Scaling the AI hierarchy of needs with TensorFlow, Spark, and Hops 🍺

@jim_dowling
“Methods that scale with computation are the future of AI”*

- Rich Sutton (Founding Father of Reinforcement Learning)

* https://www.youtube.com/watch?v=EeMCEQa85tw
Massive Increase in Compute for AI*

*https://blog.openai.com/ai-and-compute

3.5 month-doubling time

Distributed Systems
AI Hierarchy of Needs

- Data Scientists
- Data Engineers
- Data Scientists?
- DDL (Distributed Deep Learning)
- Deep Learning, RL
- Machine Learning (ML)
- Data Analytics
- Data Pipelines

Lots of GPUs
GPUs
Big Data
Model Parallelism

One Big Model on 3 GPUs

Copy of the Model on each GPU

Data Parallelism

Distributed Filesystem

32 files 32 files 32 files

Batch Size of 3200

5/53
Mini AI Arms Race

Facebook vs Google

ImageNet

CNNs

Distribution
ImageNet Challenge

- 1,000 object classes (categories).
- Images:
  - 1.2 M train
  - 100k test.

[Image from https://www.slideshare.net/xavigiro/image-classification-on-imagenet-d114-2017-upc-deep-learning-for-computer-vision]
Improvements in ImageNet – Accuracy

Top-1 Accuracy for ImageNet

- More Compute
- More Data
- 3 Bn Images
- AutoML with RL
- AutoML with GA

GOOGLE-1 (MAY '17)
GOOGLE-2 (MARCH '18)
FACEBOOK (MAY '18)

ImageNet Top-1

Facebook: https://goo.gl/ERpJyr
Google-1: https://goo.gl/EV7Xv1
Google-2: https://goo.gl/eidnyQ
Improvements in ImageNet – Training Time

Training Time (mins) ImageNet >75% top-1

FACEBOOK (JUNE '17) vs. GOOGLE-1 (DEC '17) vs. GOOGLE-2 (MARCH '18)

- 256 Nvidia P100s
- 256 TPUv2s

Training Time (mins)

Facebook: https://goo.gl/ERpJyr
Google-1: https://goo.gl/EV7Xv1
Google-2: https://goo.gl/eidnyQ
ImageNet – Files/Sec Processed

1000s Files/Sec Processed

FACEBOOK (JUNE ’17)  GOOGLE-1 (DEC ’17)  GOOGLE-2 (MARCH ’18)

June’17 Files Processed (K/s)

HDFS ~80K/s

FB - https://goo.gl/ERpJyr
Google-1: https://goo.gl/EV7Xv1
Google-2: https://goo.gl/eidnyQ
HopsFS: Distributed FS is no longer a Bottleneck

Distributed Filesystem Read Throughput

Files Processed (K/s)

FACEBOOK  GOOGLE-1  GOOGLE-2  HOPSFS

PLUG for Hops

Scale Challenge Winner (2017)

HopsFS - https://goo.gl/yFCsGc
Need for a Distributed Filesystem

- Big Training/test datasets
- Model checkpointing
- Evaluation
- Distribute Training, Parallel Experiments
- Model Serving
Methods for Improving Model Accuracy

Reduced Generalization Error

Better Regularization Methods
Hyper Parameter Optimization
Design Better Models
Larger Training Datasets
Better Optimization Algorithms
How much Labelled Data do we Need?*

All Roads Lead to Distribution

Distributed Deep Learning

- Parallel Experiments
- Hyper Parameter Optimization
- Larger Training Datasets
- Auto ML
- Elastic Model Serving
- Distributed Training
- (Commodity) GPU Clusters
What should we non-Hyperscale AI Folks do?
GPU/TPU Services in the Cloud and On-Premise

- Managed Cloud Platforms
  - RiseML
  - FloydHub
  - Google Cloud ML
  - Microsoft Batch AI

- On-Premise/Cloud Platforms
  - KubeFlow
  - Hops
Hops – Python-Only Deep Learning
1. **Security by Design**: Projects as a Sandbox for self-service and teams

2. **Ease of use**: Data Scientists need only code Python

3. **Scale-out Deep Learning**: Parallel experiments, Distributed Training – Horovod/TensorFlowOnSpark
Python in the Cluster: Per-Project Conda Envs

Conda Repo

Data Owner

search "scikit-learn"

install/remove

Conda Env "X"

python-3.6, pandas-1.4, numpy-0.9

Project "X"

Python libraries are usable by Spark/Tensorflow
Hops: TLS Certs for Security (not Kerberos)

- **User Certificates:**
  - A user has a cert for every project he/she is a member of.

- **Service Certificates:**
  - NameNode, ResourceManager, Kafka, HiveServer2, Livy, etc

- **Application Certificates**

- **Supports Certification Revocation, Renewal, Reloading.**
GPU Resource Requests in Hops YARN

- 10 GPUs on 1 host
- 4 GPUs on any host
- 100 GPUs on 10 hosts with ‘Infiniband’

Hops supports a Heterogenous Mix of GPUs
ML in Production: Machine Learning Pipelines
A Machine Learning Pipeline

Data Collection → Data Transformation & Verification → Feature Extraction → Experimentation → Training → Test → Serving
Hops Small Data ML Pipeline

Data Collection → Data Transformation & Verification → Feature Extraction → Experimentation → Training → Test → Serving

TensorFlow → TfServing
Kafka → HopsYARN → Kubernetes → HopsFS

Project Teams (Data Engineers/Scientists)
Hops Big Data ML Pipeline

Data Collection → Data Transformation & Verification → Feature Extraction → Experimentation → Training → Test → Serving

- PySpark
- TensorFlow
- TfServing
- Kafka
- HopsYARN
- Kubernetes
- HopsFS

Project Teams (Data Engineers/Scientists)
Google Facets Overview

- Visualize data distributions
- Min/max/mean/media values for features
- Missing values in columns
- Facets Overview expects test/train datasets as input
Google Facets Dive

- Visualize the relationship between the data points across the different features of a dataset.
features = ["Age", "Occupation", "Sex", ..., "Country"]

h = hdfs.get_fs()
with h.open_file(hdfs.project_path() + "/TestJob/data/census/adult.data", "r") as trainFile:
    train_data = pd.read_csv(trainFile, names=features, sep=r"\s*,\s*", engine='python', na_values='\?')
    test_data = ...

facets.overview(train_data, test_data)
facets.dive(test_data.to_json(orient='records'))
Small Data Preparation with tf.data API

```python
def input_fn(batch_size):
    files = tf.data.Dataset.list_files(IMAGE_DIR)

    def tfrecord_dataset(filename):
        return tf.data.TFRecordDataset(filename,
                                        num_parallel_reads=32,
                                        buffer_size=8*1024*1024)

    dataset = files.apply(tf.data.parallel_interleave
                           (tfrecord_dataset, cycle_length=32, sloppy=True)
                        )
    dataset = dataset.apply(tf.data.map_and_batch(parser_fn, batch_size,
                                                  num_parallel_batches=4))
    dataset = dataset.prefetch(4)
    return dataset
```
Big Data Preparation with PySpark

images = `spark.readImages`(IMAGE_PATH, recursive = True, numPartitions=10, sampleRatio = 0.1).cache()

tr = (`ImageTransformer`().setOutputCol("transformed")
       .resize(height = 200, width = 200)
       .crop(0, 0, height = 180, width = 180)
)
smallImages = tr.transform(images).select("transformed")

# Output .tfrecords using TensorFlowOnSpark utility
dfutil.saveAsTFRecords(smallImages, OUTPUT_DIR)
The Outer Loop (hyperparameters):
“I have to run a hundred experiments to find the best model,” he complained, as he showed me his Jupyter notebooks. “That takes time. Every experiment takes a lot of programming, because there are so many different parameters.
[Rants of a Data Scientist]
GridSearch with TensorFlow/Spark on Hops

```python
def train(learning_rate, dropout):
    # [TensorFlow Code here]

    args_dict = {'learning_rate': [0.001, 0.005, 0.01],
                 'dropout': [0.5, 0.6]}

    experiment.launch(spark, train, args_dict)
```

Launch 6 Spark Executors
def train_cifar10(learning_rate, dropout):

[TensorFlow Code here]

dict =
{'learning_rate': [0.005, 0.00005], 'dropout': [0.01, 0.99], 'num_layers': [1,3]}
experiment.evolutionary_search(spark, train_cifar10, dict, direction='max',
popsize=10, generations=3, crossover=0.7, mutation=0.5)
Differential Evolution in Tensorboard (1/4)

CIFAR-10

accuracy

0.00 0.10 0.20 0.30 0.40 0.50 0.60 0.70 0.80 0.90 0.99

©2018 Logical Clocks AB. All Rights Reserved
Differential Evolution in Tensorboard (2/4)

accuracy

0.000
0.200
0.400
0.600
0.800
1.000

CIFAR-10
Differential Evolution in Tensorboard (3/4)

CIFAR-10
Differential Evolution in Tensorboard (4/4)

CIFAR-10
The Inner Loop (training):

“All these experiments took a lot of computation — we used hundreds of GPUs/TPUs for days. Much like a single modern computer can outperform thousands of decades-old machines, we hope that in the future these experiments will become household.”

[Google SoTA ImageNet, Cifar-10, March18]
Distributed Training: Theory and Practice

Image from @hardmaru on Twitter.
Network Bandwidth is the Bottleneck for Distributed Training
Horovod - AllReduce Inception V4 Performance

Inception V4, 20 Nvidia1080Tis on 2 GPU Servers, 40 Gb/s

% GPU Utilization

% Network Bandwidth
Parameter Server – Inception V4 Performance

Horovod, Inception V4, 2 ParamServers, 40 Gb/s, 10 Nvidia 1080Tis

% GPU Utilization

% Network Bandwidth
Distributed Training with Horovod on Hops

```python
hvd.init()
opt = hvd.DistributedOptimizer(opt)
if hvd.local_rank() == 0:
    [TensorFlow Code here]
    ...
else:
    [TensorFlow Code here]
    ...
```

```python
allreduce.launch(spark, 'hdfs:///Projects/…/all_reduce.ipynb')
```
Hops API

• Python (also Java/Scala)
  - Manage tensorboard, Load/save models in HDFS
  - Horovod, TensorFlowOnSpark
  - Parallel experiments
    • Gridsearch
    • Model Architecture Search with Genetic Algorithms
  - Secure Streaming Analytics with Kafka/Spark/Flink
    • SSL/TLS certs, Avro Schema, Endpoints for Kafka/Zookeeper/etc
TensorFlow Model Serving

Data Acquisition
Clean/Transform Data
Feature Extraction
Experimentation
Training
Test + Serve
Training-Serving Skew

• Monitor differences between performance during training and performance during serving.
  - Differences in how you process data in training vs serving.
  - Differences in the training data and live data for serving.
  - A feedback loop between your model and your algorithm.

• When to retrain?
  - If you look at the input data and use covariant shift to see when it deviates significantly from the data that was used to train the model on.
Summary

• The future of Deep Learning is Distributed
  https://www.oreilly.com/ideas/distributed-tensorflow

• Hops is a new Data Platform with first-class support for
  Python / Deep Learning / ML / Data Governance / GPUs

“It is starting to look like deep learning workflows of the future
feature autotuned architectures running with autotuned
compute schedules across arbitrary backends.”

Andrej Karpathy - Head of AI @ Tesla

*https://twitter.com/karpathy/status/972701240017633281
The Team

Active:
Jim Dowling, Seif Haridi, Gautier Berthou, Salman Niazi, Mahmoud Ismail, Theofilos Kakantousis, Ermias Gebremeskel, Antonios Kouzoupis, Alex Ormenisan, Fabio Buso, Robin Andersson, August Bonds.

Alumni:

www.hops.io
@hopshadoop