Building Unified Big Data Analytics and AI Pipelines

Jason Dai
Senior Principal Engineer
Overview
Accelerating Data Analytics + AI Solutions At Scale

**BigDL**
Distributed, High-Performance Deep Learning Framework for Apache Spark*

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

**Analytics Zoo**
Analytics + AI Platform
Distributed TensorFlow*, Keras*, PyTorch* and BigDL on Apache Spark*

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

*Other names and brands may be claimed as the property of others.
Real-World ML/DL Applications Are Complex Data Analytics Pipelines

Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

End-to-End Big Data Analytics and AI Pipeline

Seamless Scaling from Laptop to Production with

- Prototype on laptop using sample data
- Experiment on clusters with history data
- Production deployment w/ distributed data pipeline

• “Zero” code change from laptop to distributed cluster
• Directly access production data (Hadoop/Hive/HBase) without data copy
• Easily prototype the end-to-end pipeline
• Seamlessly deployed on production big data clusters
BigDL
Bringing Deep Learning To Big Data Platform

- **Distributed** deep learning framework for Apache Spark
- Make deep learning more accessible to **big data users** and **data scientists**
  - Write deep learning applications as **standard Spark programs**
  - Run on existing Spark/Hadoop clusters (**no changes needed**)
- Feature parity with popular deep learning frameworks
  - E.g., Caffe, Torch, Tensorflow, etc.
- High performance (on CPU)
  - Powered by Intel MKL and multi-threaded programming
- Efficient scale-out
  - Leveraging Spark for distributed training & inference

https://github.com/intel-analytics/BigDL
https://bigdl-project.github.io/
### Analytics Zoo

**End-to-End, Unified Analytics + AI Platform for Big Data**

<table>
<thead>
<tr>
<th>Use case</th>
<th>Recommendation</th>
<th>Anomaly Detection</th>
<th>Text Classification</th>
<th>Text Matching</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>Image Classification</td>
<td>Object Detection</td>
<td>Seq2Seq</td>
<td>Transformer</td>
</tr>
<tr>
<td>Feature Engineering</td>
<td>image</td>
<td>3D image</td>
<td>text</td>
<td>Time series</td>
</tr>
<tr>
<td>High Level Pipelines</td>
<td>tfpark: Distributed TF on Spark</td>
<td>Distributed Keras w/ autograd on Spark</td>
<td>nnframes: Spark Dataframes &amp; ML Pipelines for Deep Learning</td>
<td>Distributed Model Serving (batch, streaming &amp; online)</td>
</tr>
<tr>
<td>Backend/Library</td>
<td>TensorFlow</td>
<td>Keras</td>
<td>BigDL</td>
<td>NLP Architect</td>
</tr>
<tr>
<td></td>
<td>MKLDNN</td>
<td>OpenVINO</td>
<td>Intel® Optane™ DCPMM</td>
<td>DL Boost (VNNI)</td>
</tr>
</tbody>
</table>

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

End-to-End, Unified Analytics + AI Platform for Big Data

Build end-to-end deep learning applications for big data
• Distributed *TensorFlow* on Spark
• *Keras* API (with autograd & transfer learning support) on Spark
• *nnframes*: native DL support for Spark DataFrames and ML Pipelines

Productionize deep learning applications for big data at scale
• Plain Java/Python *model serving* APIs (w/ OpenVINO support)
• Support Web Services, Spark, Flink, Storm, Kafka, etc.

Out-of-the-box solutions
• Built-in deep learning *models, feature engineering* operations, and reference *use cases*
Distributed TF & Keras on Spark

- Data wrangling and analysis using PySpark
- Deep learning model development using TensorFlow or Keras
- Distributed training / inference on Spark

Write TensorFlow code inline in PySpark program

```python
# 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))
```
#Spark dataframe transformations
parquetfile = spark.read.parquet(...)
train_df = parquetfile.withColumn(...)

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

#Spark ML pipeline
Estimater = NNEstimater(model, CrossEntropyCriterion()) \
    .setLearningRate(0.003).setBatchSize(40).setMaxEpoch(5) \
    .setFeaturesCol("image")
nnModel = estimater.fit(train_df)
#Spark dataframe transformations
parquetfile = spark.read.parquet(…)

\textbf{train\_df} = parquetfile.withColumn(…)

#Keras API
model = Sequential()

\begin{verbatim}
    .add(Convolution2D(32, 3, 3, activation='relu', input_shape=…)) \\
    .add(MaxPooling2D(pool_size=(2, 2))) \\
    .add(Flatten()).add(Dense(10, activation='softmax'))
\end{verbatim}

#Spark ML pipeline
Estimater = NNEstimator(model, CrossEntropyCriterion()) \ 
    .setLearningRate(0.003).setBatchSize(40).setMaxEpoch(5) \ 
    .setFeaturesCol("image")

nnModel = estimater.\textbf{fit}(\textbf{train\_df})
Distributed model serving in **Web Service, Flink, Kafka, Storm, etc.**

- Plain Java or Python API, with OpenVINO and DL Boost (VNNI) support
from zoo.common.nncontext import init_nncontext
from zoo.feature.image import ImageSet
from zoo.pipeline.inference import InferenceModel

sc = init_nncontext("OpenVINO Object Detection Inference Example")
images = ImageSet.read(options.img_path, sc,
    resize_height=600, resize_width=600).get_image().collect()
input_data = np.concatenate([image.reshape((1, 1) + image.shape) for image in images], axis=0)

model = InferenceModel()
model.load_tf(options.model_path, backend="openvino", model_type=options.model_type)
predictions = model.predict(input_data)

# Print the detection result of the first image.
print(predictions[0])

Transparently support OpenVINO in model serving, which deliver a significant boost for inference speed
Upcoming Analytics Zoo 0.6 Release

• Distributed PyTorch on Spark

• Ray on Spark
  • Run Ray programs directly on standard Hadoop/YARN clusters

• AutoML support
  • Automatic feature generation, model selection and hyper-parameter tuning for time series prediction

• Cluster serving
  • Distributed, real-time (streaming) model serving with simple pub-sub interface
Use Cases
Object Detection and Image Feature Extraction at JD.com

- Reuse existing Hadoop/Spark clusters for deep learning with no changes (image search, IP protection, etc.)
- Efficiently scale out on Spark with superior performance (3.83x speed-up vs. GPU servers) as benchmarked by JD

http://mp.weixin.qq.com/s/xUckzbHK4K06-v5qUsaNQQ
NLP Based Customer Service Chatbot for Microsoft Azure

https://www.infoq.com/articles/analytics-zoo-qa-module/
Product Recommendations in Office Depot

Computer Vision Based Product Defect Detection in Midea

Training Image Data Preprocessing → SSDLite Model Construction → TF Model → Distributed Training → RDD

Testing Image Data Preprocessing → Distributed Evaluation/Inference

HTTP Request → Model Serving

On Spark Cluster

On Java Web Server

Spark API
Tensorflow API
Analytics-Zoo API

Recommender AI Service in MasterCard

Particle Classifier for High Energy Physics in CERN

Deep learning pipeline for physics data

Model serving using Apache Kafka and Spark

Data Pipeline

Data Ingestion → Feature Preparation → Model Development → Training

- Read physics data and feature engineering
- Prepare input for Deep Learning network
- 1. Specify model topology
   2. Tune model topology on small dataset
- Train the best model

Leveraging Apache Spark and Analytics Zoo in Python Notebooks

Unsupervised Time Series Anomaly Detection for **Baosight**

## Technology
- bluedata
- cloudera
- CRAY
- databricks
- DELL EMC
- GIGASPACE
- Lightbend
- Qu bole

## Cloud Service Providers
- Alibaba Cloud
- AWS
- Azure
- Tencent
- Baidu
- IBM Cloud
- KINGSOFT

## End Users
- cdhi
- Telefonica
- 中国电信
- 中国
- Tencent
- Baidu
- JD.COM
- Midea
- CISCO
- UnionPay

[software.intel.com/AlonBigData](http://software.intel.com/AlonBigData)

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

- Analytics Zoo repo: https://github.com/intel-analytics/analytics-zoo/
- More presentations: https://analytics-zoo.github.io/master/#presentations/
End-to-End Big Data and AI Pipelines
Seamless Scaling from Laptop to Production

Unified Analytics + AI Platform
Distributed TensorFlow*, Keras*, PyTorch* & BigDL on Apache Spark*
https://github.com/intel-analytics/analytics-zoo

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

• Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at intel.com, or from the OEM or retailer.

• No computer system can be absolutely secure.

• Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit http://www.intel.com/performance.

Intel, the Intel logo, Xeon, Xeon phi, Lake Crest, etc. are trademarks of Intel Corporation in the U.S. and/or other countries.
*Other names and brands may be claimed as the property of others.
© 2019 Intel Corporation