

Introduction to Amazon Sagemaker

sunkeun.choi@gmail.com

6 Jan 2018

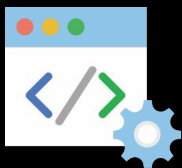
Benefits of using machine learning in cloud

1. Size of dataset
2. Clustered training
3. Hyperparameter tuning
4. Training / Serving Skew
5. De-coupling model from client
6. Autoscaling prediction



Amazon SageMaker

A **fully managed service** that enables **data scientists** and **developers** to quickly and easily **build** machine-learning based models **into production** smart applications.



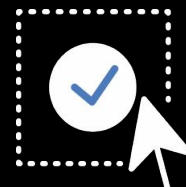
Notebook Instances



1P Algorithms



ML Training Service



ML Hosting Service



Notebook Instance

Few clicks

Amazon SageMaker > Notebook instances > Create notebook instance

Create notebook instance

Amazon SageMaker provides pre-built fully managed notebook instances that run Jupyter notebooks. The notebook instances include example code for common model training and hosting exercises. [Learn More](#)

Notebook instance settings

Notebook instance name

Maximum of 63 alphanumeric characters. Can include hyphens (-), but not spaces. Must be unique within your account in an AWS Region.

Notebook instance type

IAM role

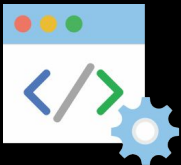
Notebook instances require permissions to call other services including SageMaker and S3. Choose a role or let us create a role with the [AmazonSageMakerFullAccess](#) IAM policy attached.

VPC - *optional*

Notebook instances will have internet access independent of your VPC setting.

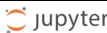
Encryption key - *optional*

An encryption key protects your data. Type the ID or ARN of the AWS KMS key that you want to use.



Notebook Instance

Get a jupyter notebook

 tensorflow_BYOM_iris (autosaved) 

File Edit View Insert Cell Kernel Widgets Help

Connecting to kernel Not Trusted conda_tensorflow_p27

Markdown

TensorFlow BYOM: Train locally and deploy on SageMaker.

- [Introduction](#)
- [Prerequisites and Preprocessing](#)
 - [Permissions and environment variables](#)
 - [Model definitions](#)
 - [Data Setup](#)
- [Training the network locally](#)
- [Set up hosting for the model](#)
 - [Export from TensorFlow](#)
 - [Import model into SageMaker](#)
 - [Create endpoint](#)
- [Validate the endpoint for use](#)

Note: Compare this with the [tensorflow bring your own model example](#)

Introduction

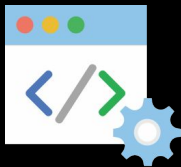
This notebook can be compared to [Iris classification example notebook](#) in terms of its functionality. We will do the same classification task, but we will train the same network locally in the box from where this notebook is being run. We then setup a real-time hosted endpoint in SageMaker.

Consider the following model definition for IRIS classification. This mode uses the `tensorflow.estimator.DNNClassifier` which is a pre-defined estimator module for its model definition. The model definition is the same as the one used in the [Iris classification example notebook](#)

Prerequisites and Preprocessing

Permissions and environment variables

Here we set up the linkage and authentication to AWS services. In this notebook we only need the roles used to give learning and hosting access to your data. The SageMaker SDK will use S3 default buckets when needed. If the `get_execution_role` does not return a role with the appropriate permissions, you'll



Notebook Instance

Pros

- Easy access to data in cloud
- Web interface
- No installation of packages

Cons

- Cost
 - Free tier for 2 month
- Region limitation
 - Seoul is not supported
- Package version constraints

US East (N. Virginia)

US East (Ohio)

US West (N. California)

US West (Oregon)

Asia Pacific (Mumbai)

Asia Pacific (Seoul)

Asia Pacific (Singapore)

Asia Pacific (Sydney)

Asia Pacific (Tokyo)

Canada (Central)

EU (Frankfurt)

EU (Ireland)

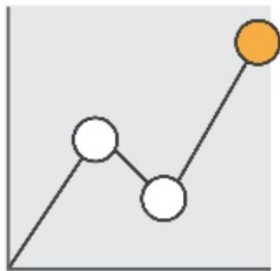
EU (London)

EU (Paris)

South America (São Paulo)



1P Algorithms



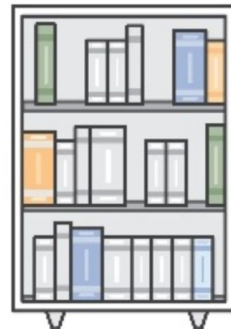
XGBoost, FM, and
Linear for
classification and
regression



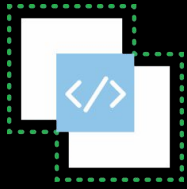
Kmeans and PCA
for clustering and
dimensionality
reduction



Image
classification with
convolutional
neural networks

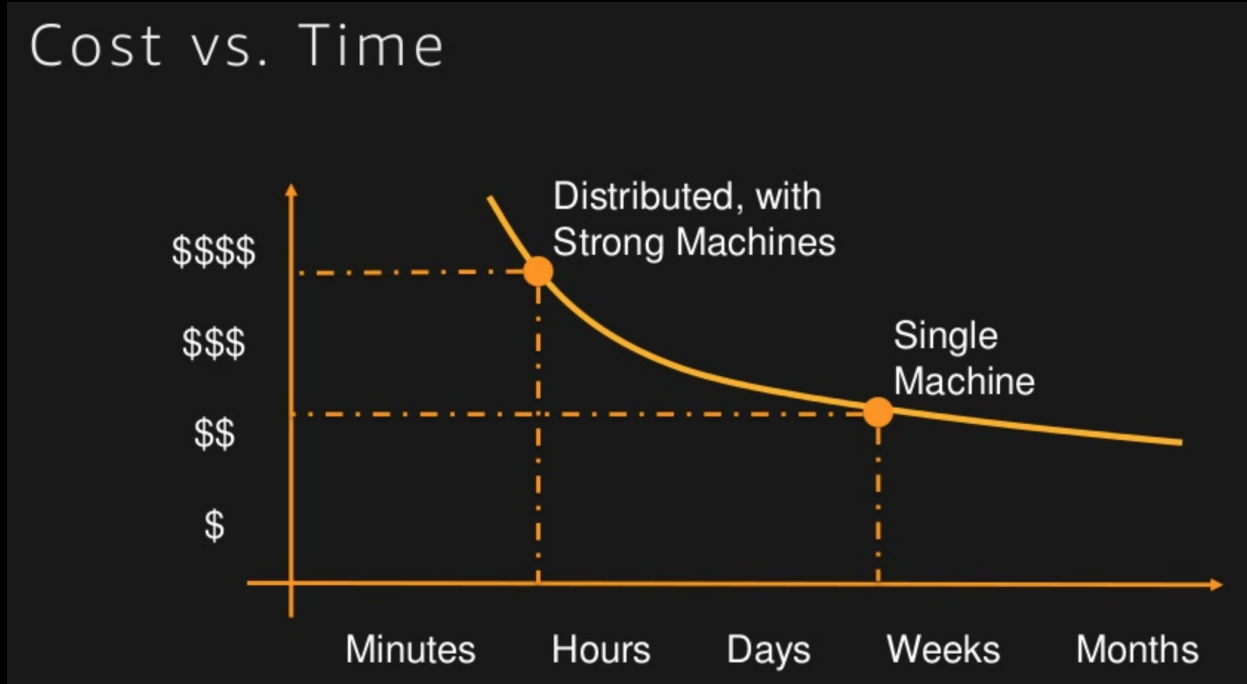


LDA and NTM for
topic modeling,
seq2seq for
translation



ML Training Service

Cost vs Time



Source:

<https://www.slideshare.net/AmazonWebServices/new-launch-infinitely-scalable-machine-learning-algorithms-with-amazon-ai-mcl341-reinvent-2017>

Distributed training

1. Scale to hundreds of GPU
2. Train outrageously large models
3. Customize models with few lines of codes

But...

Need to manage instance orchestration (Kubernetes or Mesos)

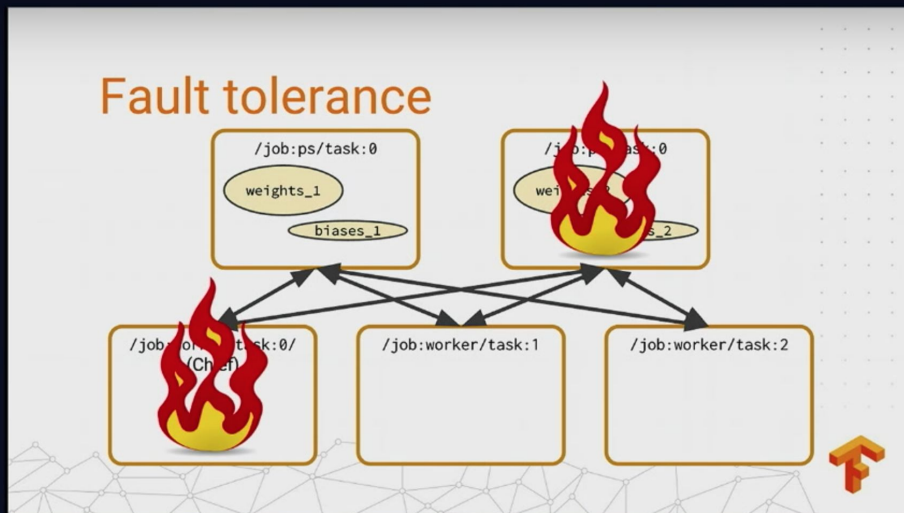
Need to manage fault tolerance

An example of managing kubernetes cluster

<https://blog.openai.com/infrastructure-for-deep-learning/>

Source: https://youtu.be/la_M6bCV91M?list=PLOU2XLXmsIKGc_NBoIhTn2Qhraj53cv

Managed vs manual



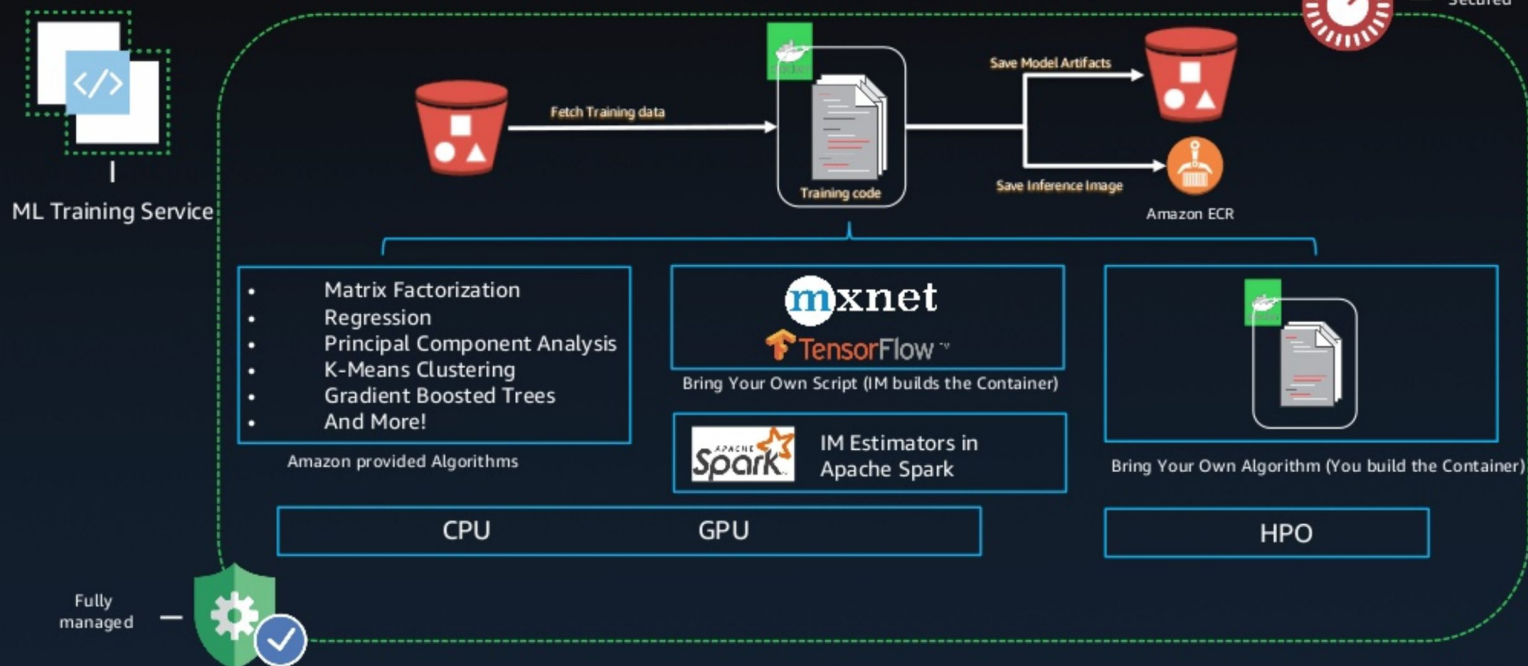
#tfdevsummit

Source: https://youtu.be/la_M6bCV91M?list=PLOU2XLYxmsIKGc_NBoIhTn2Qhraj53cv&t=1358

3

Managed Distributed Training With Flexibility

Clip slide

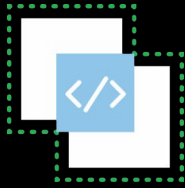


AWS re:Invent

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

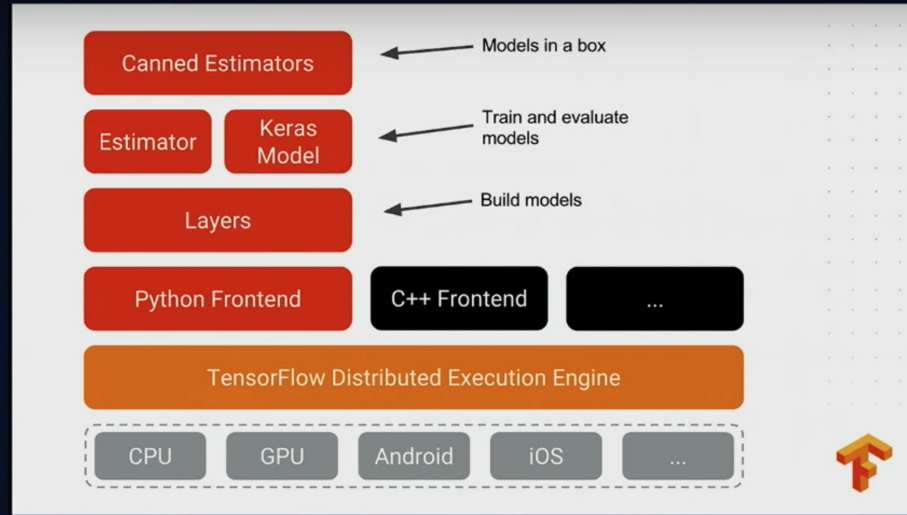
@dmbanga





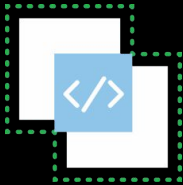
ML Training Service

To use cloud services `tf.estimator` is required



#tfdevsummit

Source: https://youtu.be/t64ortpgS-E?list=PLOU2XLYxmsIKGc_NBoIhTn2Qhraj53cv&t=361



ML Training Service

Build models by using estimators

TensorFlow™

Install

Develop

API r1.4

Deploy

Extend

Community

Versions

>

Search

GITHUB

Develop

GET STARTED

PROGRAMMER'S GUIDE

TUTORIALS

PERFORMANCE

MOBILE

Programmer's Guide

Estimators

Tensors

Variables

Graphs and Sessions

Saving and Restoring

Importing Data

Embeddings

Debugging TensorFlow Programs

TensorFlow Version Compatibility

Frequently Asked Questions

TensorFlow Versions

Estimators

This document introduces **Estimators**—a high-level TensorFlow API that greatly simplifies machine learning programming. Estimators encapsulate the following actions:

- training
- evaluation
- prediction
- export for serving

You may either use the pre-made Estimators we provide or write your own custom Estimators. All Estimators—whether pre-made or custom—are classes based on the `tf.estimator.Estimator` class.

★ **Note:** TensorFlow also includes a deprecated `Estimator` class at `tf.contrib.learn.Estimator`, which you should not use.

Advantages of Estimators

Estimators provide the following benefits:

- You can run Estimators-based models on a local host or on a distributed multi-server environment without changing your model. Furthermore, you can run Estimators-based models on CPUs, GPUs, or TPUs without recoding your model.

Contents

Advantages of Estimators

Pre-made Estimators

Structure of a pre-made Estimators program

Benefits of pre-made Estimators

Custom Estimators

Recommended workflow

Creating Estimators from Keras models



ML Training Service

Need to wrap with sagemaker api

Create a training job using the sagemaker.TensorFlow estimator

```
from sagemaker.tensorflow import TensorFlow

mnist_estimator = TensorFlow(entry_point='mnist.py',
                             role=role,
                             training_steps=1000.
```

Preparing the TensorFlow training script

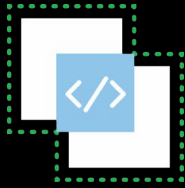
Your TensorFlow training script must be a **Python 2.7** source file. The current supported TensorFlow version is **1.4.0**.

This training script **must contain** the following functions:

- `model_fn` : defines the model that will be trained.
- `train_input_fn` : preprocess and load training data.
- `eval_input_fn` : preprocess and load evaluation data.
- `serving_input_fn` : defines the features to be passed to the model during prediction.

Version constraints (Python 2.7, Tensorflow 1.4.0)

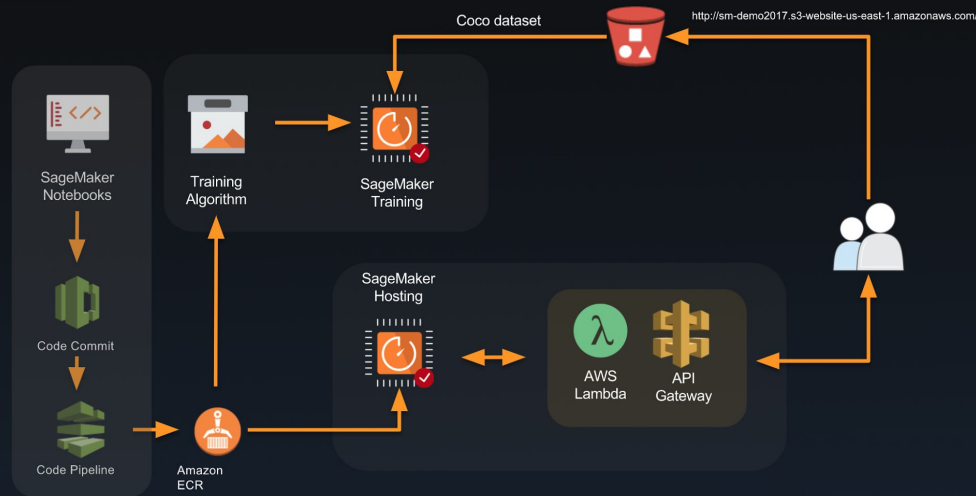
Source: <https://github.com/aws/sagemaker-python-sdk>



ML Training Service

Training PyTorch is possible

PyTorch on SageMaker Style Transfer App Architecture



AWS re:Invent

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

@dmbanga



Source: <https://www.slideshare.net/AmazonWebServices/new-launch-introducing-amazon-sagemaker-mcl365-reinvent-2017>



ML Training Service

Summary

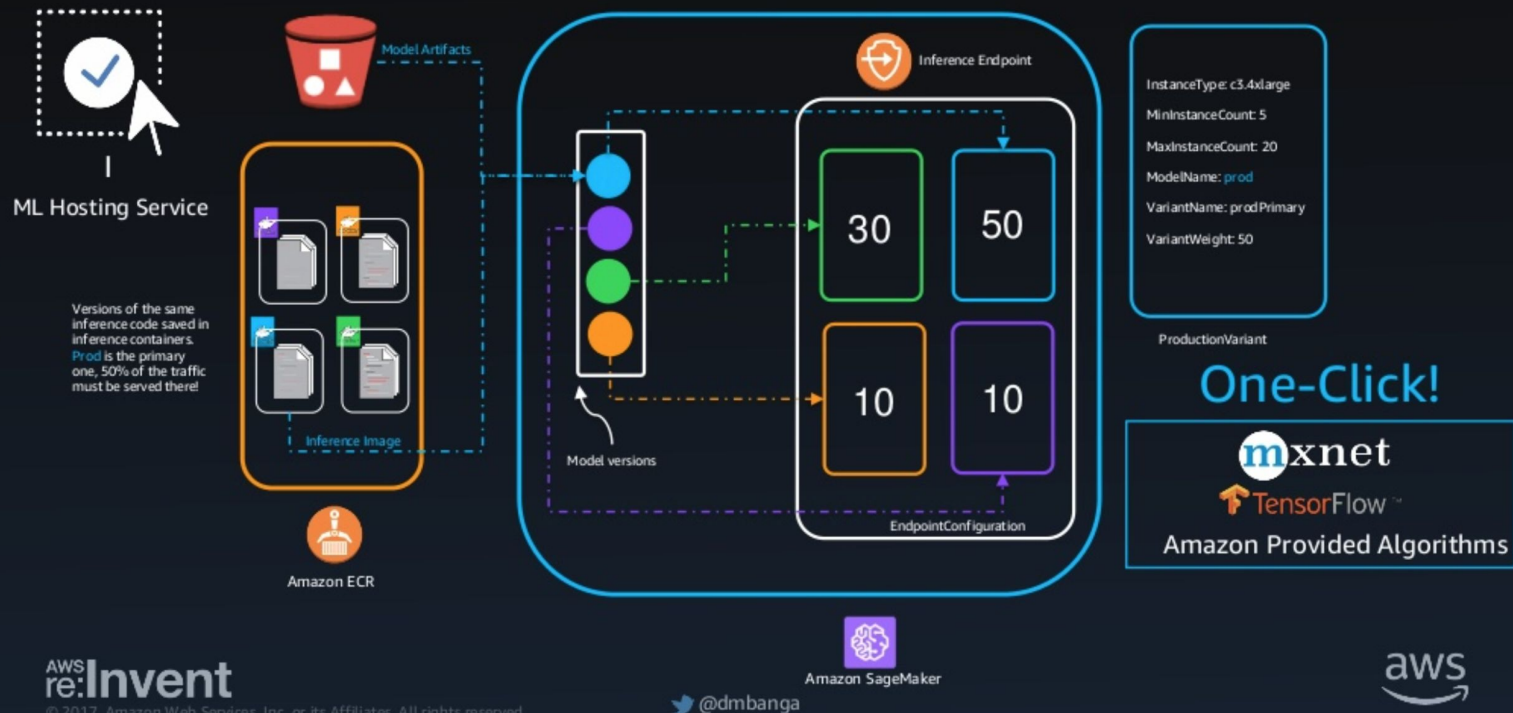
- Easy to use distributed training
- Supporting various frameworks
 - Tensorflow
 - MxNet
 - PySpark
 - PyTorch* ...

[*]: BYO Docker Containers with SageMaker Estimators

4

Easy Model Deployment to Amazon SageMaker

Clip slide

AWS
re:Invent

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



Amazon SageMaker

@dmbanga

aws



ML Hosting Service

Summary

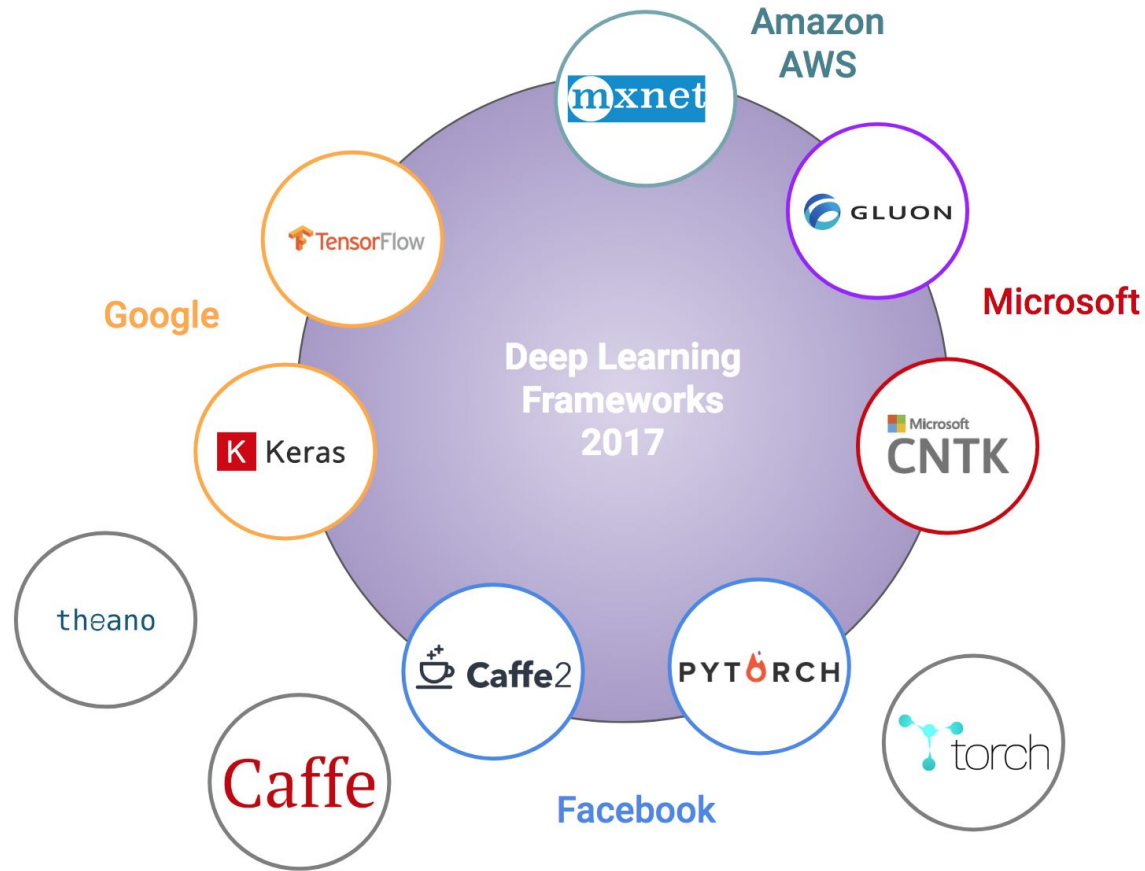
- Easy to deploy
- Auto-Scaling Inference APIs
- A/B Testing
- Low Latency & High Throughput
- Bring Your Own Model
- Python SDK

	Amazon ML	Amazon SageMaker*	Azure ML Studio	Google Prediction API	Google ML Engine**
Classification	✓	✓	✓	✓	✓
Regression	✓	✓	✓	✓	✓
Clustering	✗	✓	✓	✗	✓
Anomaly detection	✗	✓	✓	✗	✓
Recommendation	✗	✓	✓	✗	✓
Ranking	✗	✓	✓	✗	✓
Algorithms	unknown	10 built-in + custom available	100+ algorithms and modules	unknown	TensorFlow-based
Frameworks	✗	TensorFlow, MXNet	✗	✗	TensorFlow
Graphical interface	✗	✗	✓	✗	✗
Automation level	high	medium	low	high	low

*Both out-of-the-box features and possible custom-built features are marked as available in Amazon SageMaker

**The features available in TensorFlow are respectively marked as available in Google ML Engine.

Source: <https://www.altexsoft.com/blog/datascience/comparing-machine-learning-as-a-service-amazon-microsoft-azure-google-cloud-ai/>



Source: <https://towardsdatascience.com/battle-of-the-deep-learning-frameworks-part-i-cff0e3841750>

Conclusion

- Where is your data? - AWS vs GCP
- How big is your model? - Cluster vs Instance
- Try to use `tf.estimator`
- Which deep learning framework are you using?
 - Keep track model converters, model zoo
 - Training efficiency
 - Tutorials, examples, community
- Cost vs Time

Thank you