Using H2O Random Grid Search for Hyper-parameters Optimization

Jo-fai (Joe) Chow

Data Scientist

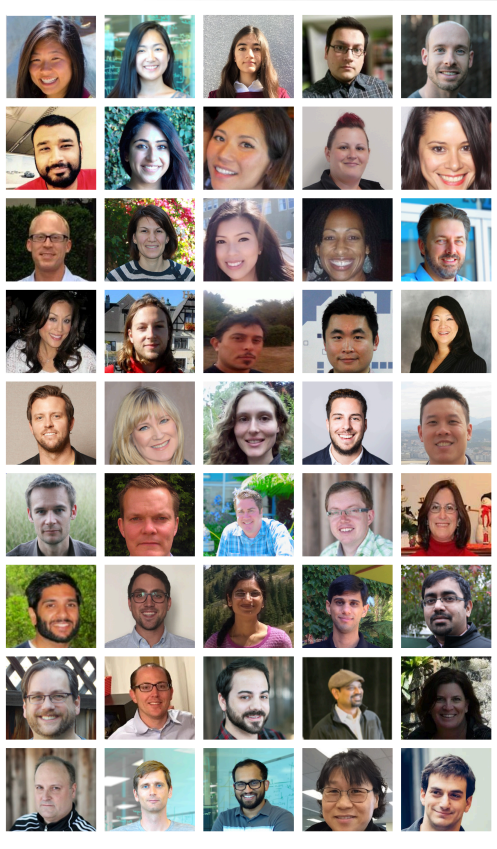
joe@h2o.ai

WHO AM I

- Customer Data Scientist at H2O.ai
- Background
  - Telecom (Virgin Media)
  - Data Science Platform (Domino Data Lab)
  - Water Engineering + Machine Learning Research (STREAM Industrial Doctorate Centre)

ABOUT H2O

- Company
  - Team: 50 (45 shown)
  - Founded in 2012, Mountain View, California.
  - Venture capital backed
- Products
  - Open-source machine learning platform.
  - Flow (Web), R, Python, Spark, Hadoop interfaces.



ABOUT THIS TALK

- Story of a baker and a data scientist
  - Why you should care
- Hyper-parameters optimization
  - Common techniques
  - H2O Python API
- Other H2O features for streamlining workflow

STORY OF A BAKER

- Making a cake
  - Source
    - Ingredients
  - Process:
    - Mixing
    - Baking
    - Decorating
  - End product
    - A nice looking cake

Credit: www.dphotographer.co.uk/image/201305/baking_a_cake

STORY OF A DATA SCIENTIST

- Making a data product
  - Source
    - Raw data
  - Process:
    - Data munging
    - Analyzing / Modeling
    - Reporting
  - End product
    - Apps, graphs or reports

Credit: www.simranjindal.com

BAKER AND DATA SCIENTIST

- What do they have in common?
  - Process is important to bakers and data scientists. Yet, most customers do not appreciate the effort.
  - Most customers only care about raw materials quality and end products.

WHY YOU SHOULD CARE

- We can use machine/software to automate some laborious tasks.
- We can spend more time on quality assurance and presentation.
- This talk is about making one specific task, hyper-parameters tuning, more efficient.

HYPER-PARAMETERS OPTIMISATION

- Overview
  - Optimizing an algorithm's performance.
    - e.g. Random Forest, Gradient Boosting Machine (GBM)
  - Trying different sets of hyper-parameters within a defined search space.
  - No rules of thumb.

HYPER-PARAMETERS OPTIMISATION

- Example of hyper-parameters in H2O
  - Random Forest:
    - No. of trees, depth of trees, sample rate …
  - Gradient Boosting Machine (GBM):
    - No. of trees, depth of trees, learning rate, sample rate …
  - Deep Learning:
    - Activation, hidden layer sizes, L1, L2, dropout ratios …

COMMON TECHNIQUES

- Manual search
  - Tuning by hand - inefficient
  - Expert opinion (not always reliable)
- Grid search
  - Automated search within a defined space
  - Computationally expensive
- Random grid search
  - More efficient than manual / grid search
  - Equal performance in less time

RANDOM GRID SEARCH – DOES IT WORK?

- Random Search for Hyper-Parameter Optimization
  - Journal of Machine Learning Research (2012)
  - James Bergstra and Yoshua Bengio
  - *“Compared with deep belief networks configured by a thoughtful combination of manual search and grid search, purely random search found statistically equal performance on four of seven data sets, and superior performance on one of seven.”*

RELATED FEATURE – EARLY STOPPING

- A technique for regularization.
- Avoid over-fitting the training set.
- Useful when combined with hyper-parameter search:
  - Additional controls (e.g. time constraint, tolerance)

H2O RANDOM GRID SEARCH

- Objectives
  - Optimize model performance based on evaluation metric.
  - Explore the defined search space randomly.
  - Use early-stopping for regularization and additional controls.

RANDOM GRID SEARCH (PYTHON API)

```
# Define search space for hyper-parameters
hyper_parameters = {'ntrees':[10,50,100,200,500], 'max_depth':[5,10,15,20,25]}

# Define search criteria
search_criteria =
{
  "strategy": "RandomDiscrete",
  "max_runtime_secs": 600,
  "max_models": 10,
  "stopping_metric": "AUTO",
  "stopping_tolerance": 0.00001,
  "stopping_rounds": 5,
  "seed": 123456
}

# Set up random grid search for Random Forest
grid_search = H2OGridSearch(H2ORandomForestEstimator, hyper_parameters, search_criteria)
grid_search.train(x=["x1", "x2"], y="y", training_frame=train)

# Show the search results
grid_search.show()
```

RANDOM GRID SEARCH

- Outputs
  - Best model based on metric
  - A set of hyper-parameters for the best model
- Other APIs
  - R, REST, Java (see documentation on GitHub)

OTHER H2O FEATURES

- h2oEnsemble
  - Better predictive performance
- Sparkling Water = Spark + H2O
- Plain Old Java Object (POJO)
  - Productionize H2O models

CONCLUSIONS

- Most people only care about the end product.
- Use H2O random grid search to save time on hyper-parameters tuning.
- Spend more time on quality assurance and presentation.

CONCLUSIONS

- H2O Random Grid Search
  - An efficient way to tune hyper-parameters
  - APIs for Python, R, Java, REST
  - Do check out the code examples on GitHub
- Combine with other H2O features
  - Streamline data science workflow

ACKNOWLEDGEMENTS

- GoDataDriven
- Conference sponsors
- My colleagues at H2O.ai

THANK YOU

- Resources
  - Slides + code – github.com/h2oai/h2o-meetups
  - Download H2O – www.h2o.ai
  - Documentation – www.h2o.ai/docs/
  - joe@h2o.ai
- We are hiring – www.h2o.ai/careers/