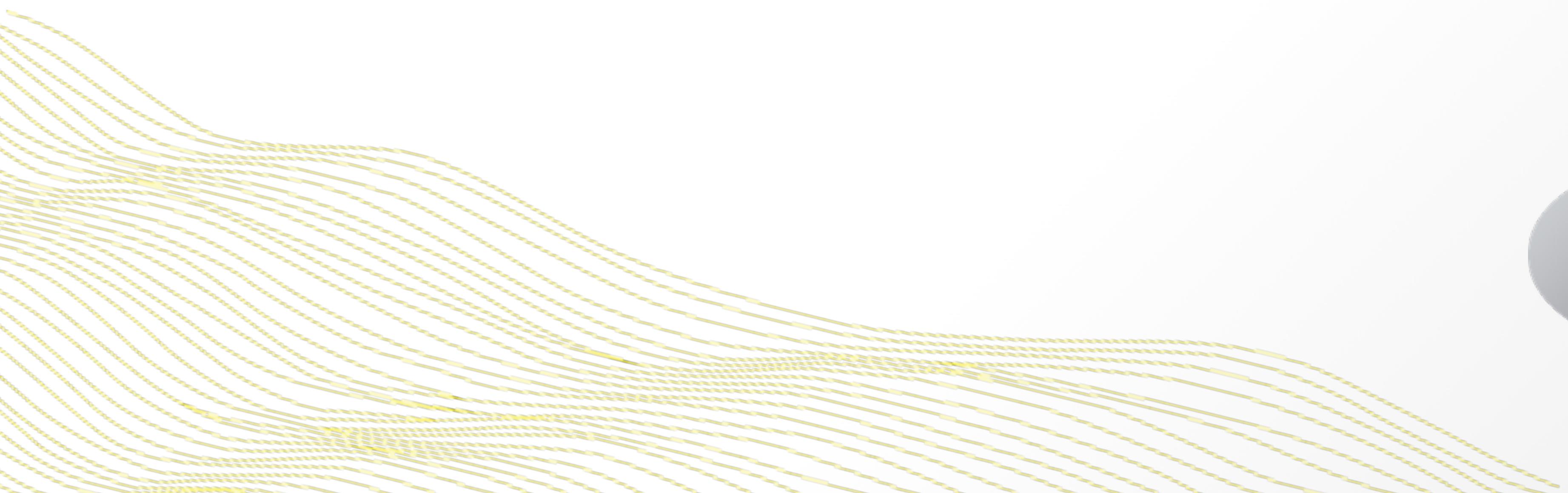
Solvers & Optimization

- 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 and Python



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 Startup & Load Data

Example

```
library(h2o) # First install from CRAN
localH2O <- h2o.init() # Initialize the H2O cluster

# Data directly into H2O cluster (avoids R)
train <- h2o.importFile(path = "train.csv")

# Data into H2O from R data.frame
train <- as.h2o(my_df)
```

H2O Supervised ML

Example

```
y <- "Class"  
x <- setdiff(names(train), y)  
  
fit <- h2o.gbm(x = x, y = y, training_frame = train)  
  
pred <- h2o.predict(fit, test)
```

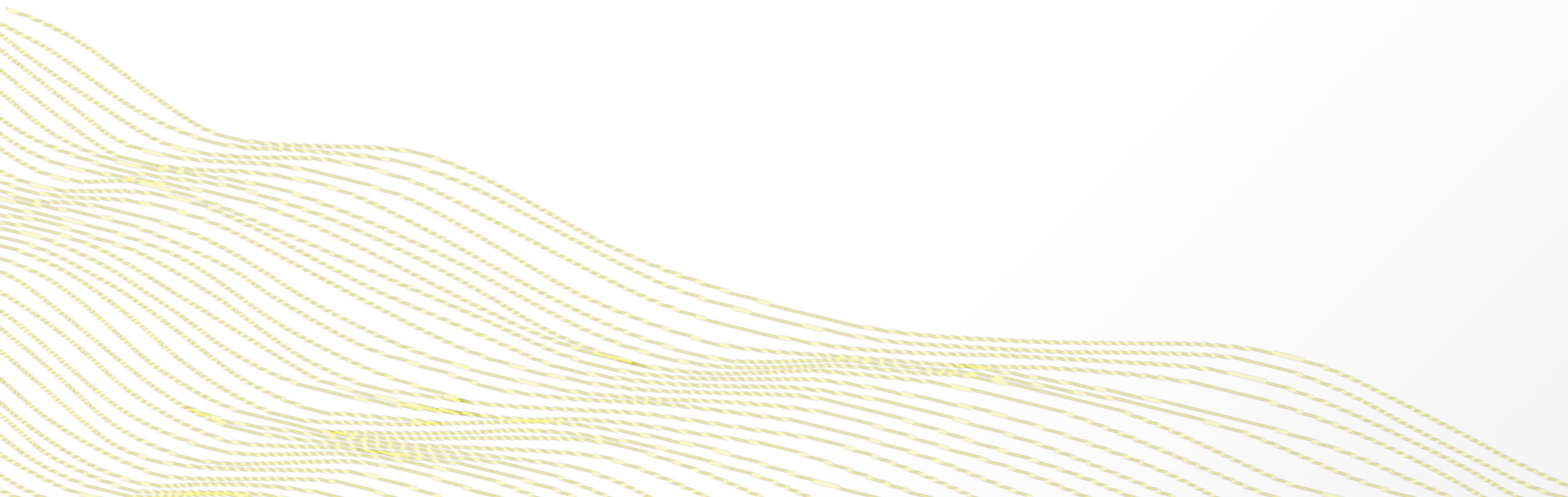
H2O Cartesian Grid Search

Example

```
hidden_opt <- list(c(200,200), c(100,300,100), c(500,500))
l1_opt <- c(1e-5,1e-7)
hyper_params <- list(hidden = hidden_opt, l1 = l1_opt)

grid <- h2o.grid(algorithm = "deeplearning",
                  hyper_params = hyper_params,
                  x = x, y = y,
                  training_frame = train,
                  validation_frame = valid)
```

Super Learning with Random Grids



H2O Random Grid Search

Example

```
search_criteria <- list(strategy = "RandomDiscrete",
                         max_runtime_secs = 600)

grid <- h2o.grid(algorithm = "deeplearning",
                  hyper_params = hyper_params,
                  search_criteria = search_criteria,
                  x = x, y = y,
                  training_frame = train,
                  validation_frame = valid)
```

Super Learning (Stacking) with Random Grids

Example

```
# Create a list of all the base models
models <- c(gbm_models, rf_models, dl_models, glm_models)

# Let's stack!
stack <- h2o.stack(models = models,
                    response_frame = train[,y],
                    metalearner = metalearner)
```

H2O Ensemble Resources

H2O Ensemble training guide:

<http://tinyurl.com/learn-h2o-ensemble>

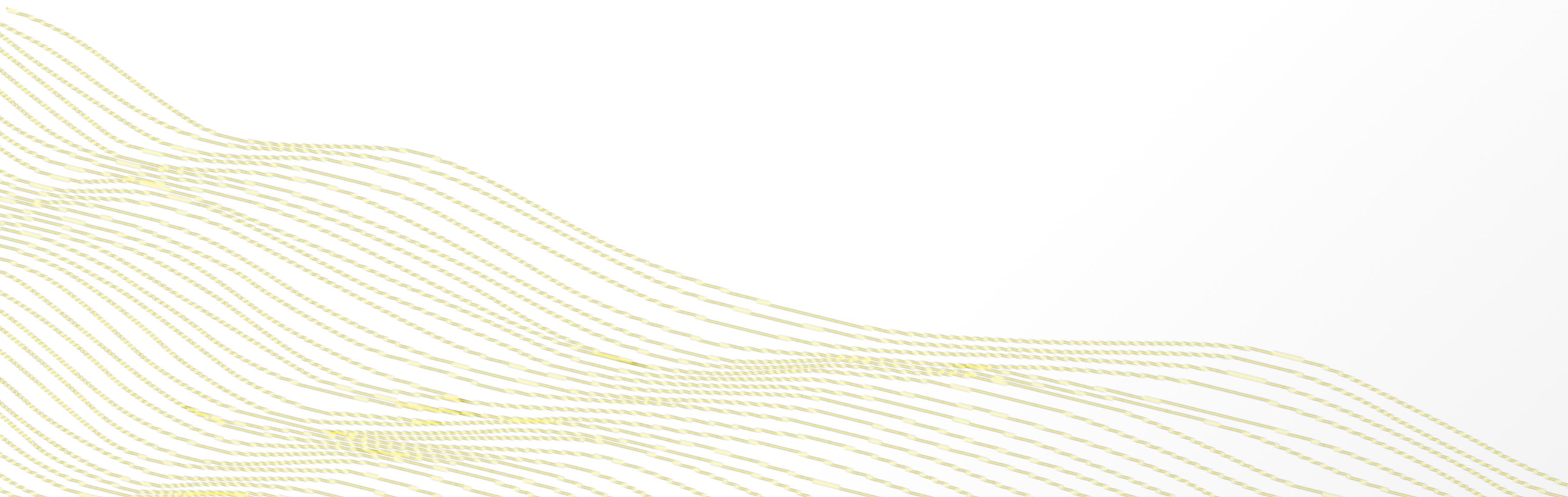
H2O Ensemble homepage on Github:

<http://tinyurl.com/github-h2o-ensemble>

H2O Ensemble R Demos:

<http://tinyurl.com/h2o-ensemble-demos>

H₂O + Hyperopt



h2o-hyperopt

- A Python package which is an extension to the H2O and Hyperopt Python packages.
- This is a convenience wrapper around Hyperopt for fast prototyping with H2O models. This allows H2O users to utilize the power of the Hyperopt library without learning to use it.
- Instead, just define optimizers with the parameters you want to optimize over.

<http://tinyurl.com/h2o-hyperopt>

h2o-hyperopt Example Code

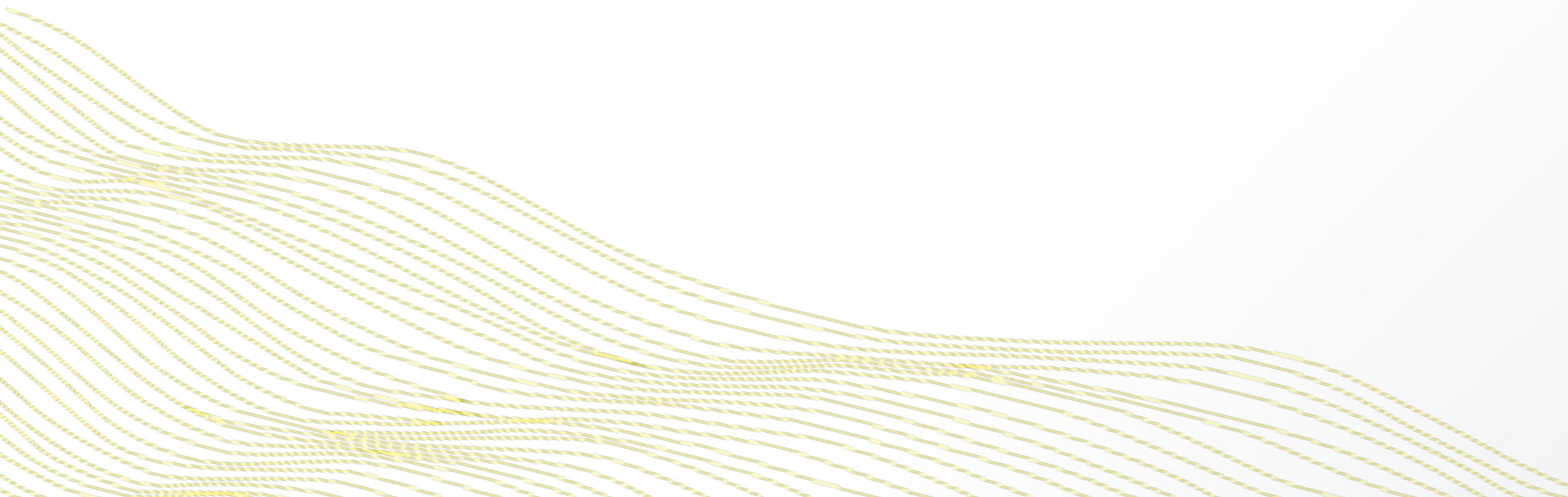
```
import h2ohyperopt
model = h2ohyperopt.GBMOptimizer(metric='auc')
# Selecting the optimization parameters
model.select_optimization_parameters({'col_sample_rate': 'Default',
                                       'ntrees': 200,
                                       'learn_rate': ('uniform',(0.05, 0.2)),
                                       'nfolds': 7})

# trainFr - Training frame
# validFr - Validation frame
# predictors - List of training variables
# response - String with response column name

model.start_optimization(num_evals=10, trainingFr=trainFr,
                         validationFr=validFr, response=response,
                         predictors=predictors)

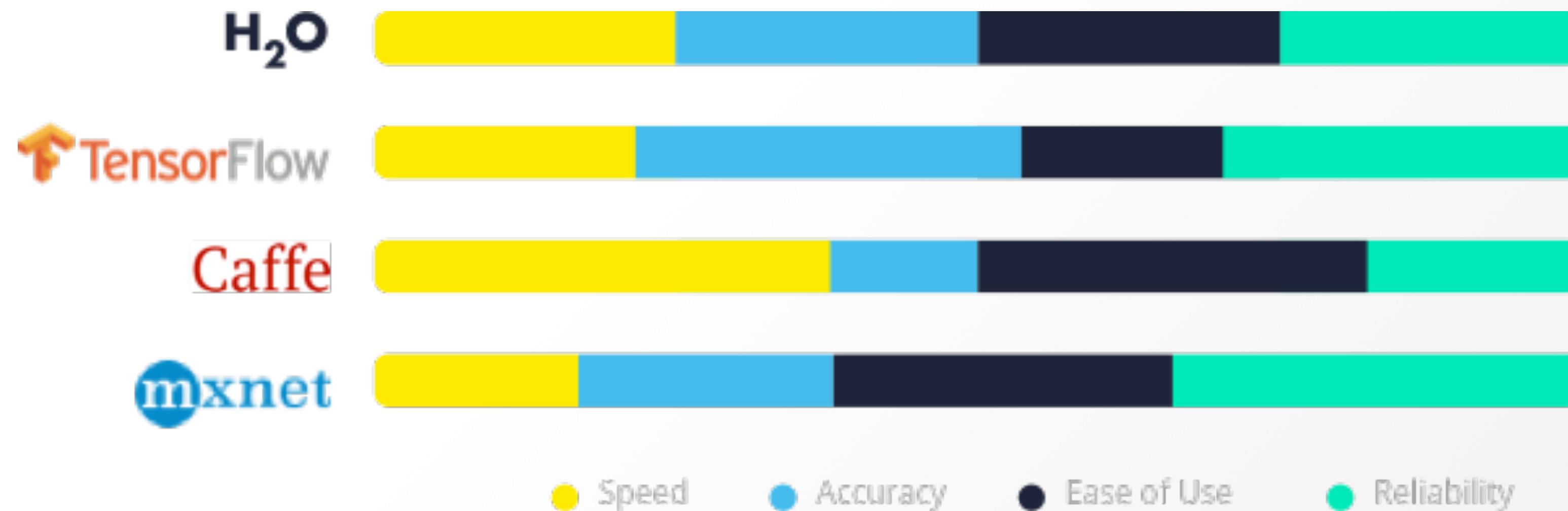
#To get the best model
print model.best_model
```

Deep Water



Deep Water

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

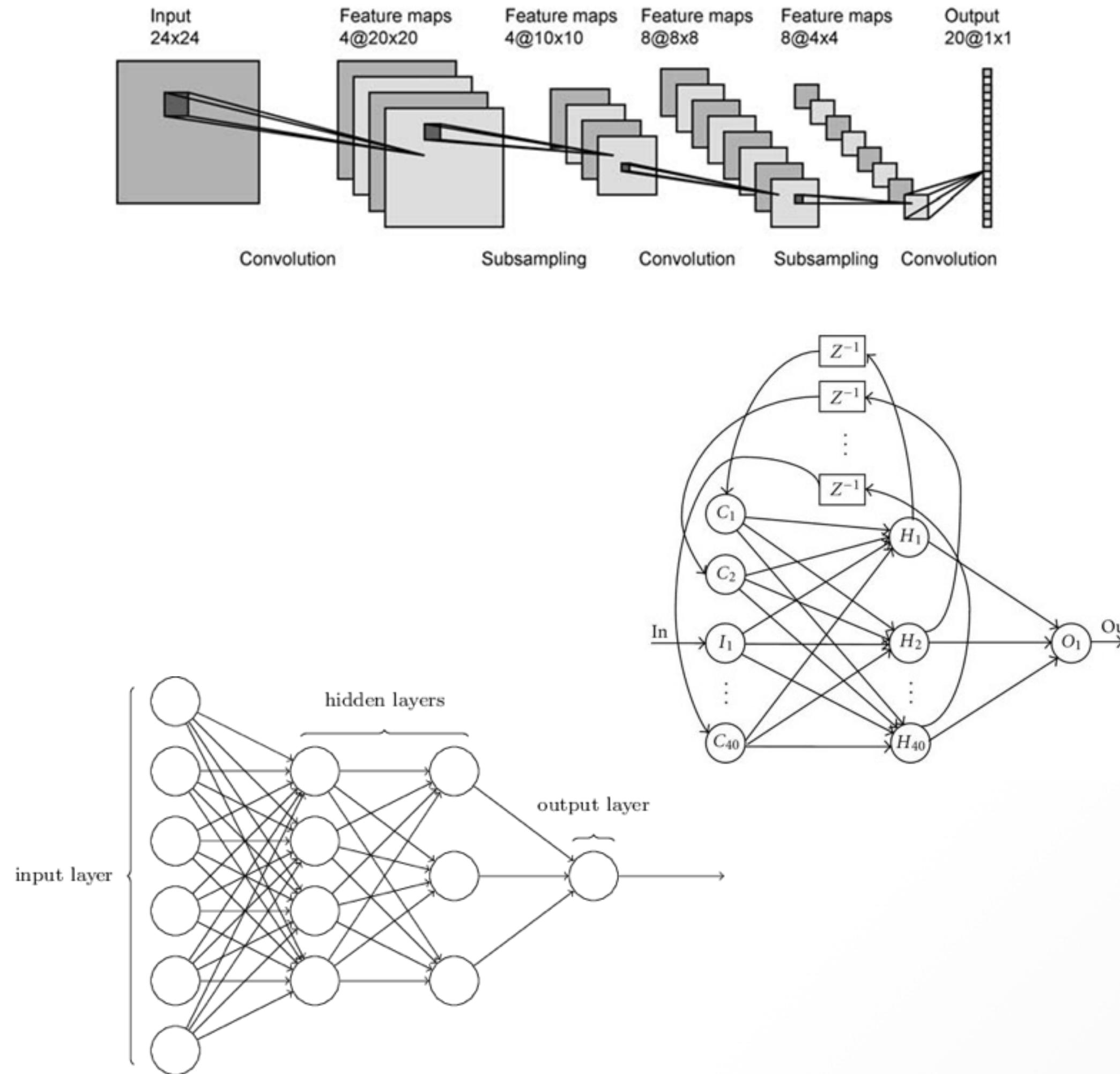


Deep Water

- 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

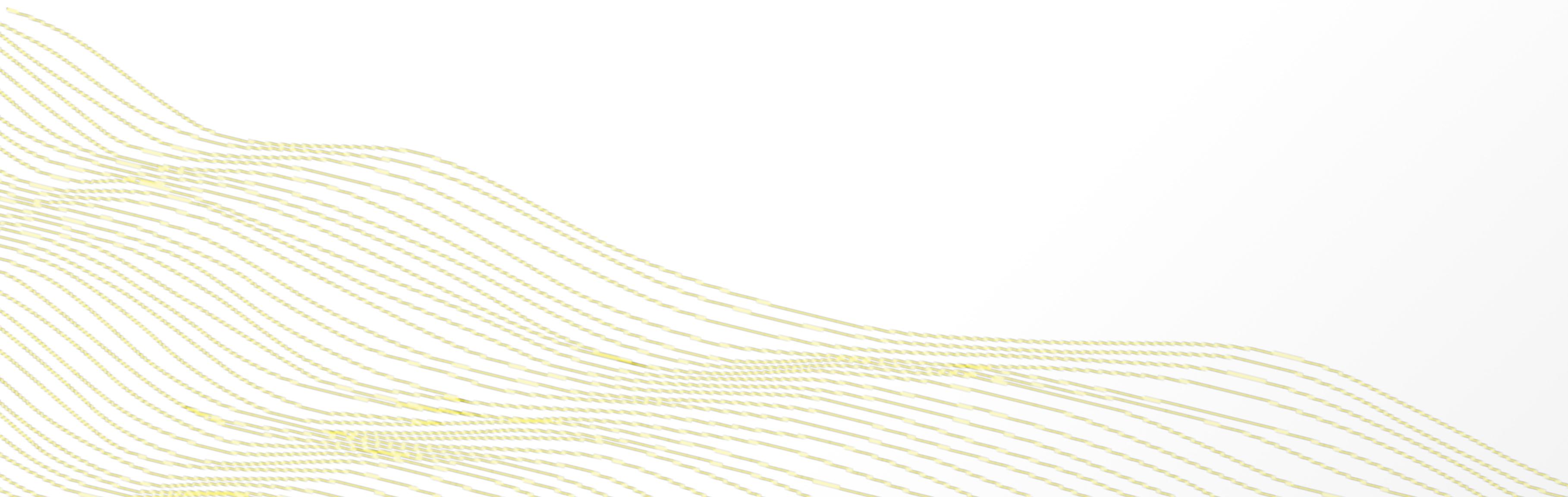
<https://github.com/h2oai/deepwater>

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

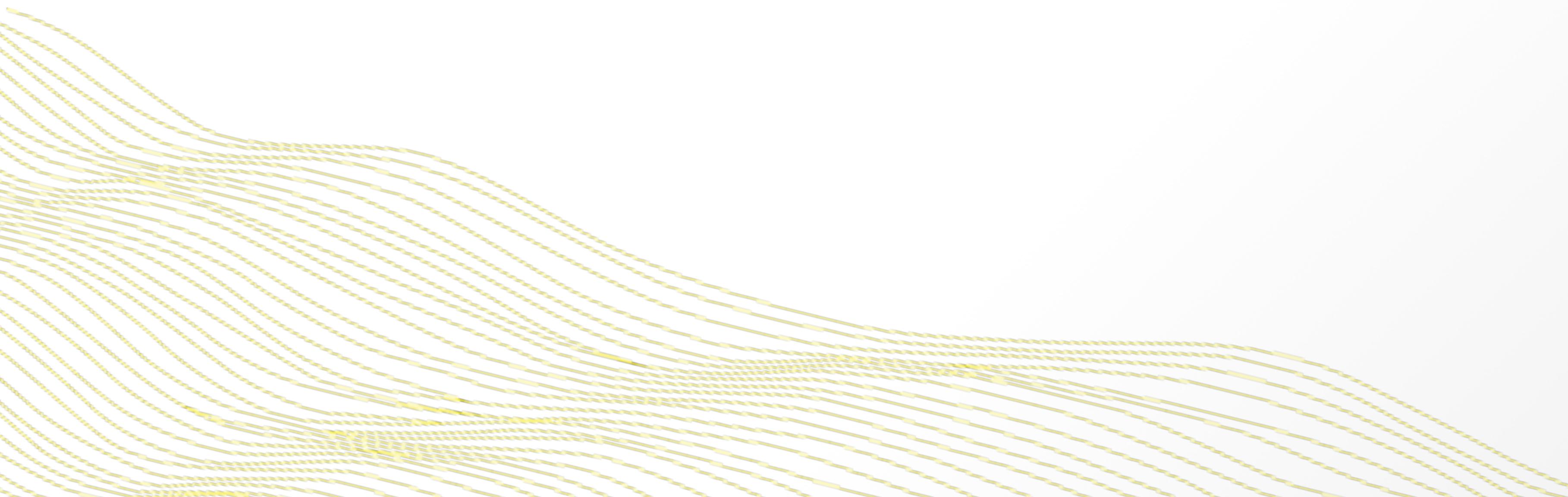
- Provides access to high performance, distributed machine learning algorithms to Spark workflows.
- Alternative to the default MLlib library in Spark.

RSparkling

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

<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 not yet available.

H2O R & Python Tutorial

<http://tinyurl.com/h2o-tutorial-rpy>



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 evaluating model performance on a test set.

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

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
```

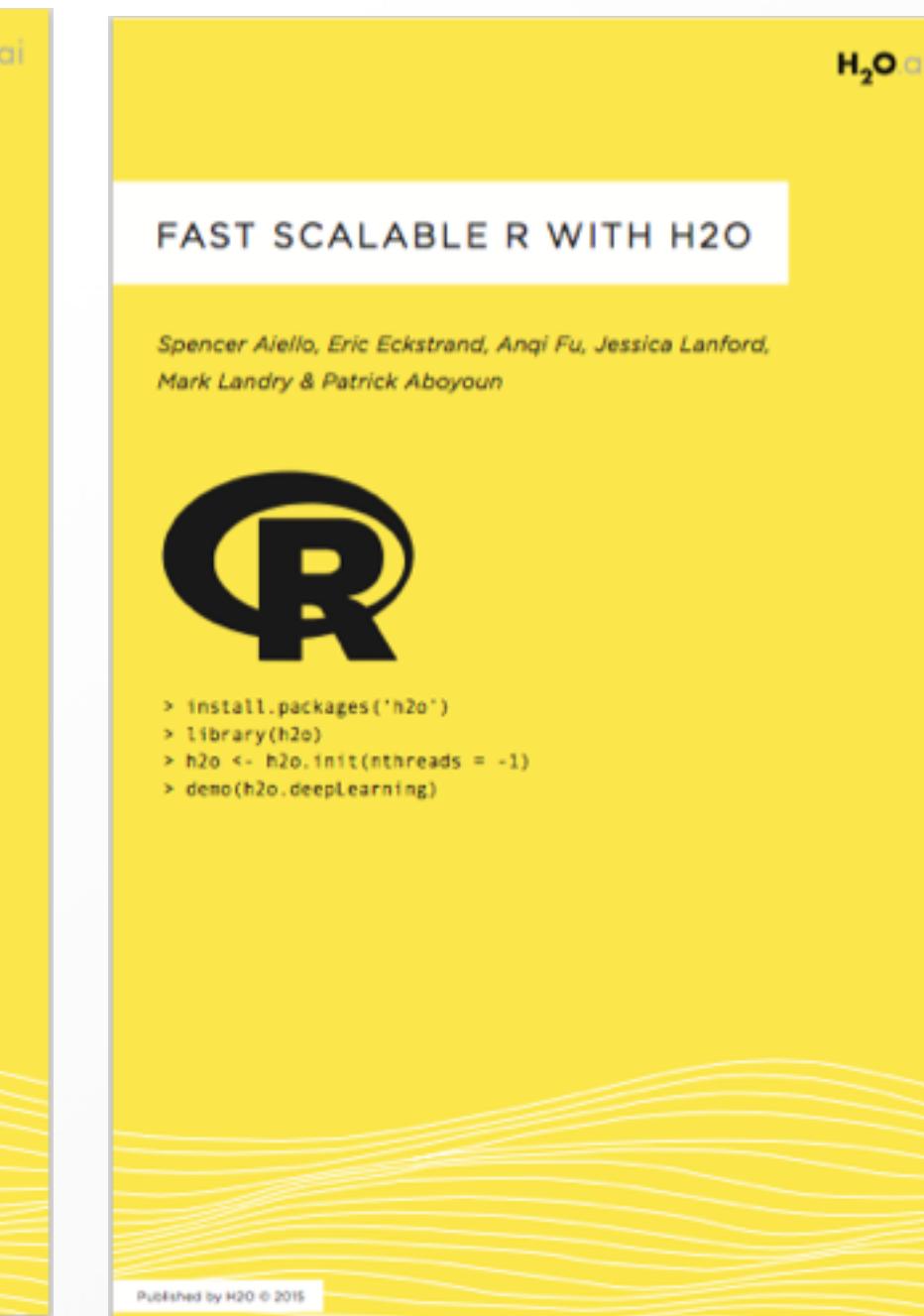
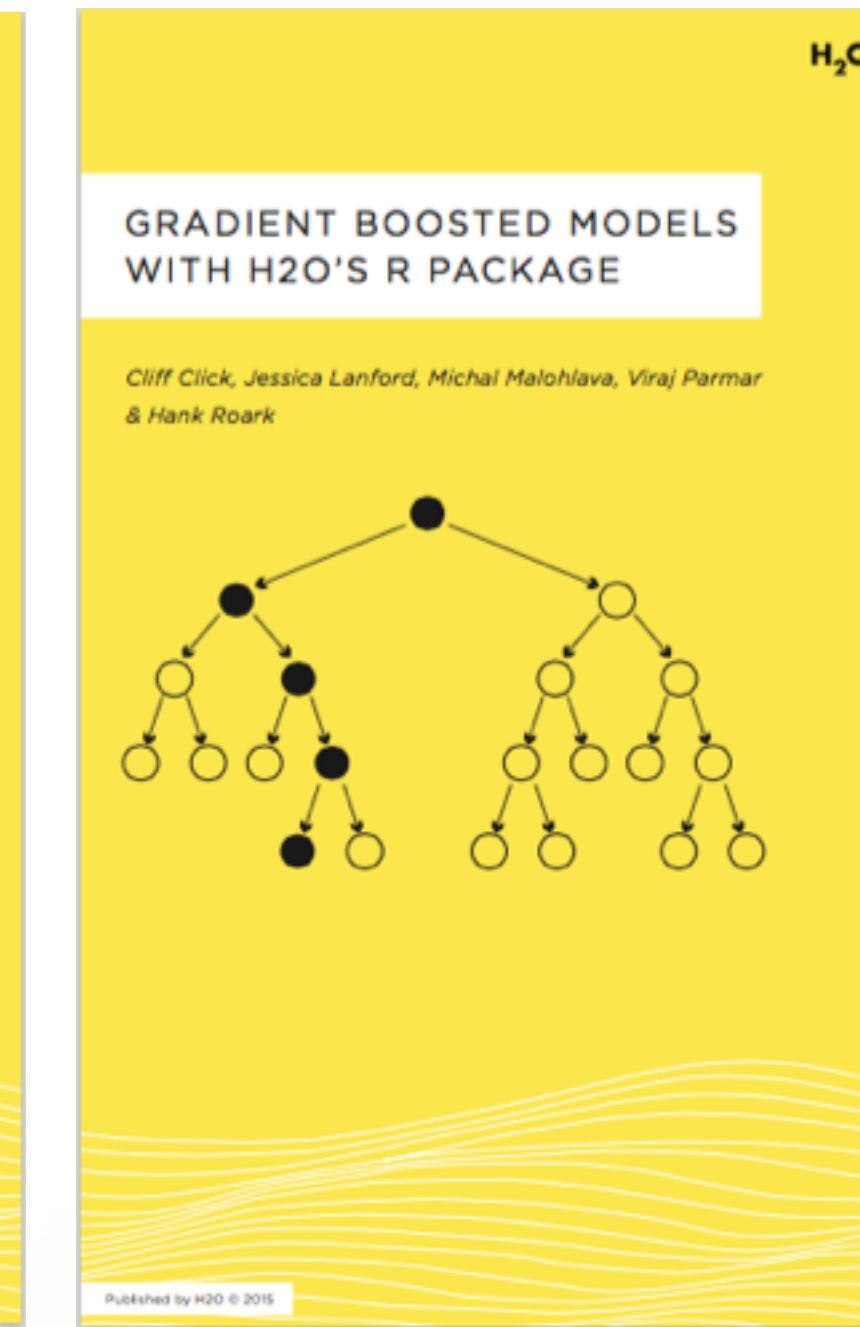
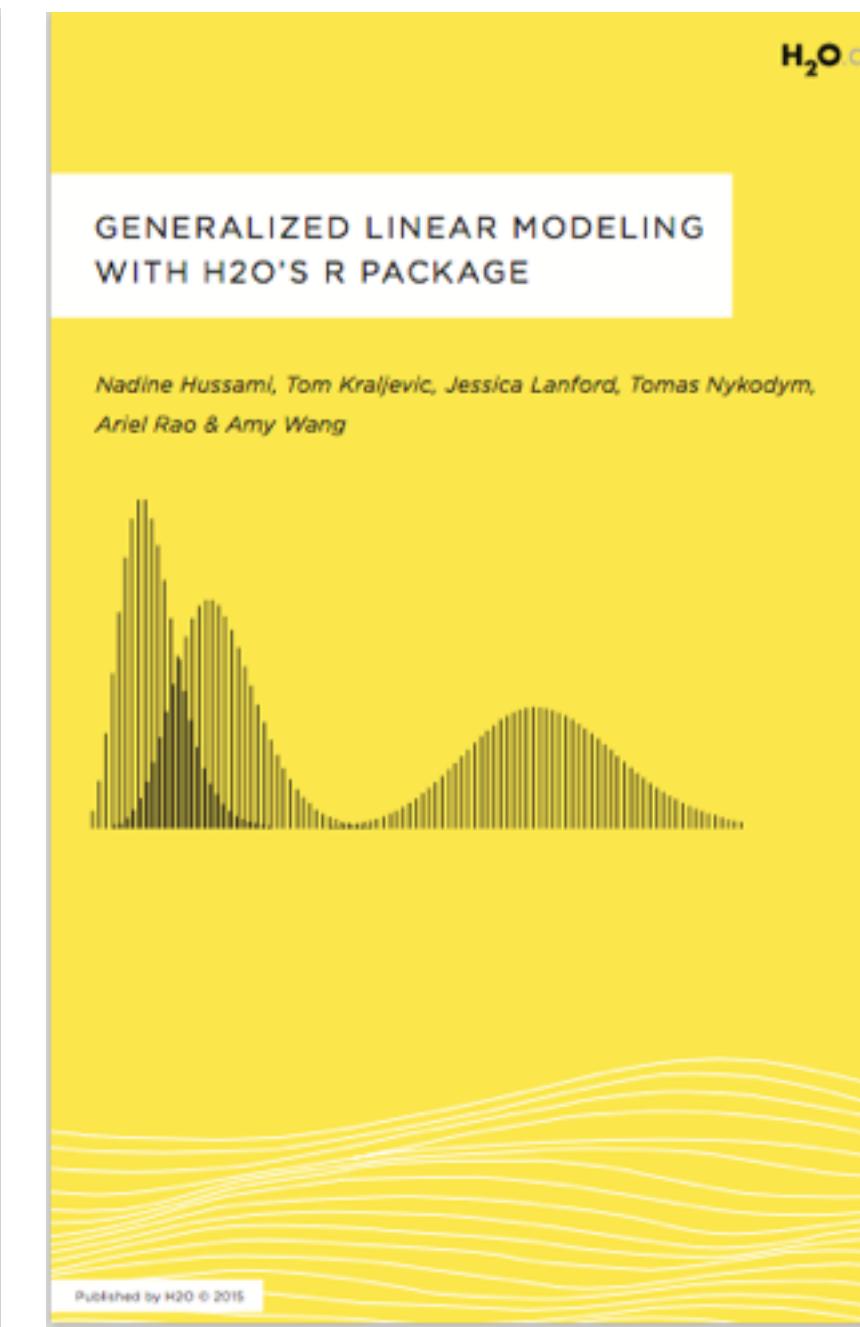
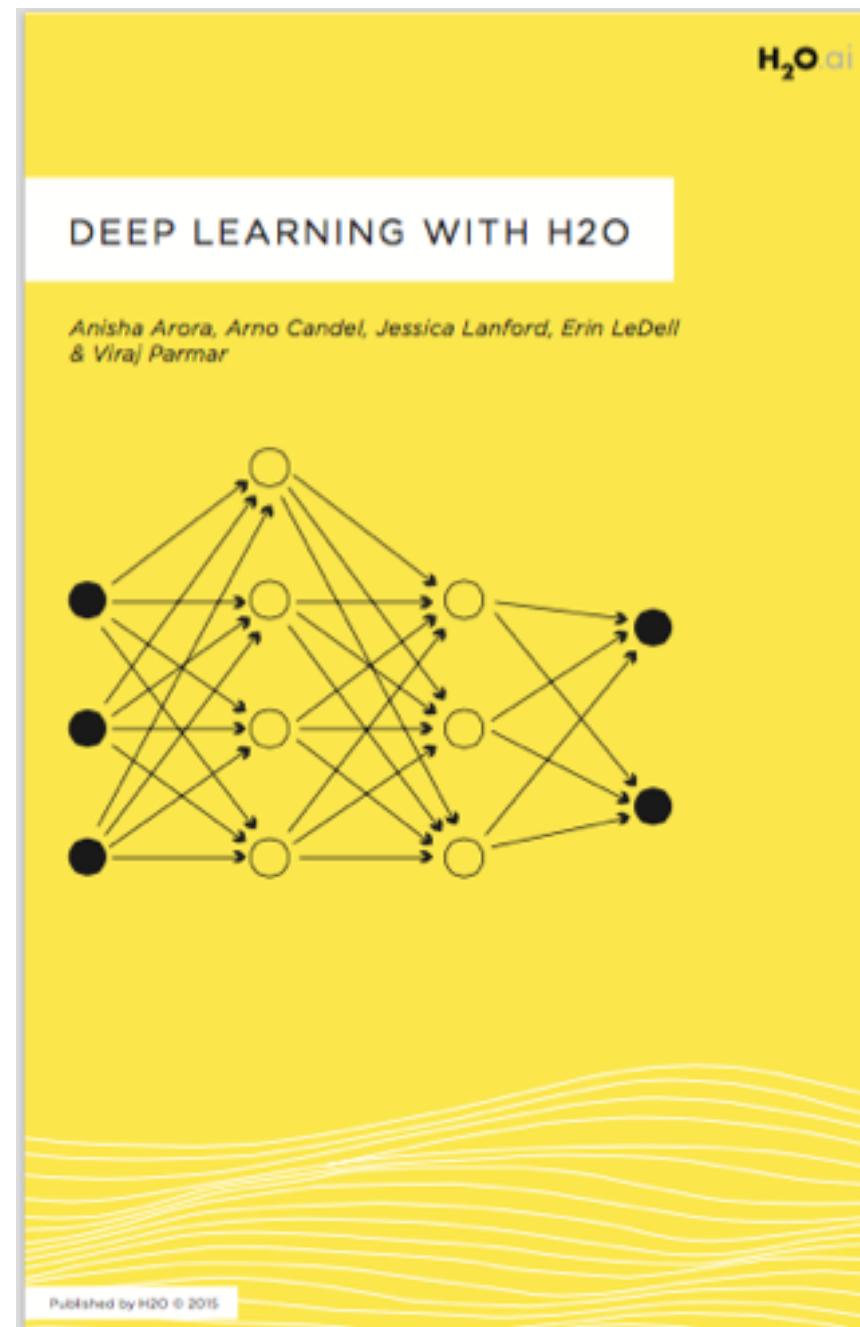
The second training module demonstrates how to find the best set of model parameters for each model using Grid Search.

H2O Resources

- H2O Online Training: <http://learn.h2o.ai>
- H2O Tutorials: <https://github.com/h2oai/h2o-tutorials>
- H2O Slidedecks: <http://www.slideshare.net/0xdata>
- H2O Video Presentations: <https://www.youtube.com/user/0xdata>
- H2O Community Events & Meetups: <http://h2o.ai/events>



H2O Booklets



<http://docs.h2o.ai/>

Thank you!

@ledell on Github, Twitter
erin@h2o.ai

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