Automated Machine Learning Workflow for Distributed Big Data Using Analytics Zoo

Jason Dai

CVPR 2020 Tutorial

Overview

AI ON BIG DATA

Distributed, High-Performance

Deep Learning Framework

for Apache Spark

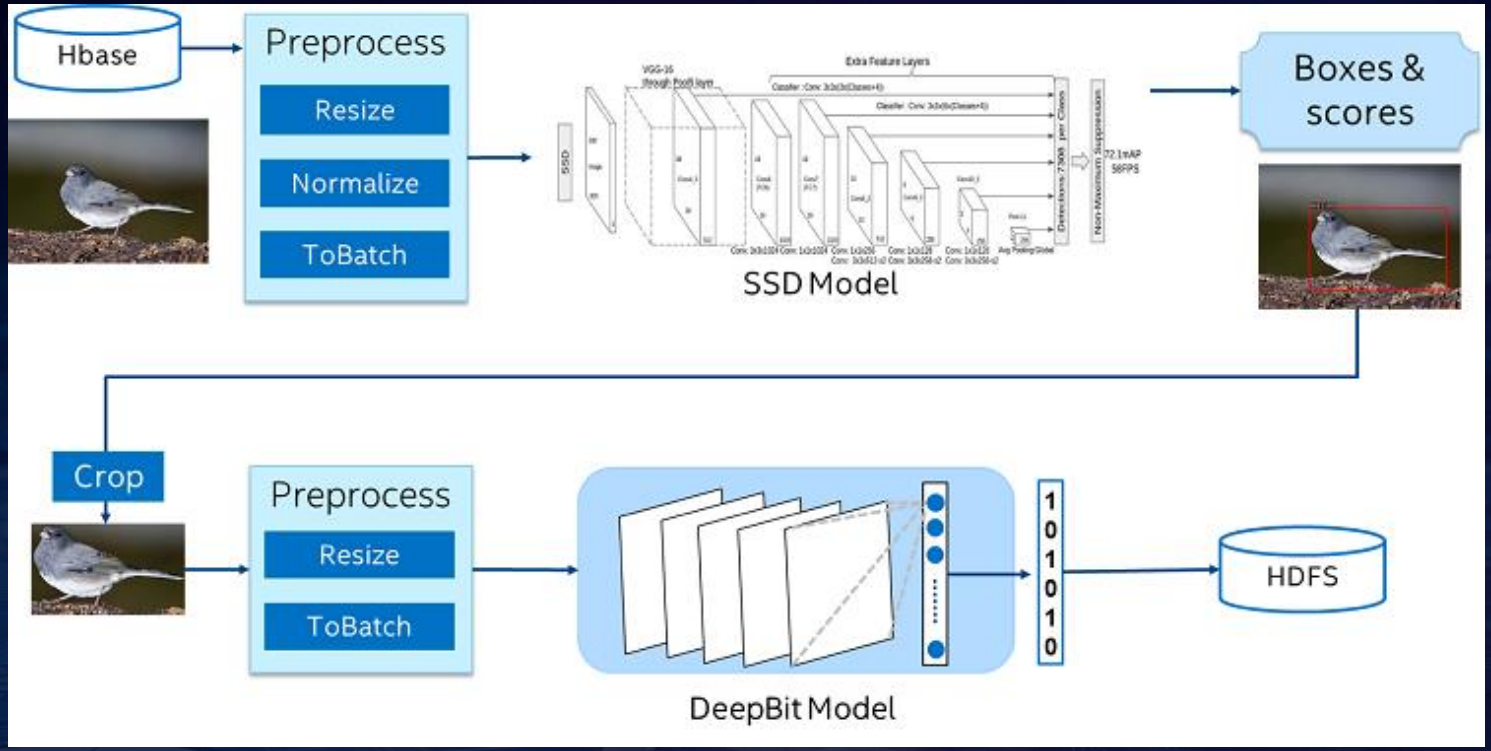
https://github.com/intel-analytics/bigdl

Unified Analytics + AI Platform

for TensorFlow, PyTorch, Keras, BigDL, Ray and Apache Spark

https://github.com/intel-analytics/analytics-zoo

Motivation: Object Feature Extraction at JD.com



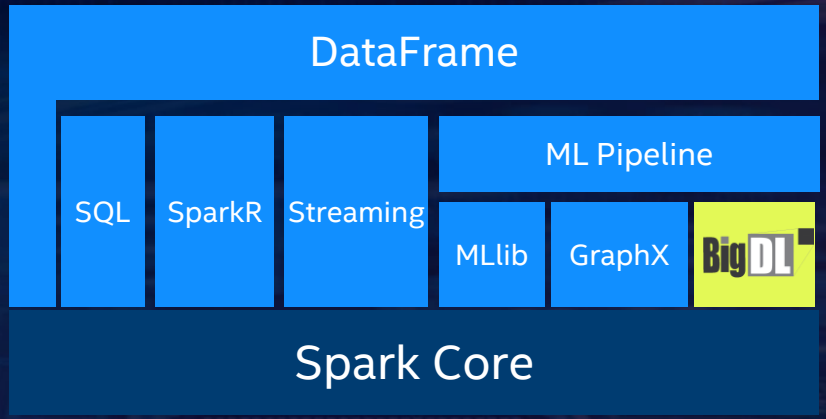
https://software.intel.com/en-us/articles/building-large-scale-image-feature-extraction-with-bigdl-at-jdcom

Efficiently scale out with BigDL with *3.83x* speed-up (vs. GPU severs) as benchmarked by JD

BigDL

Distributed deep learning framework for Apache Spark

- Write deep learning applications as ***standard Spark programs***
- Run on existing Spark/Hadoop clusters (***no changes needed***)
- Scalable and high performance
  - Optimized for large-scale big data clusters



https://github.com/intel-analytics/BigDL

"BigDL: A Distributed Deep Learning Framework for Big Data", ACM Symposium of Cloud Computing conference (SoCC) 2019, https://arxiv.org/abs/1804.05839

Analytics Zoo

Unified Data Analytics and AI Platform

Models
(Built-in models and algorithms)

ML Workflow
(Automate tasks for building end-to-end pipelines)

End-to-End Pipelines
(Automatically scale AI models to distributed Big Data)

Laptop | K8s Cluster | Hadoop Cluster | Cloud

Analytics Zoo

Unified Data Analytics and AI Platform

Models & Algorithms

- Recommendation
- Time Series
- Computer Vision
- NLP

ML Workflow

- AutoML
- Automatic Cluster Serving

End-to-end Pipelines

- Distributed TensorFlow & PyTorch on Spark
- RayOnSpark
- Spark Dataframes & ML Pipelines for DL
- InferenceModel

Compute Environment

- Laptop
- K8s Cluster
- Hadoop Cluster
- Cloud

- DL Frameworks (TF/PyTorch/OpenVINO/...)
- Distributed Analytics (Spark/Flink/Ray/...)
- Python Libraries (Numpy/Pandas/sklearn/...)

Powered by oneAPI

https://github.com/intel-analytics/analytics-zoo

Integrated Big Data Analytics and AI

Seamless Scaling from Laptop to Distributed Big Data

- Easily prototype **end-to-end pipelines** that apply AI models to big data
- **"Zero" code change** from laptop to distributed cluster
- Seamlessly deployed on **production Hadoop/K8s clusters**
- **Automate the process** of applying machine learning to big data

Getting Started

Getting Started with Analytics Zoo

- **Try Analytics Zoo on Google Colab**

- **Pull Analytics Zoo Docker image**

  sudo docker pull intelanalytics/analytics-zoo:latest

- **Install Analytics Zoo with pip**

  pip install analytics-zoo

https://colab.research.google.com/drive/1Ck-rcAYiI54ot0L9lU93Wglr2SMSYq27

Features

End-To-End Pipelines

Distributed TensorFlow/PyTorch on Spark in Analytics Zoo

Write TensorFlow/PyTorch inline with Spark code

```
#pyspark code
train_rdd = spark.hadoopFile(…).map(…)
dataset = TFDataset.from_rdd(train_rdd,…)

#tensorflow code
import tensorflow as tf
slim = tf.contrib.slim
images, labels = dataset.tensors
with slim.arg_scope(lenet.lenet_arg_scope()):
   logits, end_points = lenet.lenet(images, …)
loss = tf.reduce_mean( \
   tf.losses.sparse_softmax_cross_entropy( \
   logits=logits, labels=labels))

#distributed training on Spark
optimizer = TFOptimizer.from_loss(loss, Adam(…))
optimizer.optimize(end_trigger=MaxEpoch(5))
```

Analytics Zoo API in blue

Image Segmentation using TFPark

https://github.com/intel-analytics/zoo-tutorials/blob/master/tensorflow/notebooks/image_segmentation.ipynb

Face Generation Using Distributed PyTorch on Analytics Zoo

https://github.com/intel-analytics/analytics-zoo/blob/master/apps/pytorch/face_generation.ipynb

Spark Dataframe & ML Pipeline for DL

```
#Spark dataframe code
parquetfile = spark.read.parquet(…)
train_df = parquetfile.withColumn(…)

#Keras API
model = Sequential()
          .add(Convolution2D(32, 3, 3)) \
          .add(MaxPooling2D(pool_size=(2, 2))) \
          .add(Flatten()).add(Dense(10)))

#Spark ML pipeline code
estimater = NNEstimater(model, \
                        CrossEntropyCriterion())\
              .setMaxEpoch(5) \
              .setFeaturesCol("image")
nnModel = estimater.fit(train_df)
```

Analytics Zoo API in blue

Image Similarity using NNFrame

https://github.com/intel-analytics/analytics-zoo/blob/master/apps/image-similarity/image-similarity.ipynb

RayOnSpark

Run Ray programs directly on YARN/Spark/K8s cluster

```
sc = init_spark_on_yarn(...)
ray_ctx = RayContext(sc=sc, ...)
ray_ctx.init()

#Ray code
@ray.remote
class TestRay():
    def hostname(self):
        import socket
        return socket.gethostname()

actors = [TestRay.remote() for i in range(0, 100)]
print([ray.get(actor.hostname.remote()) \
      for actor in actors])

ray_ctx.stop()
```

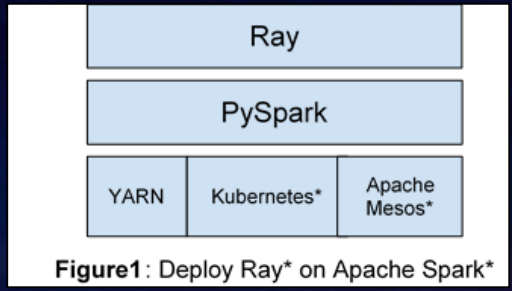
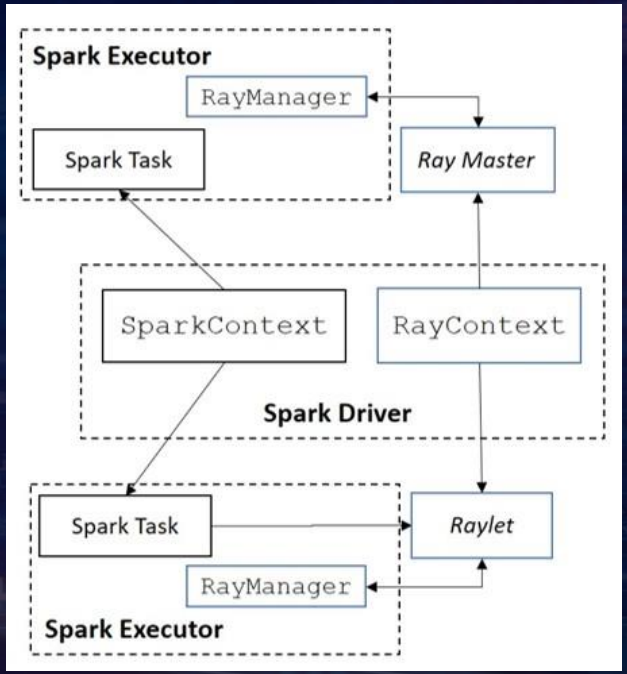


Figure1: Deploy Ray* on Apache Spark*



"RayOnSpark: Running Emerging AI Applications on Big Data Clusters with Ray and Analytics Zoo"
https://medium.com/riselab/rayonspark-running-emerging-ai-applications-on-big-data-clusters-with-ray-and-analytics-zoo-923e0136ed6a

Analytics Zoo API in blue

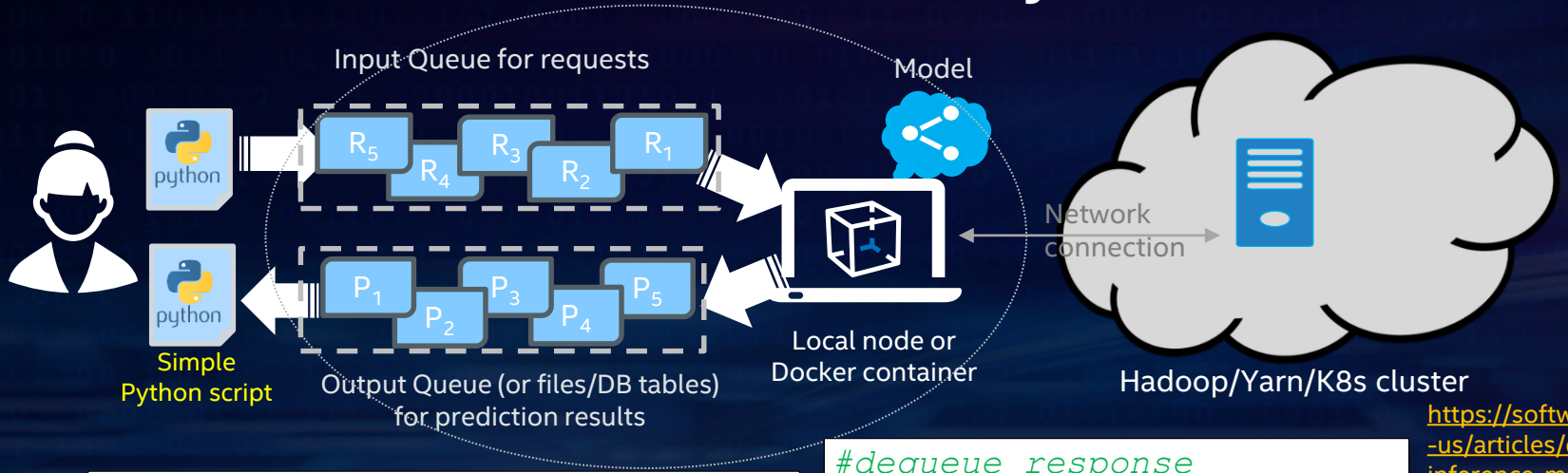
Sharded Parameter Server With RayOnSpark

https://github.com/intel-analytics/analytics-zoo/blob/master/apps/image-similarity/image-similarity.ipynb

Features

ML Workflow

Distributed Inference Made Easy with *Cluster Serving*



https://software.intel.com/en-us/articles/distributed-inference-made-easy-with-analytics-zoo-cluster-serving

```
#enqueue request
input = InputQueue()
img = cv2.imread(path)
img = cv2.resize(img, (224, 224))
input.enqueue_image(id, img)
```

```
#dequeue response
output = OutputQueue()
result = output.dequeue()
for k in result.keys():
  print(k + ": " + \
     json.loads(result[k]))
```

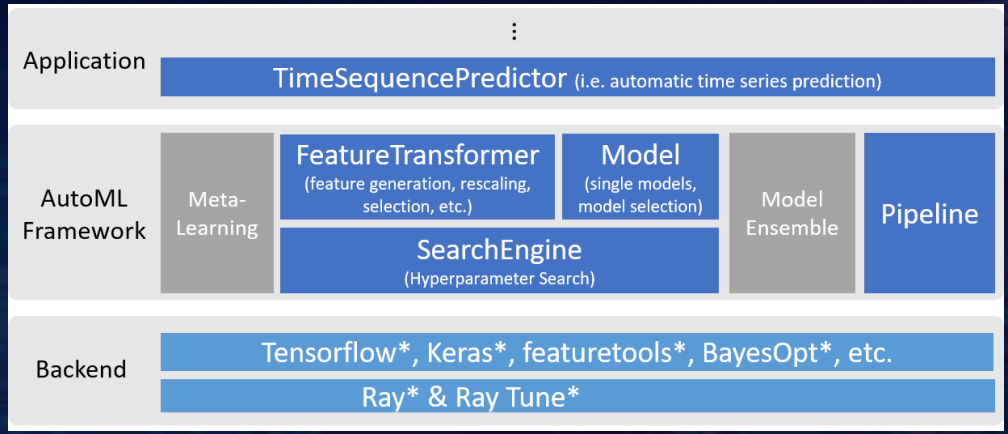
- √ **Users freed from complex distributed inference solutions**
- √ **Distributed, real-time inference automatically managed by Analytics Zoo**
  - ***TensorFlow, PyTorch, Caffe, BigDL, OpenVINO, …***
  - ***Spark Streaming, Flink, …***

Analytics Zoo API in blue

Scalable AutoML for Time Series Prediction

Automated feature selection, model selection and hyper parameter tuning using Ray

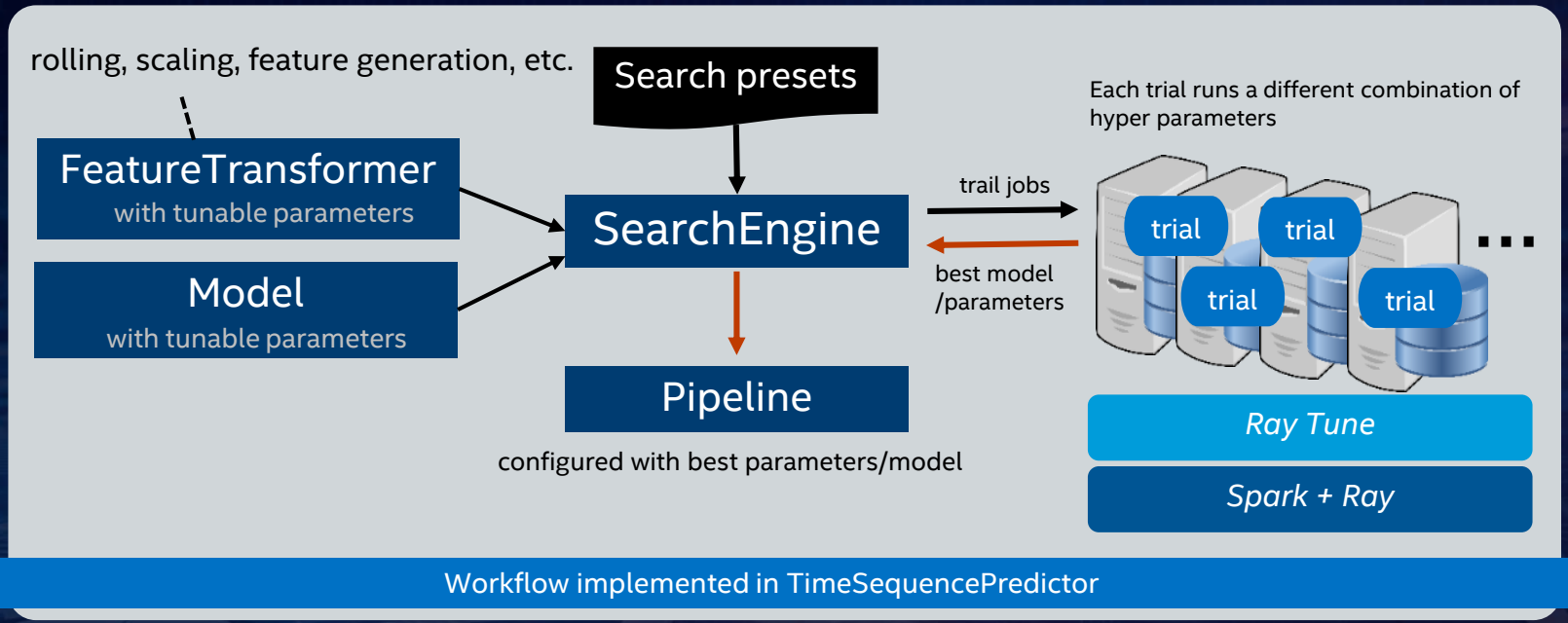
```
tsp = TimeSequencePredictor( \
            dt_col="datetime",
            target_col="value")

pipeline = tsp.fit(train_df,
               val_df, metric="mse",
               recipe=RandomRecipe())

pipeline.predict(test_df)
```



"Scalable AutoML for Time Series Prediction using Ray and Analytics Zoo"
https://medium.com/riselab/scalable-automl-for-time-series-prediction-using-ray-and-analytics-zoo-b79a6fd08139

Analytics Zoo API in blue

AutoML Training



AutoML Notebook

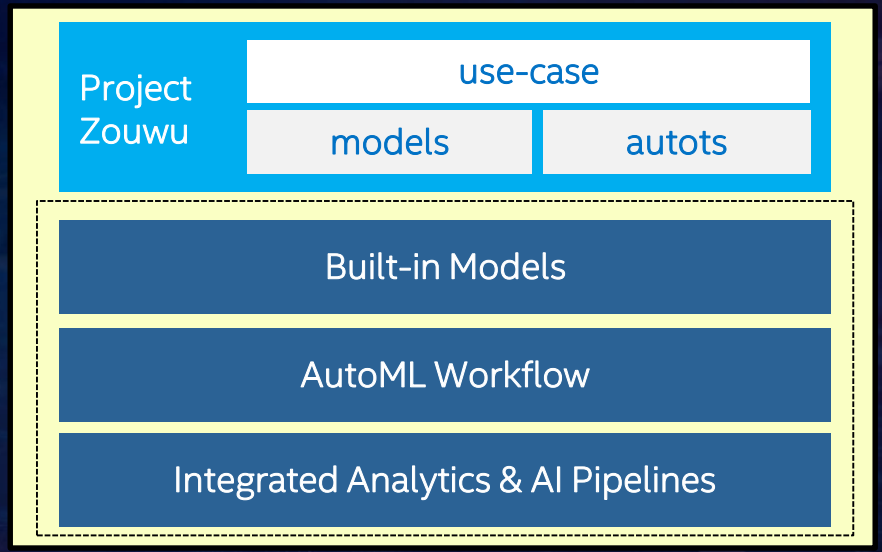
https://github.com/intel-analytics/analytics-zoo/blob/master/apps/automl/nyc_taxi_dataset.ipynb

Work in Progress

Project Zouwu: ANALYTICS ZOO Time Series for Telco

Project Zouwu

- ***Use case*** – reference time series use cases for Telco *(such as network traffic forecasting, etc.)*
- ***Models*** – built-in models for time series analysis *(such as LSTM, MTNet, DeepGlo)*
- ***AutoTS*** – AutoML support for building E2E time series analysis pipelines *(including automatic feature generation, model selection and hyperparameter tuning)*



https://github.com/intel-analytics/analytics-zoo/tree/master/pyzoo/zoo/zouwu

*Joint-collaborations with NPG

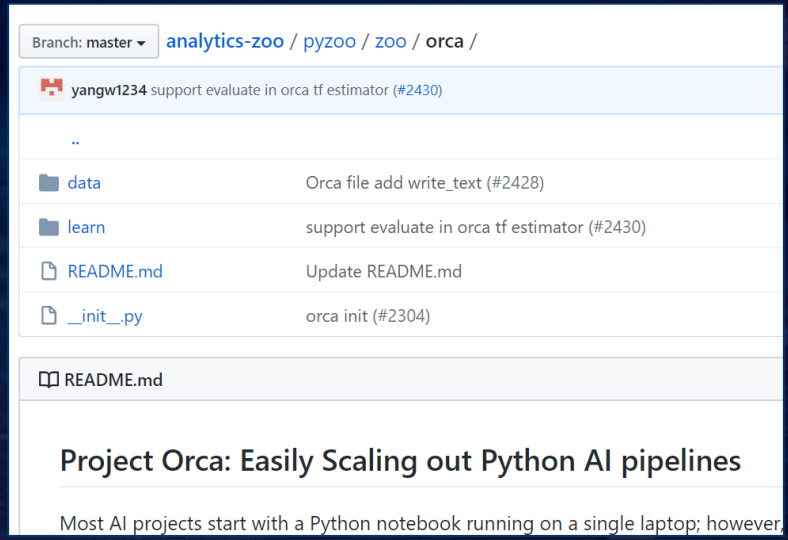
Network Traffic KPI Prediction using Zouwu

https://github.com/intel-analytics/analytics-zoo/blob/master/pyzoo/zoo/zouwu/use-case/network_traffic/network_traffic_autots_forecasting.ipynb

Project Orca: Easily Scaling Python AI pipeline on Analytics Zoo

Seamless scale Python notebook from laptop to distributed big data

- ***orca.data***: data-parallel pre-processing for (any) Python libs
  - pandas, numpy, sklearn, PIL, spacy, tensorflow Dataset, pytorch dataloader, spark, etc.
- ***orca.learn***: transparently distributed training for deep learning
  - sklearn style estimator for TensorFlow, PyTorch, Keras, Horovod, MXNet, etc.

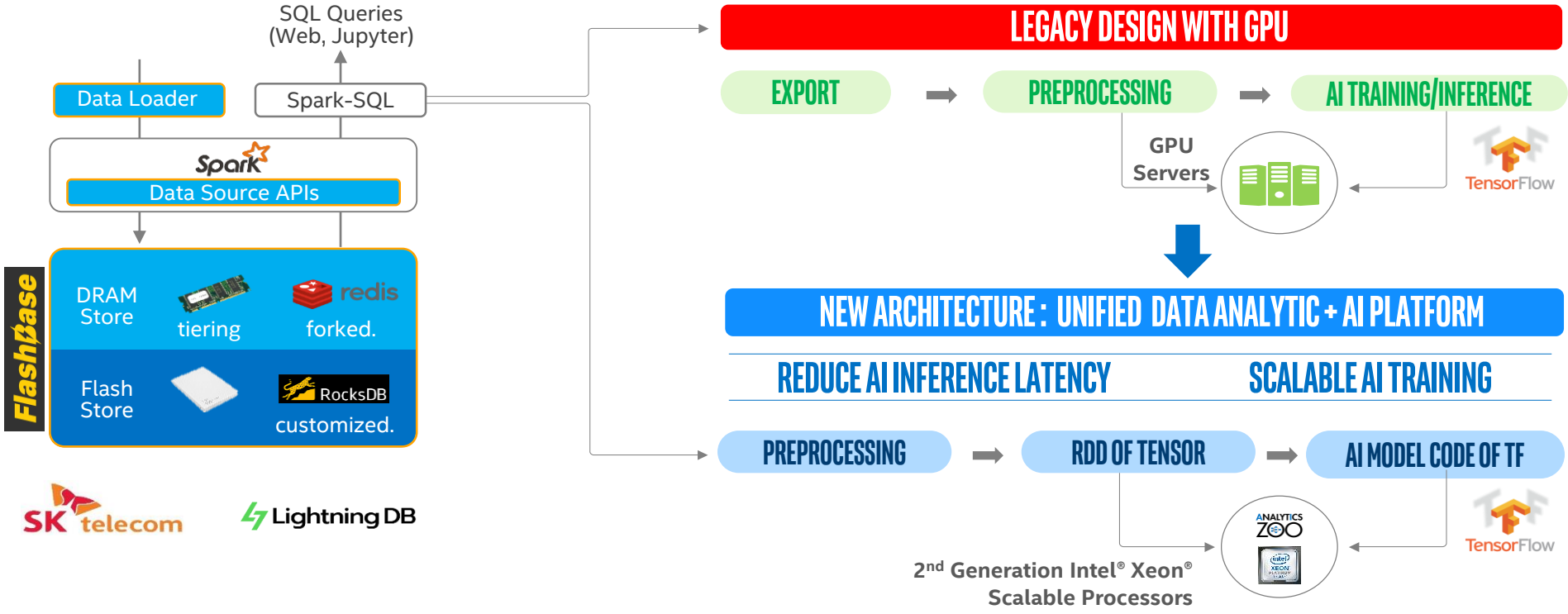


https://github.com/intel-analytics/analytics-zoo/tree/master/pyzoo/zoo/orca

Use Cases

Migrating from GPU in SK Telecom

Time Series Based Network Quality Prediction



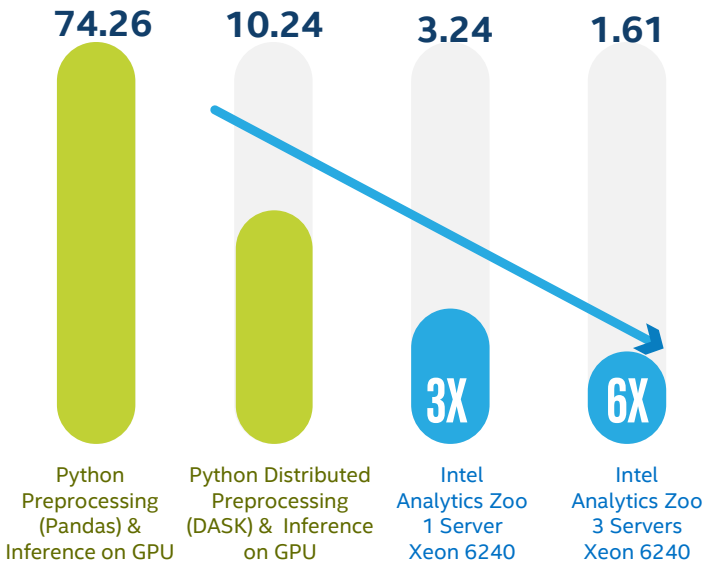
https://webinar.intel.com/AI_Monitoring_WebinarREG

Migrating from GPU in SK Telecom

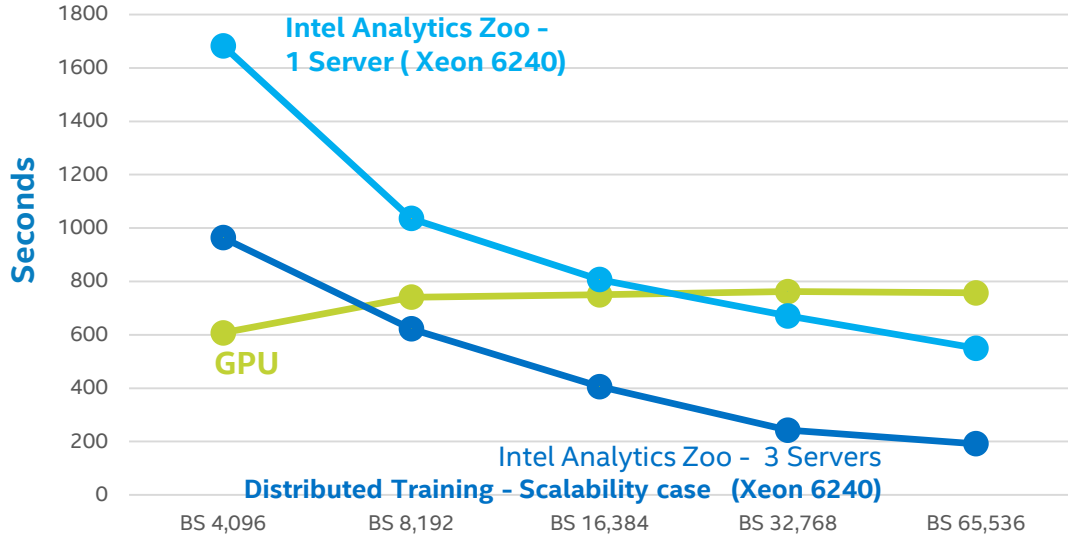
Time Series Based Network Quality Prediction

TCO OPTIMIZED AI PERFORMANCE WITH [1] ANALYTICS ZOO [2] INTEL OPTIMIZED TENSORFLOW [3] DISTRIBUTED AI PROCESSING

[1] PRE-PROCESSING & INFERENCE LATENCY



[2] TIME-TO-TRAINING PERFORMANCE



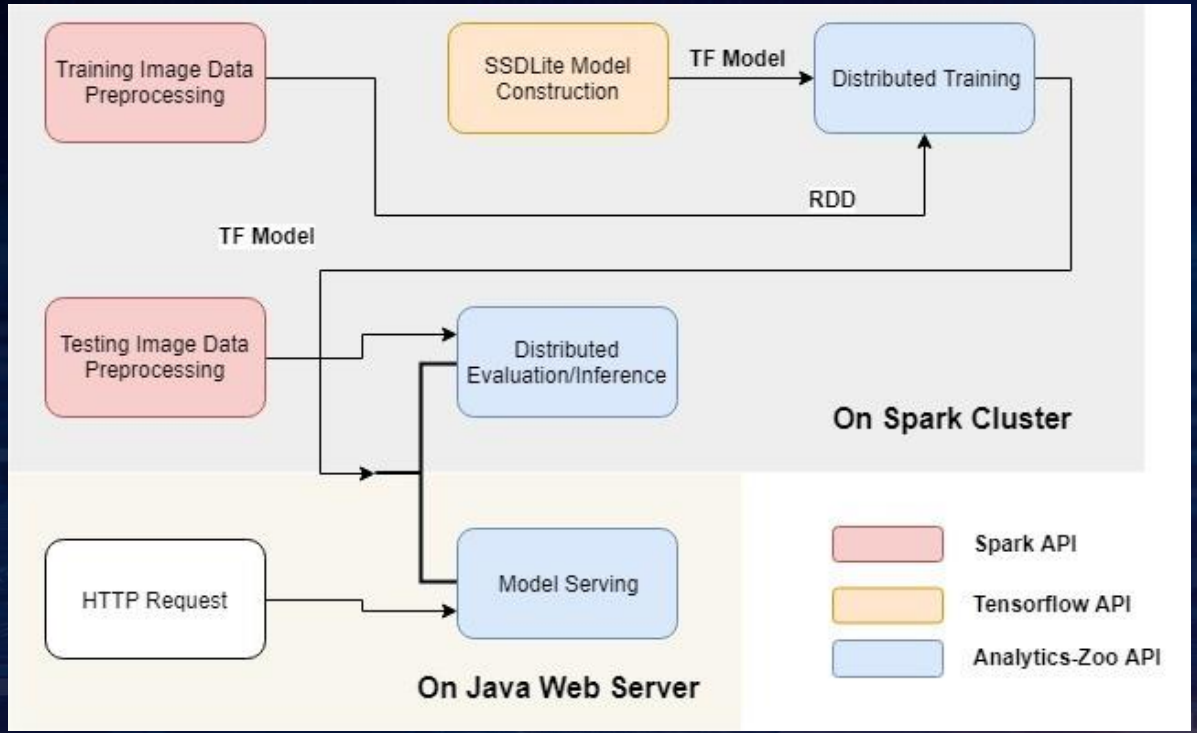
Test Data: **80K** Cell Tower, **8** days, **5**mins period, **8** Quality Indicator

https://webinar.intel.com/AI_Monitoring_WebinarREG

Performance test validation @ SK Telecom Testbed

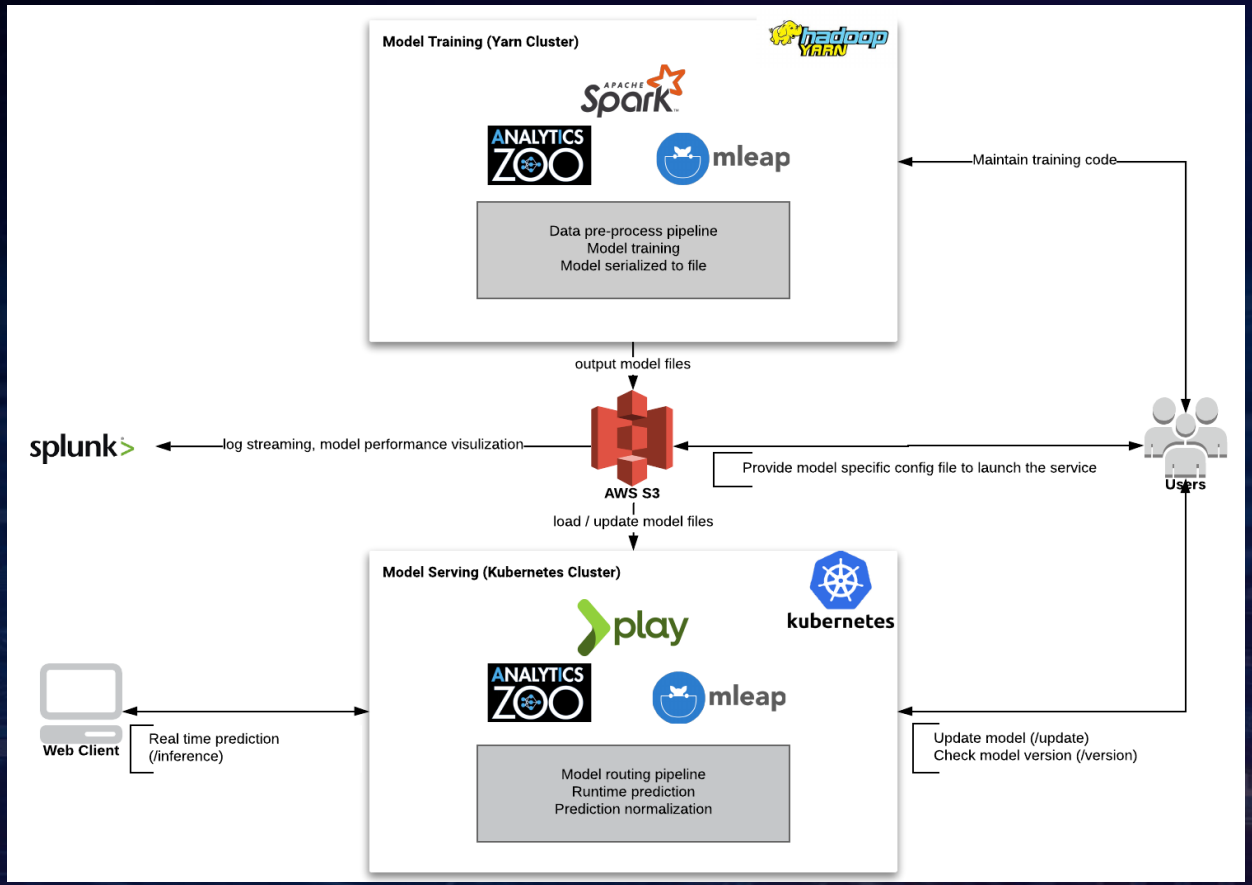
Edge to Cloud Architecture in Midea

Computer Vision Based Product Defect Detection



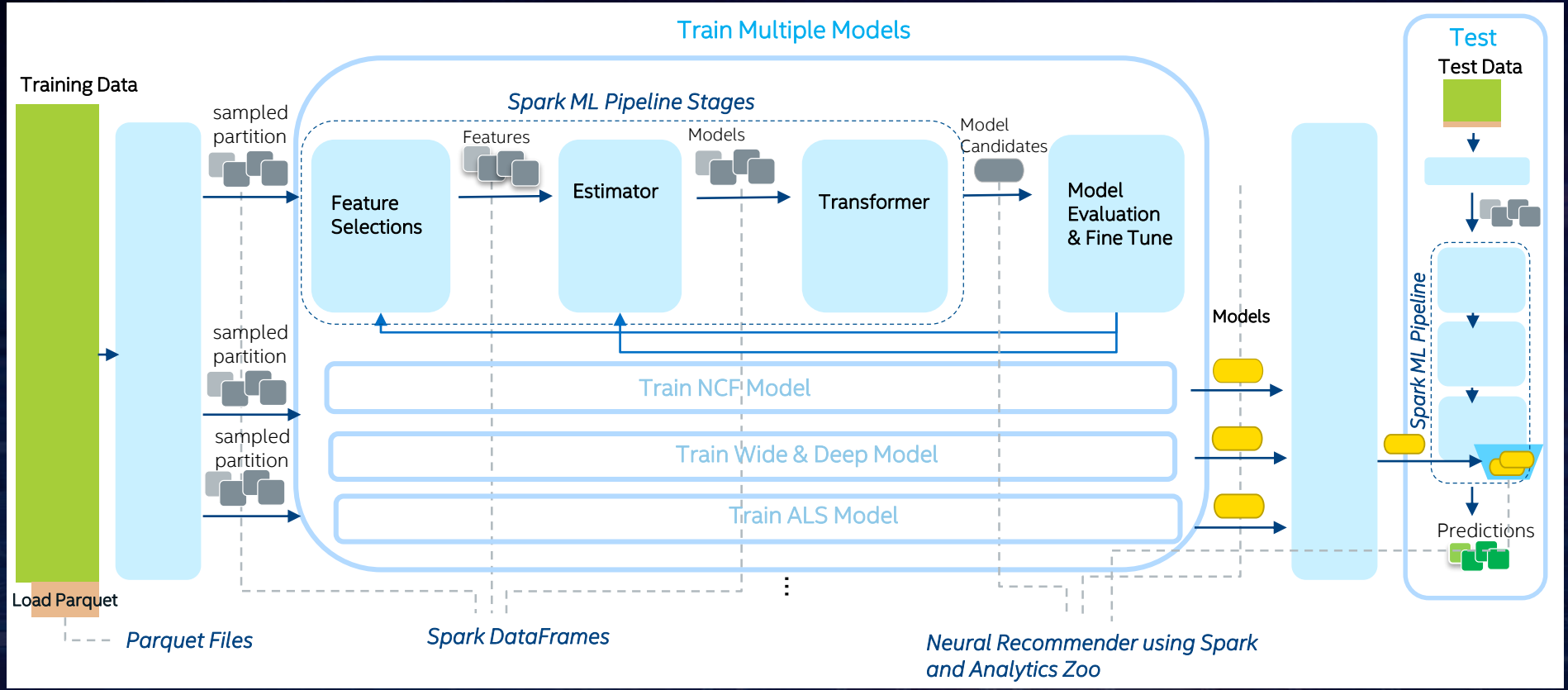
https://software.intel.com/en-us/articles/industrial-inspection-platform-in-midea-and-kuka-using-distributed-tensorflow-on-analytics

Product Recommendation on AWS in Office Depot



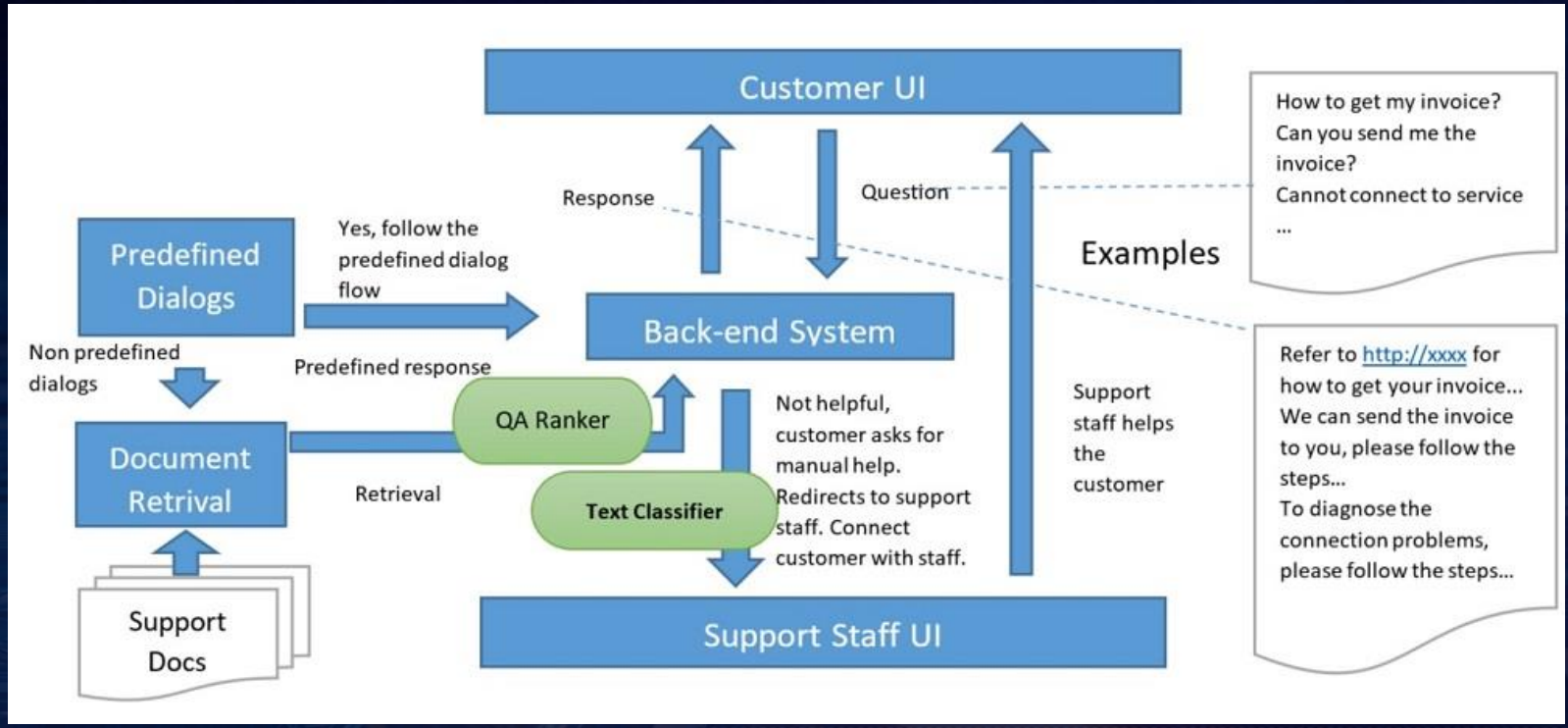
https://software.intel.com/en-us/articles/real-time-product-recommendations-for-office-depot-using-apache-spark-and-analytics-zoo-on

Recommender Service on Cloudera in MasterCard



https://software.intel.com/en-us/articles/deep-learning-with-analytic-zoo-optimizes-mastercard-recommender-ai-service

NLP Based Customer Service Chatbot for Microsoft Azure



https://software.intel.com/en-us/articles/use-analytics-zoo-to-inject-ai-into-customer-service-platforms-on-microsoft-azure-part-1
https://www.infoq.com/articles/analytics-zoo-qa-module/

And Many More

TECHNOLOGY

bluedata
cloudera
CRAY THE SUPERCOMPUTER COMPANY
databricks
DELL EMC
inspur
GIGASPACES innovate with confidence
Lightbend
Qubole

CLOUD SERVICE PROVIDERS

Alibaba Cloud aliyun.com
aws
Azure
Tencent 腾讯
Baidu 百度
IBM Cloud
KINGSOFT
Cloud Dataproc

END USERS

cdhi
Telefonica
中国电信 CHINA TELECOM
THE WORLD BANK
JD.COM 京东
CERN openlab
Midea
韵达 EXPRESS
CISCO
UnionPay 银联

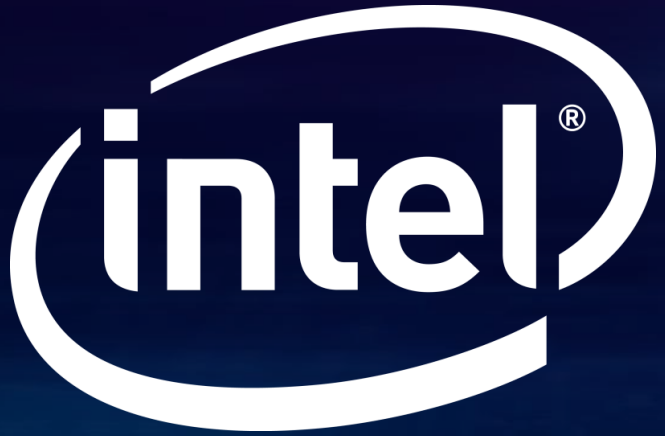
software.intel.com/data-analytics

Not a full list

*Other names and brands may be claimed as the property of others.

Summary

- **Github**
  - Project repo: *https://github.com/intel-analytics/analytics-zoo*
    https://github.com/intel-analytics/BigDL
  - Getting started: *https://analytics-zoo.github.io/master/#gettingstarted/*
- **Technical paper/tutorials**
  - CVPR 2018: *https://jason-dai.github.io/cvpr2018/*
  - AAAI 2019: *https://jason-dai.github.io/aaai2019/*
  - SoCC 2019: *https://arxiv.org/abs/1804.05839*
- **Use cases**
  - *Azure, CERN, MasterCard, Office Depot, Tencent, Midea, etc.*
  - *https://analytics-zoo.github.io/master/#powered-by/*


intel

Legal Notices and Disclaimers

- Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations, and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information visit intel.com/performance.
- Intel does not control or audit the design or implementation of third-party benchmark data or websites referenced in this document. Intel encourages all of its customers to visit the referenced websites or others where similar performance benchmark data are reported and confirm whether the referenced benchmark data are accurate and reflect performance of systems available for purchase.
- Optimization notice: Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.
- Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software, or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at intel.com/benchmarks.
- Intel, the Intel logo, Intel Inside, the Intel Inside logo, Intel Atom, Intel Core, Iris, Movidius, Myriad, Intel Nervana, OpenVINO, Intel Optane, Stratix, and Xeon are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries.
- *Other names and brands may be claimed as the property of others.
- © Intel Corporation