S91030 - Hybrid Machine Learning with the Kubeflow Pipelines and RAPIDS

Sina Chavoshi
Technical Program Manager
Cloud AI Strategy:
The right approach for the right problem
Cloud AI Strategy:
The right approach for the right problem

Building blocks

Platform

Solutions

Google Cloud
### Building Blocks

#### Sight
- **Cloud Vision API**: Image recognition and classification.
- **Cloud Video Intelligence API**: Scene-level video annotation.
- **AutoML Vision**<sup>BETA</sup>: Custom image classification models.

#### Language
- **Cloud Translation API**: Language detection and translation.
- **Cloud Natural Language API**: Text parsing and analysis.
- **AutoML Translation**<sup>BETA</sup>: Custom domain-specific translation.
- **AutoML Natural Language**<sup>BETA</sup>: Custom text classification models.

#### Conversation
- **Dialogflow Enterprise Edition**: Build conversational interfaces.
- **Cloud Text-to-Speech API**: Convert text to speech.
- **Cloud Speech-to-Text API**: Convert speech to text.

---

**Google Cloud**
Cloud AI Strategy:
The right approach for the right problem
Cloud AI Strategy:

The right approach for the right problem
Cloud AI Platform

Data pipeline
- Cloud Dataprep
- BigQuery
- Cloud Dataflow
- Cloud Dataproc
- Spark

Model development
- Cloud ML Engine
- TensorFlow
- K
- learn
- XGBoost

Model deployment and management
- Cloud ML Engine
- Cloud Kubernetes Engine
- TensorFlow
- K
- learn
- Spark
- XGBoost
- Kubeflow

Tools
- Jupyter Notebooks

Services
- ASL

Community
- kaggle

Google Cloud
Building & deploying real-life ML applications is **hard** and **costly** because of lack of tooling that covers **end-to-end ML development & deployment**.
In addition to the actual ML...
You have to worry about so much more.
AI problems today

Problems

Deployment
Brittle, opinionated infrastructure that is hard to productionize and breaks between cloud and on-prem

Talent
Machine Learning expertise is scarce

Collaboration
Difficult to find, leverage existing solutions

Solutions

01 Kubeflow

02 Reusable pipelines

03 Google Cloud AI Hub
01: Kubeflow

Scalable ML services on Kubernetes

Easy to get started
• Out-of-box support for top frameworks
  – pytorch, caffe, tf and xgboost
• Kubernetes manages dependencies, resources

Swappable & scalable
• Library of ML services
• GPU support
• Massive scale

Meet customer where they are
• GCP
• On-prem with Cisco
THE BIG PROBLEM IN DATA SCIENCE

Slow Training Times for Data Scientists
RAPIDS — OPEN GPU DATA SCIENCE
Software Stack Python
**BENCHMARKS**

**cuIO/cuDF — Load and Data Preparation**
- 20 CPU Nodes: 2741 seconds
- 30 CPU Nodes: 1675 seconds
- 50 CPU Nodes: 715 seconds
- 100 CPU Nodes: 379 seconds
- DGX-2: 42 seconds
- 5x DGX-1: 19 seconds

**cuML — XGBoost**
- 20 CPU Nodes: 2290 seconds
- 30 CPU Nodes: 1956 seconds
- 50 CPU Nodes: 1999 seconds
- 100 CPU Nodes: 1948 seconds
- DGX-2: 169 seconds
- 5x DGX-1: 157 seconds

**End-to-End**
- 20 CPU Nodes: 2290 seconds
- 30 CPU Nodes: 1956 seconds
- 50 CPU Nodes: 1999 seconds
- 100 CPU Nodes: 1948 seconds
- DGX-2: 169 seconds
- 5x DGX-1: 157 seconds

---

**Benchmark**
200GB CSV dataset; Data preparation includes joins, variable transformations.

**CPU Cluster Configuration**
- CPU nodes (61 GiB of memory, 8 vCPUs, 64-bit platform), Apache Spark

**DGX Cluster Configuration**
- 5x DGX-1 on InfiniBand network
AI Hub & Pipelines: Fast & simple adoption of AI

1. Search & Discover
   Find best-of-breed solutions on the AI Hub which leverage Cloud AI solutions.

2. Deploy
   Quick 1-click implementation of ML pipelines onto Google Cloud Platform.

3. Customize
   Experiment and adjustment out-of-the-box pipelines to custom use cases.

4. Run in production
   Deploy customized pipelines in production.

5. Publish
   Upload & share pipelines running best within your org or publicly.

The Flywheel of AI Adoption

Network effect
02: Reusable Pipelines

Enable developers to build custom ML applications by easily “stitching” and connecting various components.

- Reuse instead of reimplement or reinvent
- Discover, learn and replicate successful pipelines
What constitutes a Kubeflow Pipeline

- Containerized implementations of ML Tasks
  - Containers provide portability, repeatability and encapsulation
  - A task can be single node or *distributed*
  - A containerized task can invoke other services

- Specification of the sequence of steps
  - Specified via Python SDK

- Input Parameters
  - A “Job” = Pipeline invoked w/ specific parameters
03: AI Hub at a glance

1. **All AI content in one place**
   Quick discovery of plug & play AI pipelines & other content built by teams across Google and by partners and customers.

2. **Fast & simple implementation of AI on GCP**
   One-click deployment of AI pipelines via Kubeflow on GCP as the go-to platform for AI + hybrid & on premise.

3. **Enterprise-grade internal & external sharing**
   Foster reuse by sharing deployable AI pipelines & other content privately within organizations & publicly.
Mission

The one place for everything AI, from experimentation to production.
Public and private AI Hub

Public content

- By Google
  Unique AI assets by Google
  AutoML, TPUs, kaggle, Cloud AI Platform, etc.
  Research at Google
  DeepMind

By partners
Created, shared & monetized by anyone

By customers
Content shared securely within and with other organizations

+ Private content
Kubeflow Pipelines enable workflow orchestration, rapid reliable experimentation, and share, re-use & compose.
Demo
Visual depiction of pipeline topology
<table>
<thead>
<tr>
<th>Experiment name</th>
<th>Last 5 runs</th>
<th>Created on</th>
<th>Created by</th>
</tr>
</thead>
<tbody>
<tr>
<td>tfma-experiment</td>
<td></td>
<td>6:17 PM, Aug 24, 2018</td>
<td>John Doe</td>
</tr>
<tr>
<td>xgboost-train</td>
<td></td>
<td>6:17 PM, Aug 24, 2018</td>
<td>John Doe</td>
</tr>
<tr>
<td>promo-email</td>
<td></td>
<td>6:17 PM, Aug 24, 2018</td>
<td>Walter Fisher</td>
</tr>
<tr>
<td>data-prep</td>
<td></td>
<td>6:17 PM, Aug 24, 2018</td>
<td>Walter Fisher</td>
</tr>
<tr>
<td>tf-preprocessing</td>
<td></td>
<td>6:17 PM, Aug 24, 2018</td>
<td>John Doe</td>
</tr>
<tr>
<td>tf-training</td>
<td></td>
<td>6:17 PM, Aug 24, 2018</td>
<td>Walter Fisher</td>
</tr>
<tr>
<td>tf-serving</td>
<td></td>
<td>6:17 PM, Aug 24, 2018</td>
<td>Walter Fisher</td>
</tr>
</tbody>
</table>

View all current and historical runs, grouped as “Experiments”
Rich visualizations of metrics
Clone an existing pipeline
Access to all config params, inputs and outputs for each run

<table>
<thead>
<tr>
<th>Run details</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>Succeeded</td>
</tr>
<tr>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>Created at</td>
<td>11/25/2018, 12:56:44 PM</td>
</tr>
<tr>
<td>Started at</td>
<td>11/25/2018, 12:56:44 PM</td>
</tr>
<tr>
<td>Finished at</td>
<td>11/25/2018, 1:16:37 PM</td>
</tr>
<tr>
<td>Duration</td>
<td>0:19:53</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Run parameters</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>output</td>
<td>gs://mlpipelines</td>
</tr>
<tr>
<td>project</td>
<td>foo2thebar</td>
</tr>
<tr>
<td>region</td>
<td>us-central1</td>
</tr>
<tr>
<td>train-data</td>
<td>gs://ml-pipeline-playground/sfpd/train.csv</td>
</tr>
<tr>
<td>eval-data</td>
<td>gs://ml-pipeline-playground/sfpd/eval.csv</td>
</tr>
<tr>
<td>schema</td>
<td>gs://ml-pipeline-playground/sfpd/schema.json</td>
</tr>
<tr>
<td>target</td>
<td>resolution</td>
</tr>
<tr>
<td>true-label</td>
<td>ACTION</td>
</tr>
</tbody>
</table>
Update parameters and submit
## All runs

<table>
<thead>
<tr>
<th>Runs</th>
<th>Status</th>
<th>Duration</th>
<th>Pipeline</th>
<th>Recurring run config</th>
<th>Start time</th>
<th>rmse</th>
<th>eta</th>
</tr>
</thead>
<tbody>
<tr>
<td>ccard-recommender-run3</td>
<td>✔️</td>
<td>1m 59s</td>
<td>linear-classifier</td>
<td></td>
<td>9:32 AM, Aug 26, 2018</td>
<td>0.88</td>
<td>0.92</td>
</tr>
<tr>
<td>ccard-recommender-run2-clone(2)</td>
<td>✔️</td>
<td>2m 12s</td>
<td>linear-classifier</td>
<td></td>
<td>11:42 AM, Aug 25, 2018</td>
<td>0.72</td>
<td>0.86</td>
</tr>
<tr>
<td>ccard-recommender-run2-clone(1)</td>
<td>✔️</td>
<td>2m 44s</td>
<td>linear-classifier</td>
<td></td>
<td>10:48 AM, Aug 25, 2018</td>
<td>0.74</td>
<td>0.84</td>
</tr>
<tr>
<td>ccard-recommender-run2</td>
<td>✔️</td>
<td>2m 18s</td>
<td>linear-classifier</td>
<td></td>
<td>10:22 PM, Aug 25, 2018</td>
<td>0.82</td>
<td>0.76</td>
</tr>
<tr>
<td>ccard-recommender-run1-clone(1)</td>
<td>✔️</td>
<td>2m 20s</td>
<td>linear-classifier</td>
<td></td>
<td>10:10 AM, Aug 24, 2018</td>
<td>0.80</td>
<td>0.84</td>
</tr>
<tr>
<td>ccard-recommender-run1</td>
<td>✔️</td>
<td>3m 20s</td>
<td>linear-classifier</td>
<td></td>
<td>6:17 PM, Aug 24, 2018</td>
<td>0.72</td>
<td>0.76</td>
</tr>
</tbody>
</table>

---

**Easy comparison of Runs**
# Experiments

## image-classifier

<table>
<thead>
<tr>
<th>Fastest run time</th>
<th>Slowest run time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1m 59s</td>
<td>3m 20s</td>
</tr>
</tbody>
</table>

### All runs

Filter runs

<table>
<thead>
<tr>
<th>Runs</th>
<th>Status</th>
<th>Duration</th>
<th>Pipeline</th>
<th>Recurring run config</th>
<th>Start time</th>
<th>rmse</th>
<th>eta</th>
</tr>
</thead>
<tbody>
<tr>
<td>ccard-recommender-run3</td>
<td>✔️</td>
<td>1m 59s</td>
<td>linear-classifier</td>
<td></td>
<td>9:32 AM, Aug 26, 2018</td>
<td>0.88</td>
<td>0.92</td>
</tr>
<tr>
<td>ccard-recommender-run2-clone(2)</td>
<td>✔️</td>
<td>2m 12s</td>
<td>linear-classifier</td>
<td></td>
<td>11:42 AM, Aug 25, 2018</td>
<td>0.72</td>
<td>0.86</td>
</tr>
<tr>
<td>ccard-recommender-run2-clone(1)</td>
<td>✔️</td>
<td>2m 44s</td>
<td>linear-classifier</td>
<td></td>
<td>10:48 AM, Aug 25, 2018</td>
<td>0.74</td>
<td>0.84</td>
</tr>
<tr>
<td>ccard-recommender-run2</td>
<td>✔️</td>
<td>2m 18s</td>
<td>linear-classifier</td>
<td></td>
<td>10:22 PM, Aug 25, 2018</td>
<td>0.82</td>
<td>0.76</td>
</tr>
<tr>
<td>ccard-recommender-run1-clone(1)</td>
<td>✔️</td>
<td>2m 20s</td>
<td>linear-classifier</td>
<td></td>
<td>10:10 AM, Aug 24, 2018</td>
<td>0.80</td>
<td>0.84</td>
</tr>
<tr>
<td>ccard-recommender-run1</td>
<td>✔️</td>
<td>3m 20s</td>
<td>linear-classifier</td>
<td></td>
<td>6:17 PM, Aug 24, 2018</td>
<td>0.72</td>
<td>0.76</td>
</tr>
</tbody>
</table>

Rows per page: 10

Easy comparison of Runs
## Run overview

<table>
<thead>
<tr>
<th>Show</th>
<th>Run name</th>
<th>Status</th>
<th>Pipeline</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ccard-recommender-run3</td>
<td>✔</td>
<td>linear-classifier</td>
<td>3m 20s</td>
</tr>
<tr>
<td></td>
<td>ccard-recommender-run2-clone(2)</td>
<td>✔</td>
<td>linear-classifier</td>
<td>3m 20s</td>
</tr>
<tr>
<td></td>
<td>ccard-recommender-run2-clone(1)</td>
<td>✔</td>
<td>linear-classifier</td>
<td>3m 20s</td>
</tr>
</tbody>
</table>

## Precision Recall

<table>
<thead>
<tr>
<th></th>
<th>ccard-recommender-run3</th>
<th>ccard-recommender-run2-clone(2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recall</td>
<td><img src="image1" alt="Graph" /></td>
<td><img src="image2" alt="Graph" /></td>
</tr>
<tr>
<td>Precision</td>
<td><img src="image3" alt="Graph" /></td>
<td><img src="image4" alt="Graph" /></td>
</tr>
</tbody>
</table>

## ROC curve

<table>
<thead>
<tr>
<th></th>
<th>ccard-recommender-run3</th>
<th>ccard-recommender-run2-clone(2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TPR</td>
<td><img src="image5" alt="Graph" /></td>
<td><img src="image6" alt="Graph" /></td>
</tr>
<tr>
<td>FPR</td>
<td><img src="image7" alt="Graph" /></td>
<td><img src="image8" alt="Graph" /></td>
</tr>
</tbody>
</table>

### Parameters

### Metrics
That's a wrap.