Scalable AutoML for Time Series Forecasting using Ray

Shengsheng Huang, Jason Dai

Intel Corporation
Outline

• Background
• Scalable AutoML for Time Series on Ray
• Use Case Sharing & Learnings
Outline

• **Background**
  - analytics-zoo, time series, automl, rayonspark

• Scalable AutoML for Time Series on Ray

• Use Case Sharing & Learnings
AI on BIG DATA

BigDL

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

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

Analytical Zoo

Unified Analytics + AI Platform for TensorFlow*, PyTorch*, Keras*, BigDL, OpenVINO, Ray* and Apache Spark*

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

Seamless Scaling from Laptop to Distributed Big Data

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

Jul 9, 2020
## Analytics Zoo
Unified Data Analytics and AI Platform

<table>
<thead>
<tr>
<th>Models &amp; Algorithms</th>
<th>Recommendation</th>
<th>Time Series</th>
<th>Computer Vision</th>
<th>NLP</th>
</tr>
</thead>
<tbody>
<tr>
<td>ML Workflow</td>
<td>AutoML</td>
<td>Automatic Cluster Serving</td>
<td></td>
<td></td>
</tr>
<tr>
<td>End-to-end Pipelines</td>
<td>Distributed TensorFlow &amp; PyTorch on Spark</td>
<td>RayOnSpark</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Spark Dataframes &amp; ML Pipelines for DL</td>
<td>InferenceModel</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**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
Time Series Forecasting

• Problem Definition
  • Given all history observations \( y_1, \ldots, y_t \), Predict
  values of next \( h \) steps, \( y_{t+1}, \ldots, y_{t+h} \)
  • Usually only lookback \( k \) steps, \( y_{t-k+1}, \ldots, y_t \)

• Applications
  • demand prediction, network quality analysis,
    predictive maintenance, AIOps

• Forecasting Methods
  • Autoregression, Exponential Smoothing, ARIMA, ...
  • Machine Learning and Deep Learning methods
AutoML

RayOnSpark: Motivations

• **Ray** : A distributed framework for emerging AI applications
  • Ray Tune, Ray Rllib, RaySGD, etc.
  • [https://github.com/ray-project/ray/](https://github.com/ray-project/ray/)

• **RayOnSpark**
  • Directly run Ray programs on big data cluster
  • Integrate Ray into Spark pipeline
RayOnSpark: Architecture

- Ray provision with Spark
- Data Sharing between Ray & Spark

https://ray.readthedocs.io/en/latest/
RayOnSpark: How to Use

- Init SparkContext
- **Init RayContext**
- **... run ray programs**
- Stop RayContext
- Stop SparkContext

```python
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()
```

"RayOnSpark: Running Emerging AI Applications on Big Data Clusters with Ray and Analytics Zoo"
Outline

• Background

• **Scalable AutoML for Time Series on Ray**
  • architecture, execution flow, apis, etc.

• Use Case Sharing & Learnings
Time Series Solution In Analytics Zoo

Rich models and algorithms
(neural-networks, hybrid, state-of-art)

AutoML
(automatic feature generation, model selection, hyper-parameter tuning, etc.)

Seamless scaling
(with integrated analytics and AI pipelines)
**Software Stack**

- **AutoML Framework**
  - FeatureTransformer
  - Model
  - SearchEngine
  - Pipeline
- **Time Series upon AutoML**
  - TimeSequencePredictor
  - TimeSequencePipeline

---

### Application

- TimeSequencePredictor (i.e. automatic time series prediction)

### AutoML Framework

<table>
<thead>
<tr>
<th>Meta-Learning</th>
<th>FeatureTransformer (feature generation, rescaling, selection, etc.)</th>
<th>Model (single models, model selection)</th>
<th>Model Ensemble</th>
<th>Pipeline</th>
</tr>
</thead>
<tbody>
<tr>
<td>SearchEngine</td>
<td>(Hyperparameter Search)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Backend

- Tensorflow*, Keras*, featuretools*, BayesOpt*, etc.
- Ray* & Ray Tune*
- RayOnSpark
- Apache Spark*

---

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

Training at Runtime

Workflow implemented in TimeSequencePredictor
A glimpse of the APIs

• **Training a Pipeline**
  • `fit` (w/ `automl`)
  • `recipe`
  • `distributed`

• **Using a Pipeline**
  • `save/load`
  • `evaluate/predict`
  • `fit` (incremental)
**Project Zouwu**: application framework for building E2E time series analysis

- **Use case** - reference time series use cases
- **Models** - built-in models for time series analysis
  - **AutoTS** - AutoML support for building E2E time series analysis pipelines

[GitHub Link](https://github.com/intel-analytics/analytics-zoo/tree/master/pyzoo/zoo/zouwu)
Outline

• Background
• Scalable AutoML for Time Series on Ray
• Use Case Sharing & Learnings
Use Case Description

Use aggregated traffic KPI's (i.e. total bytes, average rate in Mbps/Gbps) in the past week to forecast the KPI in the next two hours.

2 ways to solve this problem using Zouwu

• Use built-in “Forecaster” models for training, and forecasting (example notebook link)

• Use “AutoTS” (with built-in AutoML support) to train an E2E Time Series Analysis Pipeline, and forecast (example notebook link)

Example result of network traffic average rate forecasting on the test period
Time Series Based Network Quality Prediction in SK Telecom

Data Loading
- File
- HTTP
- Kafka

Data Source APIs
- Spark-SQL

Preprocess
- RDD of Tensor

Model Code of TF
- Model

DL Training & Inferencing
- SIMD Acceleration

FlashDB Store
- DRAM
- Flash Store
- tiering
- redis forked
- RocksDB customized

Forecast-based Analysis for AIOps at Neusoft

https://platform.neusoft.com/2020/01/17/xw-intel.html
https://platform.neusoft.com/2017/03/04/qt-baomaqiche.html
Takeaways from Early Users

• **Highlights**
  - Additional features allowed
  - Less efforts in tuning
  - Satisfactory accuracy

• **Data quality matters**

• **Scale, scale, scale**
  - hundreds of thousands of cells X N KPIs
  - millions of servers/containers X N metrics
Future Work

- Handle massive time series
- Search algo, meta-learning, ensemble, ...
- More models, features, ...
- Automatic data preprocessing (e.g. missing data & outliers)
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