Using Apache Spark with Amazon SageMaker

Julien Simon
Principal Technical Evangelist, AI and Machine Learning
@julsimon

October 2018

Agenda

- Apache Spark on AWS
- Amazon SageMaker
- Combining Spark and SageMaker
- Demos with the SageMaker SDK for Spark
- Getting started

Services covered: Amazon EMR, Amazon SageMaker

aws

Apache Spark on AWS

Apache Spark

https://spark.apache.org/

- Open-source, distributed processing system.
- In-memory caching and optimized execution for fast performance (typically 100x faster than Hadoop).
- Batch processing, streaming analytics, machine learning, graph databases and ad hoc queries.
- API for Java, Scala, Python, R, and SQL.

Apache Spark – *DataFrame*

- Distributed collection of data organized into named columns.
- Conceptually equivalent to a table in a relational database.
- Wide array of sources: structured files, databases.
- Wide array of formats: text, CSV, JSON, Avro, ORC, Parquet.

```
{"name": "Jeff"}
{"name": "Boaz", "age":72}
{"name": "Julien", "age":12}
```

```
df = spark.read.json("people.json")
df.show()
+----+-------+
| age| name  |
+----+-------+
|null| Jeff  |
| 72 | Boaz  |
| 12 | Julien|
+----+-------+
```

MLlib – Machine learning library

https://spark.apache.org/docs/latest/ml-guide.html

- ML algorithms: classification, regression, clustering, collaborative filtering.

- Featurization: feature extraction, transformation, dimensionality reduction.

- Tools for constructing, evaluating and tuning ML pipelines
- Transformer – a transform function that maps a *DataFrame* into a new one
    - Adding a column, changing the rows of a specific column, etc.
    - Predicting the label based on the feature vector.
- Estimator – an algorithm that trains on data
    - Consists of a *fit*() function that maps a *DataFrame* into a *Model*.

Spark ML on Amazon EMR: spam detector

Adapted from https://github.com/databricks/learning-spark/blob/master/src/main/scala/com/oreilly/learningsparkexamples/scala/MLlib.scala

ETL

Train

Predict

```
// Load 2 types of emails from text files: spam and ham (non-spam).
// Each line has text from one email.
val spam = sc.textFile("s3://jsimon-public/spam")
val ham = sc.textFile("s3://jsimon-public/ham")

// Create a HashingTF instance to map email text to vectors of 1000 features.
val tf = new HashingTF(numFeatures = 1000)
// Each email is split into words, and each word is mapped to one feature.
val spamFeatures = spam.map(email => tf.transform(email.split(" ")))
val hamFeatures = ham.map(email => tf.transform(email.split(" ")))

// Create LabeledPoint datasets for positive (spam) and negative (ham) examples.
val positiveExamples = spamFeatures.map(features => LabeledPoint(1, features))
val negativeExamples = hamFeatures.map(features => LabeledPoint(0, features))

val data = positiveExamples.union(negativeExamples)
data.cache()
val Array(trainingData, testData) = data.randomSplit(Array(0.8, 0.2))
trainingData.cache()

// Create a Naive Bayes trainer
val model = NaiveBayes.train(trainingData, 1.0)

val predictionLabel = testData.map(x=> (model.predict(x.features),x.label))
val accuracy = 1.0 * predictionLabel.filter(x => x._1 == x._2).count() / testData.count()
```


aws

Apache Spark on Amazon EMR

https://aws.amazon.com/emr/

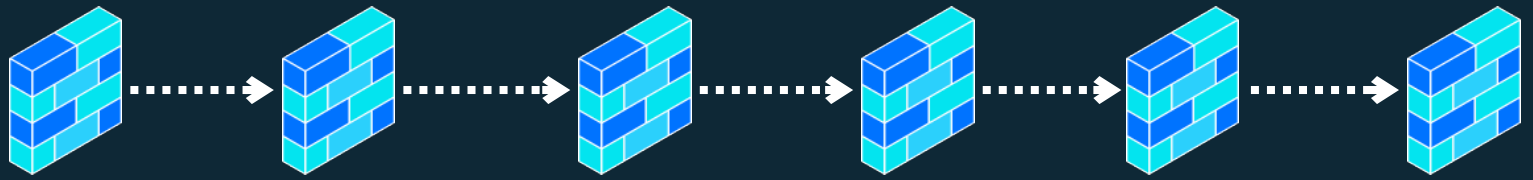
- Spark is natively supported in Amazon EMR.
- Amazon S3 connectivity using the EMR File System (EMRFS).
- Amazon Kinesis, Redshift and DynamoDB as data sources.
- Integration with the AWS Glue Data Catalog.
- Auto Scaling to add or remove instances from your cluster.
- Integration with the Amazon EC2 Spot market.

aws

Amazon SageMaker

Amazon SageMaker

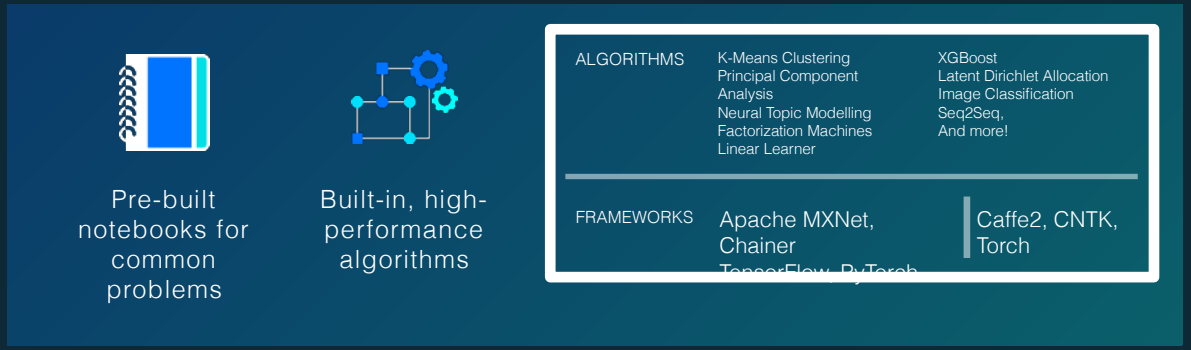
Easily build, train, and deploy Machine Learning models

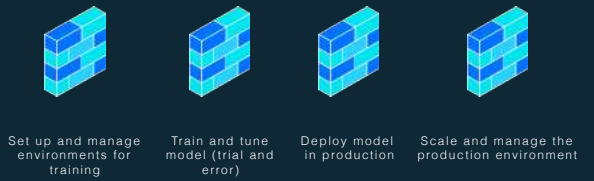


FREE TIER

aws

Amazon SageMaker





Build

Amazon SageMaker

Pre-built notebooks for common problems

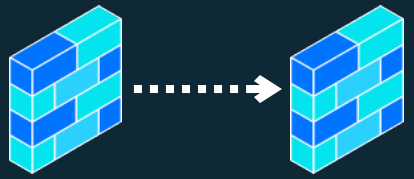
Built-in, high-performance algorithms

Build

One-click training

Hyperparameter optimization

Train



Deploy model in production

Scale and manage the production environment

aws

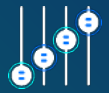
Amazon SageMaker

Pre-built notebooks for common problems

Built-in, high-performance algorithms

Build

One-click training



Hyperparameter optimization

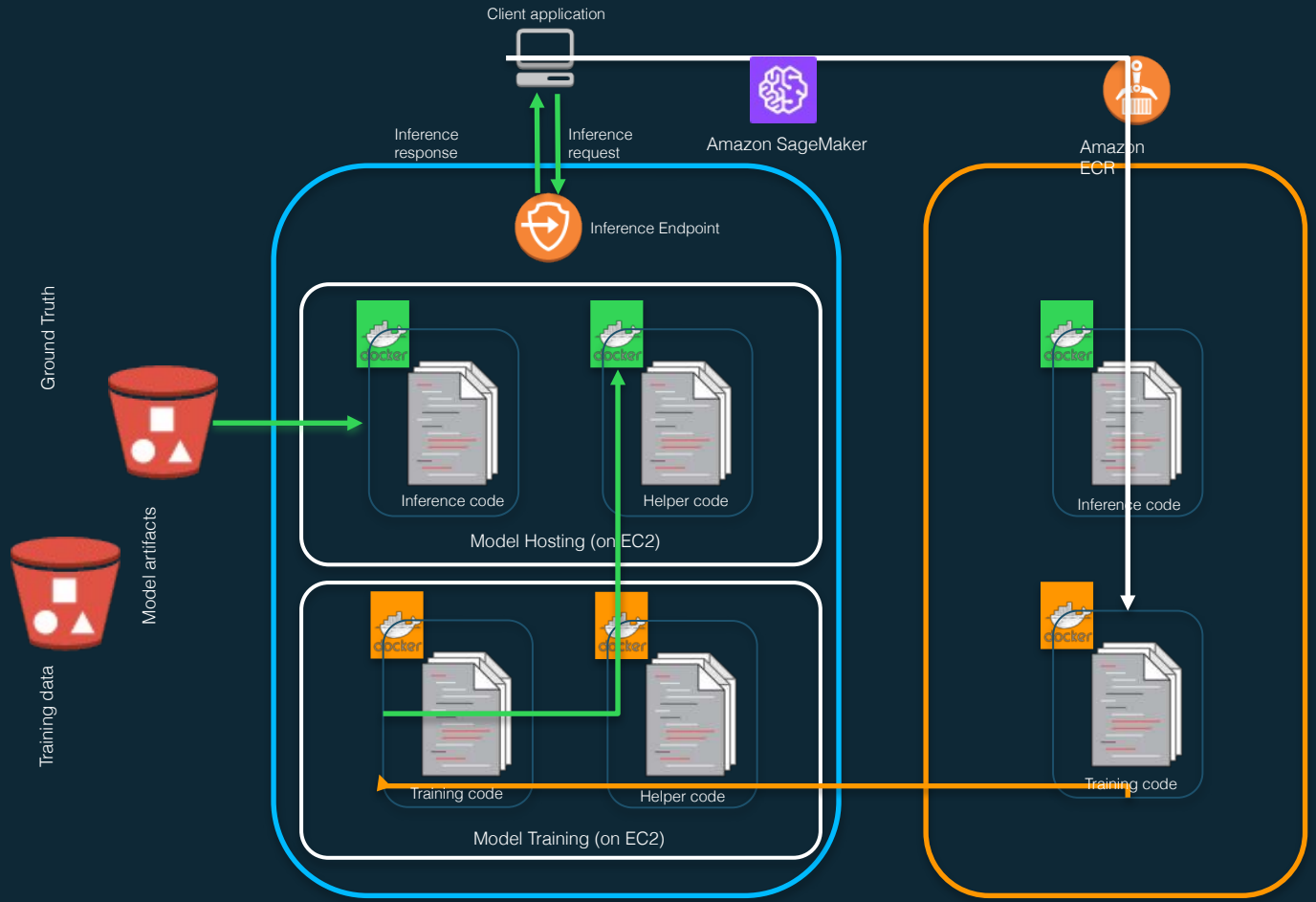
Train

One-click deployment

Fully managed hosting with auto-scaling

Deploy

aws



Combining Spark and SageMaker

Decouple ETL and Machine Learning

- Different workloads require different instance types.
  - Say, M4 for ETL, P3 for training and C5 for prediction?
  - If you need GPUs for training, running your EMR cluster on GPU instances wouldn't be cost-efficient.
- Scale them independently.
  - Avoid oversizing your Spark cluster.
  - Avoid time-consuming resizing operations on EMR.
  - Run ETL once, train many models in parallel.
  - SageMaker terminates training instances automatically.

aws

Run any ML algorithm in any language

- Spark MLlib is great, but you may need something else.

- SageMaker built-in algorithms (ML, DL, NLP).
- Deep Learning libraries, like TensorFlow or Apache MXNet.
- Your own custom code in any language.

aws

Deploy ML models in production

- Perform ML predictions without using Spark.
  - Save the overhead of the Spark framework.
  - Save loading your data in a *DataFrame*.
- Improve latency for small-batch predictions.
  - It can be difficult to achieve low-latency predictions with Spark ML models.
  - You can get real-time predictions with models hosted in SageMaker.
  - You can use very powerful instances for prediction endpoints.

aws

Sample use cases for Spark+SageMaker

- Data preparation and feature engineering before training.
- Data transformation before batch prediction (model reuse).
- Data enrichment with predictions.
  - Predict missing values instead of using median.
  - Add new predicted features.
- Train on extremely large datasets with built-in algos.

aws

SageMaker SDK for Spark

https://github.com/aws/sagemaker-spark

- Python and Scala SDK, for Apache Spark 2.1.1 and 2.2.
- Pre-installed on EMR 5.11 and later.

- Train, import, deploy and predict with SageMaker models directly from your Spark application.
  - Standalone,
  - Integration in Spark MLlib pipelines.

- *DataFrames* in, *DataFrames* out:
  automatic conversion to and from protobuf (crowd goes wild!)

aws

SageMaker SDK for Spark – built-in algorithms

https://docs.aws.amazon.com/sagemaker/latest/dg/algos.html

- High-level API for:
  - Linear Learner
  - Factorization Machines
  - K-Means
  - PCA
  - LDA
  - XGBoost

Infinitely scalable algorithms:
no limit to the amount of data that they can process

- The SageMakerEstimator object lets you use other built-in containers as well any containerized code stored in Amazon ECR (just like the regular SageMaker SDK).

aws

Demos

https://gitlab.com/juliensimon/dlnotebooks

1 – Classifying MNIST in Python with XGBoost (SageMaker)
2 – Clustering MNIST in Scala with K-Means (SageMaker)
3 – Clustering MNIST in Scala with a Pipeline: PCA (MLlib) + K-Means (SageMaker)

aws

Getting started

https://ml.aws
https://aws.amazon.com/sagemaker
https://aws.amazon.com/emr

https://github.com/aws/sagemaker-python-sdk
https://github.com/aws/sagemaker-spark

https://medium.com/@julsimon
https://gitlab.com/juliensimon/dlnotebooks

Thank you!

Julien Simon
Principal Technical Evangelist, AI and Machine Learning
@julsimon

aws