Machine Learning
Models as a Service

May 20, 2019
Tobias Wenzel
Vigith Maurice
Data Science Operations is not easy

“We spend more time bringing the model to production than developing and training it”
"We spend more time bringing the model to production than developing and training it"
Let's build a platform

Centralized or Decentralized?
Let's build a platform

Centralized!
Let's build a centralized platform

The platform controls the environment in which models are deployed
Let's build a centralized platform

The platform can implement actions of the machine learning model life cycle centrally
Let's build a centralized platform

Create self service APIs for every interaction with the platform
Malicious software libraries found in PyPI posing as well known libraries (gov.sk)
475 points by nariinano on Sept 15, 2017 | hide | past | web | favorite | 245 comments
## Security for an ML Platform

### Table: Package Security

<table>
<thead>
<tr>
<th>Package</th>
<th>Typo-squatting Info</th>
<th>Exploit Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>simplejson</td>
<td>Standard Library</td>
<td>Research / Data leakage</td>
</tr>
<tr>
<td>pkgutil</td>
<td>Standard Library</td>
<td>Research / Data leakage</td>
</tr>
<tr>
<td>timer</td>
<td>Standard Library</td>
<td>Research / Data leakage</td>
</tr>
<tr>
<td>django</td>
<td>Django package</td>
<td>Research / Data leakage</td>
</tr>
<tr>
<td>django</td>
<td>Django package</td>
<td>Research / Data leakage</td>
</tr>
<tr>
<td>django</td>
<td>Django package</td>
<td>Research / Data leakage</td>
</tr>
<tr>
<td>django</td>
<td>Django package</td>
<td>Research / Data leakage</td>
</tr>
<tr>
<td>easy_install</td>
<td>easy_install package</td>
<td>Gain persistence through modifying the .bashrc script.</td>
</tr>
<tr>
<td>asyncio</td>
<td>None</td>
<td>Gain persistence through modifying the .bashrc script.</td>
</tr>
<tr>
<td>pyconfig</td>
<td>None</td>
<td>Gain persistence through modifying the .bashrc script.</td>
</tr>
<tr>
<td>random</td>
<td>None</td>
<td>Reverse shell</td>
</tr>
<tr>
<td>myblobblob</td>
<td>None</td>
<td>Data leakage</td>
</tr>
</tbody>
</table>

Let's build a centralized platform

Runtime security can be taken care of by the platform
Let's build a centralized platform

Updates can be rolled out to every running model by updating the platform
Intuit's ML Platform

Machine Learning Platform

Model **Training** \( \rightarrow \) Model **Execution**

**User Interfaces** (Web, API, CLI)
- Continuous Training (Argo)
- Secure Data Access (Intuit Data Lake)
- Self Service (Model Lifecycle Management)
- Access Control (Authn, Authz)
- Monitoring (Wavefront)
- Logging Tracing (Splunk)
- Data Aggregation (Pre-fetch, Cache, Optimize)
- Auto Scaling (Cost Control)
- Prediction Quality Feedback (Beacons)
- Billing (Chargeback)

**Notebooks** (via SageMaker)
- Storage (Training Data, Model Artifacts)
- ML Compute (via SageMaker)
- Continuous Training (Argo)
- Access Control (Authn, Authz)
- Monitoring (Wavefront)
- Logging Tracing (Splunk)
- Data Aggregation (Pre-fetch, Cache, Optimize)
- Auto Scaling (Cost Control)
- Prediction Quality Feedback (Beacons)
- Billing (Chargeback)

**Intuit's ML Platform**
And Now for Something Completely Different
Let's build a centralized platform

Observability

Beaconing / Monitoring / Logging
Let's build a centralized platform

Introspection

Inspect prediction response (Monitoring Contd.)
Meaning of life = 43

I expected 42!
Beaconing

Re-training and Monitoring
What is a beacon?

```json
{
    "modelName": "meaning-of-life",
    "modelVersion": "1",
    "environment": "PROD",
    "requestBody": "What is the meaning of life?",
    "responseBody": "42"
}

... Metadata ...
```
Realtime Beacons for Predict Request and Responses

```
{
  "modelName": "meaning-of-life",
  "modelVersion": "1",
  "environment": "PROD",
  "requestBody": "What is the meaning of life?",
  "responseBody": "42"
}.encode()
```
Monitoring

the bottom layer of the **Hierarchy of Production Needs**, is fundamental to running a stable service\(^1\).
### Core Building Blocks

- **Model Hosting Service**
- **Cache**
- **Store**
- **Hosted Models**
- **Beacon**

### Monitoring - Platform on Platform

<table>
<thead>
<tr>
<th>Counters</th>
<th>Timers</th>
<th>Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requests</td>
<td>Core Process Time</td>
<td>Request Size</td>
</tr>
<tr>
<td>HTTP Status</td>
<td>Total Response Time</td>
<td>Response Size</td>
</tr>
<tr>
<td>Store Requests</td>
<td>Store Process Time</td>
<td>Store get/put size</td>
</tr>
<tr>
<td>Cache Hit/Miss</td>
<td>Cache Process Time</td>
<td>Cache get/put size</td>
</tr>
<tr>
<td>Beacons</td>
<td>Beacon Emit Time</td>
<td>Beacon size</td>
</tr>
<tr>
<td>Authentication</td>
<td>LMA Process Time</td>
<td></td>
</tr>
</tbody>
</table>

- Cardinality of each is increased by tagging
- Availability is formulated from granular metrics
- Most alerting is based on P99
Let's build a centralized platform

Logging
Centralized Logging

- Accessibility (easy to access)
- Compartmentalized (logs are grouped by Model)
- Traceability (transactional id flows from end user app all the way to Model)
- Near real time
- Log Retention
Machine Learning Models as a Service
1. Make actions in ML lifecycle *self service*

2. Take the *operations burden off* of the data scientist

3. Make sure models are *run securely*

4. Provide *common functionalities* to all models at scale

5. Provide *logging, tracing and monitoring* out of the box