SONY

SoVeC Smart VideoにおけるAmazon SageMakerを用いた開発運用事例

Feb. 27th, 2020

ソニー R&Dセンター Tokyo Laboratory 16 大石壮一郎

自己紹介

大石壮一郎

- ✓ 所属: ソニー（株） R&Dセンター
- ✓ 技術: 画像処理、クラウド、機械学習
- ✓ 職種: PLを中心に何でも雑多に
- ✓ 業務: 新規事業の立上げ（通算6回目）
  - ✓ 機械学習関連: 4回、SageMaker: 今回初
  - ✓ 前回: Edgeでの機械学習（Python 2.7）

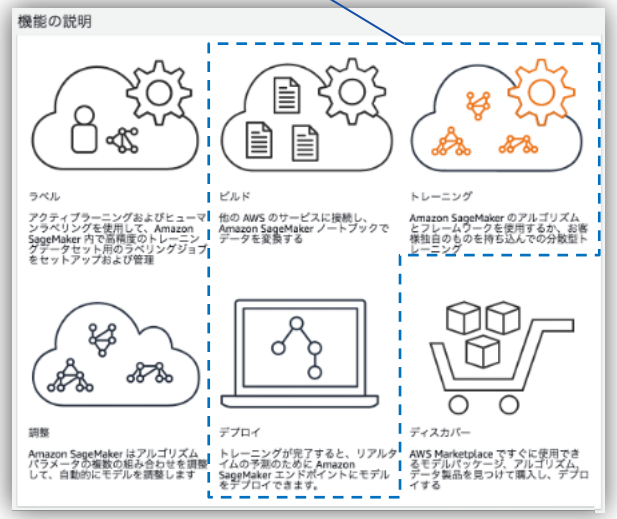
- ✓ 好きなサービス

  AWS Lambda、SageMaker

executive summary (, agenda)

- ✓ SoVeC Smart Videoでは深層学習を用いた超解像技術を提供している
- ✓ 超解像技術はAWS SageMakerの複数機能をベースに組み合わせて構築した
- ✓ SageMakerにより短期間で機械学習のDevOpsを実現した

SoVeC Smart Videoに採用



AWS SageMaker機能群

SoVeC Corp.

SoVeC株式会社

SoVeC

ソニーネットワークコミュニケーションズと総合PR会社ベクトルの
ジョイントベンチャー

SONY
Sony Network Communications

AIやIoTなど、ソニーグループの
技術を活用したソニーネットワーク
コミュニケーションズの先進的な
サービス事業における知見

vector

ベクトルグループの
コミュニケーション戦略
実行力を活用

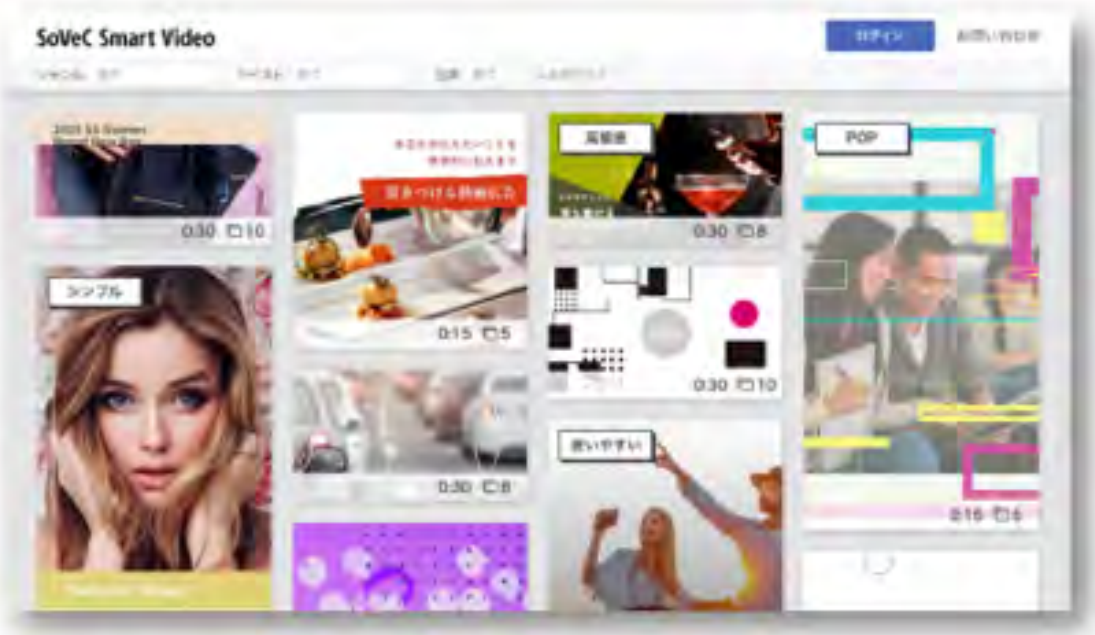
デジタルコミュニケーションの民主化を目指し、日々高度
化する企業のマーケティング活動の最適化・効果の最大化
を図る新たなソリューションを提供

SoVeC Smart Video

コンセプト

SoVeC

デザイナークオリティの動画を、誰でも簡単・リーズナブルに
～デザイナーが伴走するように動画が出来る～



<https://smartvideo.sovec.net/cp/>

5

SoVeC Smart Video: サービス概要

SoVeC Smart Video 概要

SoVeC

AIを活用した動画自動生成クラウドサービス

ブラウザだけでプロのデザイナー級の動画を誰でもあっという間に

貴社でお持ちの情報や資料

ウェブサイト
記事・コラム
イベント情報

プレスリリース
営業資料
社内報

EC商材
商品情報

クラウド上で簡単に動画制作

映像・画像
テキスト

素材を投入

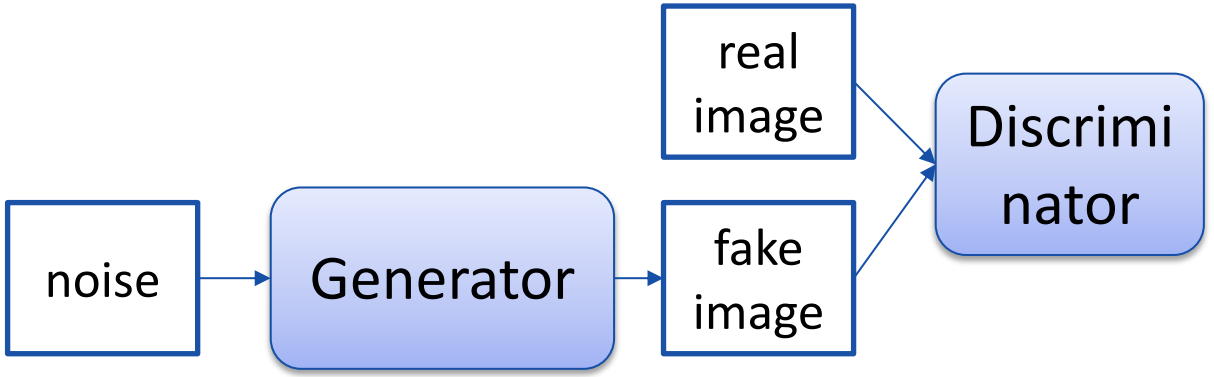
SoVeC Smart Video

様々な用途に使える動画を出力

4

SoVeC Smart Video: SRGAN を用いた超解像機能

低解像な静止画素材を高精細化する

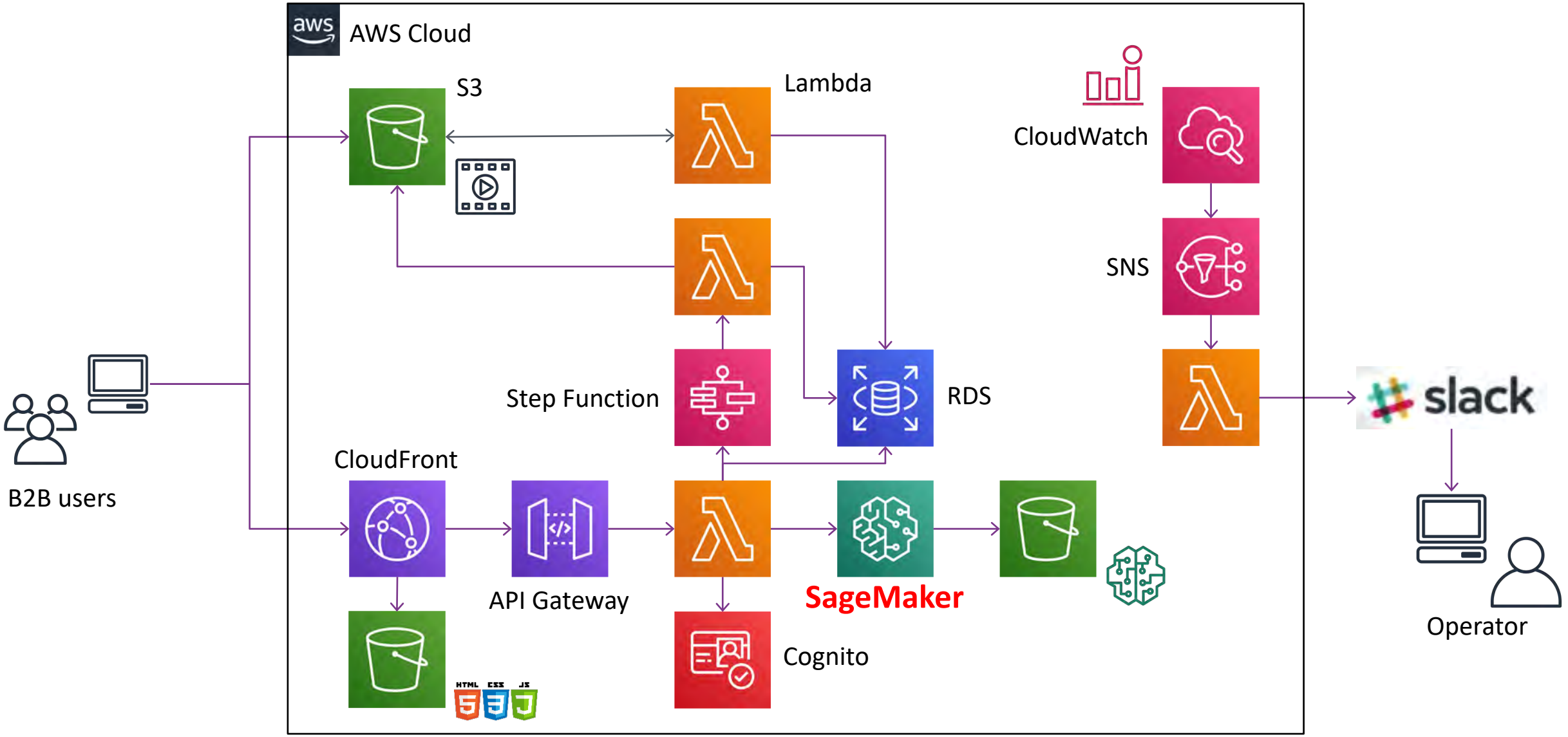


Super Resolution Generative Adversarial Network

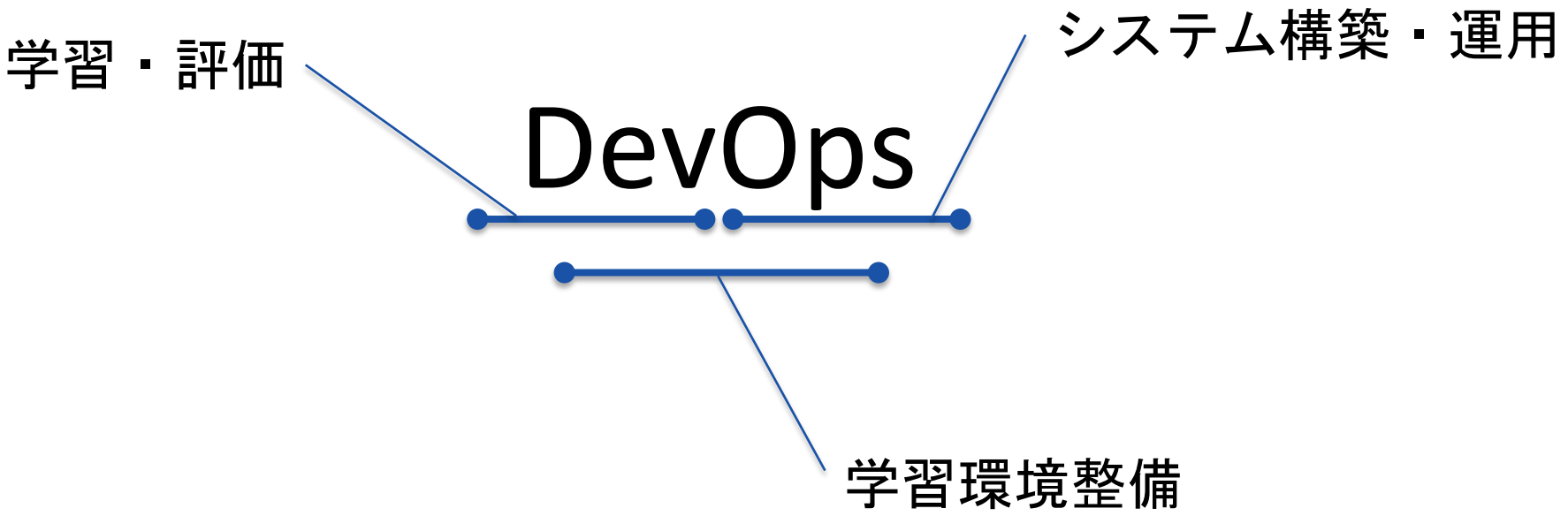
Original

Super Resolution

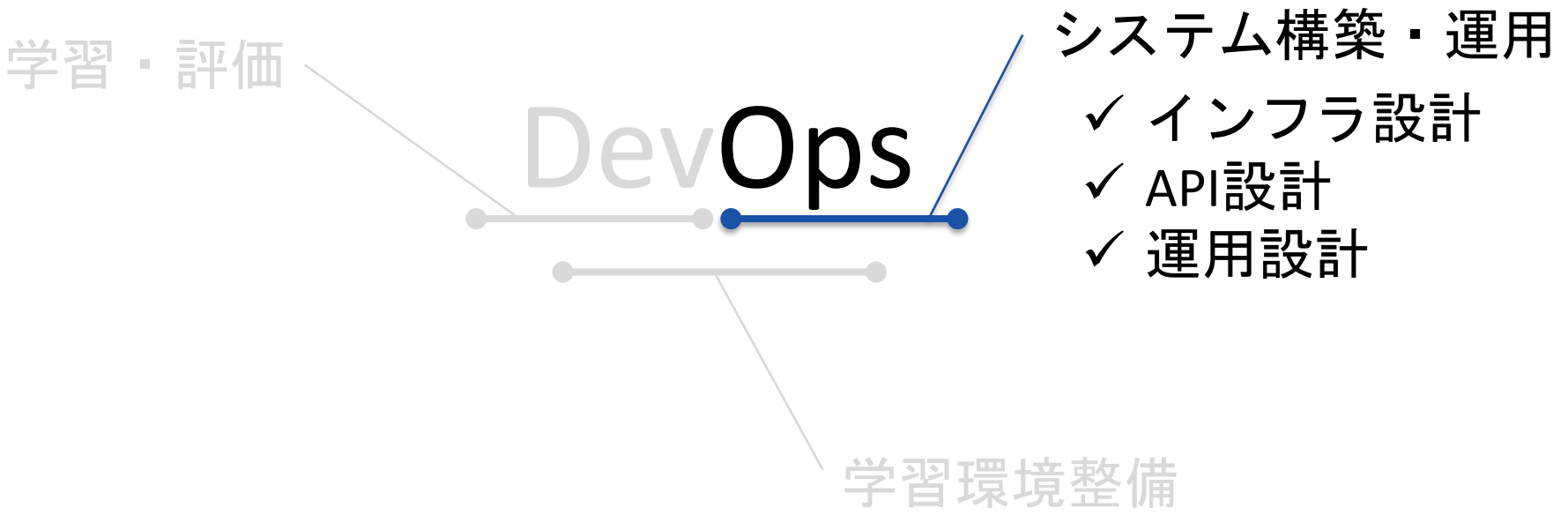
SoVeC Smart Video: システム概要



機械学習のDevOps



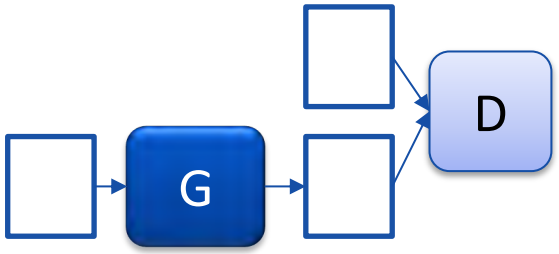
機械学習のDevOps: システム構築・運用



システム構築・運用: Before

DevOps

機能提供方法も課題



推論精度の向上

機能の提供

<u>インフラ設計</u> / Computing resource案

ECS	OSレイヤーの運用必要
Fargate	GPUインスタンスなし
Lambda	スペック不足

(2020.02.27 現在)

- ✓ インフラ構成どうする？
- ✓ どのようにI/Fを設計する？
- ✓ 運用どうする？

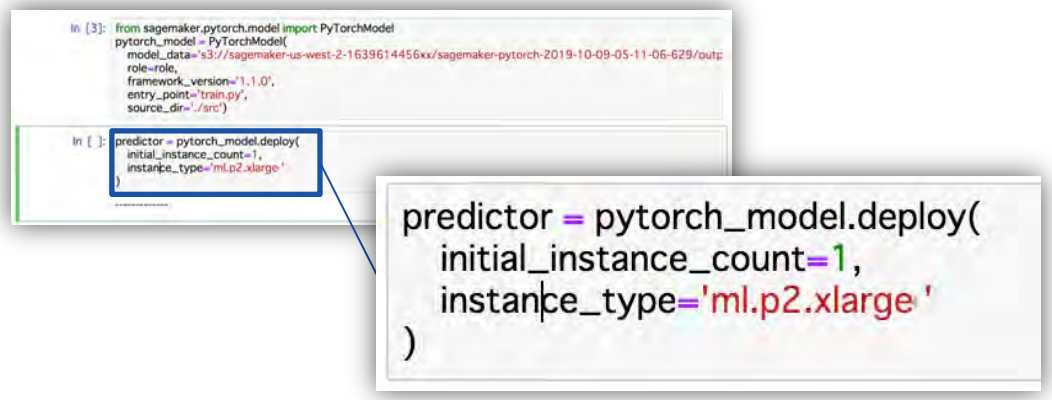
<u>実装</u>: cf. Edge(前回): 同じリソース上に配置

<u>商用向け運用設計</u>

システム構築・運用: After

DevOps

SageMaker / endpoint機能



ミニマムのシステム構築例

- ✓ サーバーレス/フルマネージド
- ✓ 充実したSDKやサンプル
- ✓ シンプルなデプロイ

シンプルなインフラ構成案

具体的な標準メソッド

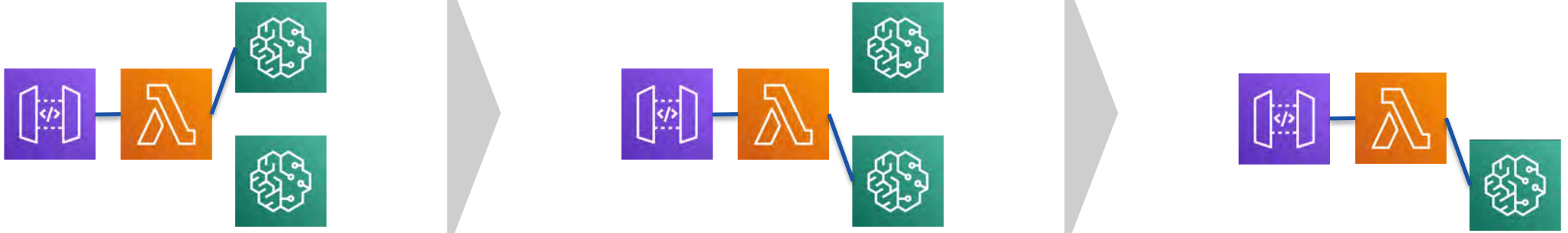
```
def model_fn(model_dir):        # モデル読込み
def input_fn(in, content_type): # 入力データ受取り
def predict_fn(data, model):    # 推論の実行
def output_fn(prediction, accept): #レスポンス生成
```

商用運用に耐えられるデプロイ（次頁）

システム構築・運用: After/デプロイ

DevOps

Blue/Greenデプロイを実現



新たなendpointを生成

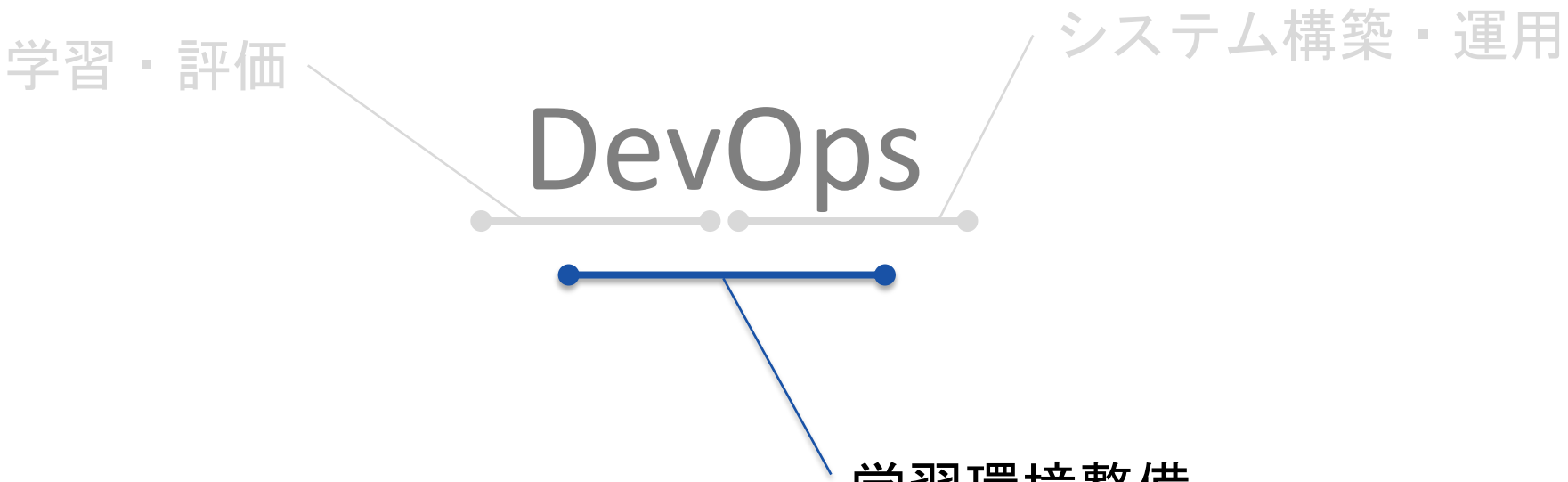
endpointを変更

古いendpointを削除

ベストプラクティスなのか？

- ✓ マルチモデルエンドポイント…は違う気がする
- ✓ より良い運用方法があるのではないか？

機械学習のDevOps: 学習環境整備



学習環境整備

- ✓ 環境構築に工数がかかる
- ✓ 環境維持に工数がかかる
- ✓ 一方であまり工数をかけたくない

学習環境整備: きっかけ

DevOps

<https://www.python.org/doc/sunset-python-2/>

Python 2.7サポート終了のお知らせ / 2019.09.10（立上げ終盤）

- ✓ 2020年1月1日、Python 2.7 サポート終了
- ✓ Deep Learningの環境を Python 2.7で構築済み
- ✓ 順次サポートが打ち切られるはず…

突きつけられる選択肢

現行のまま リリースするか

vs.

Python 3.xに 載せ替えるか

ユーザのデータが入ってきた後は、更に載せ替えに
手間がかかるので初回リリース前の載せ替えを選択

学習環境整備: Before

DevOps

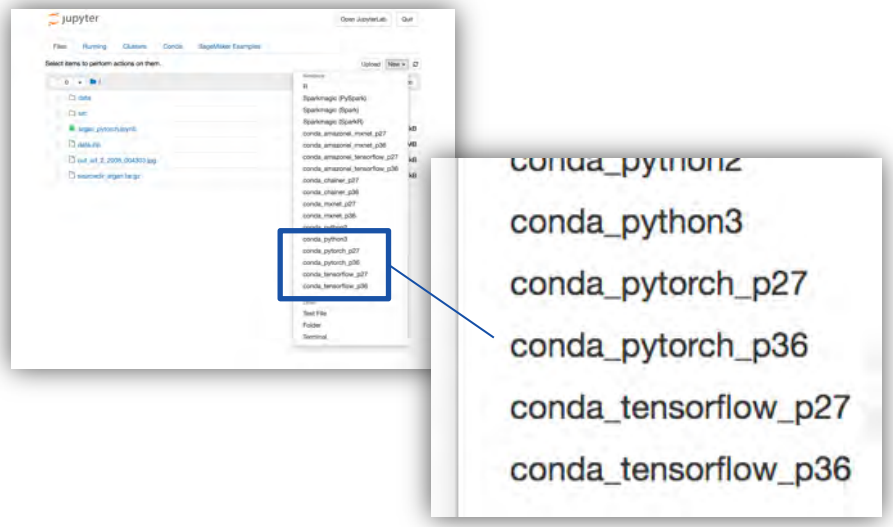
単純なPython 2.7 → 3.x だけ変更に留まらない影響範囲（環境構築の煩雑さ）

- ✓ Ubuntu 18.04は意外とHWと相性悪い
- ✓ GPUの進化に対するSWの追従
- ✓ CUDAの最新版は非対応のDL F/W
- ✓ cuDNNとCUDAのバージョンの関係
- ✓ Anacondaとpipの混合
- ✓ DL Framework: 進化が早い
- ✓ 周辺モジュール: 細かく動作しない
- ✓ etc. etc...

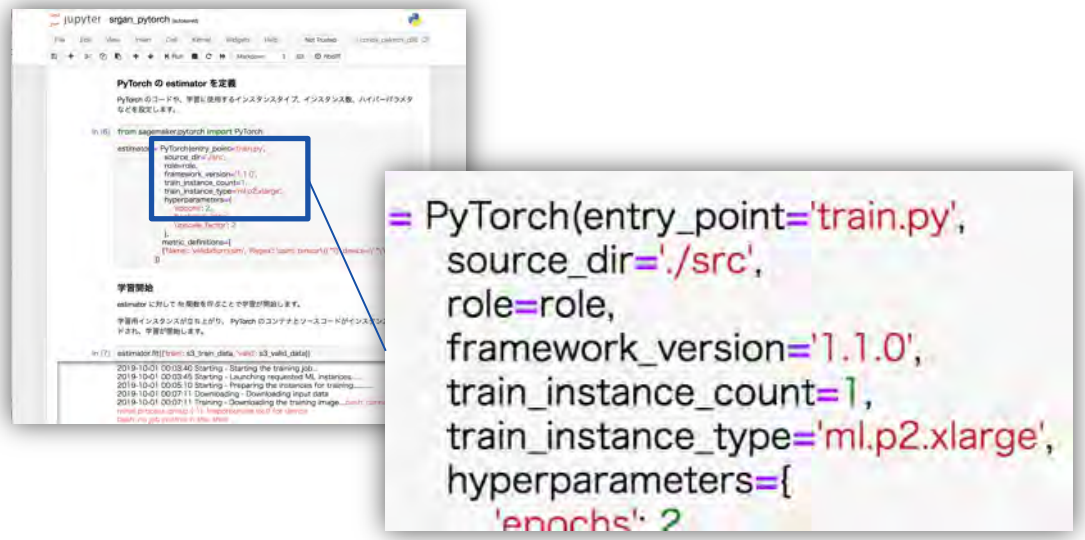
環境の構築・維持に工数が取られる

学習環境整備: After

SageMaker / Notebook



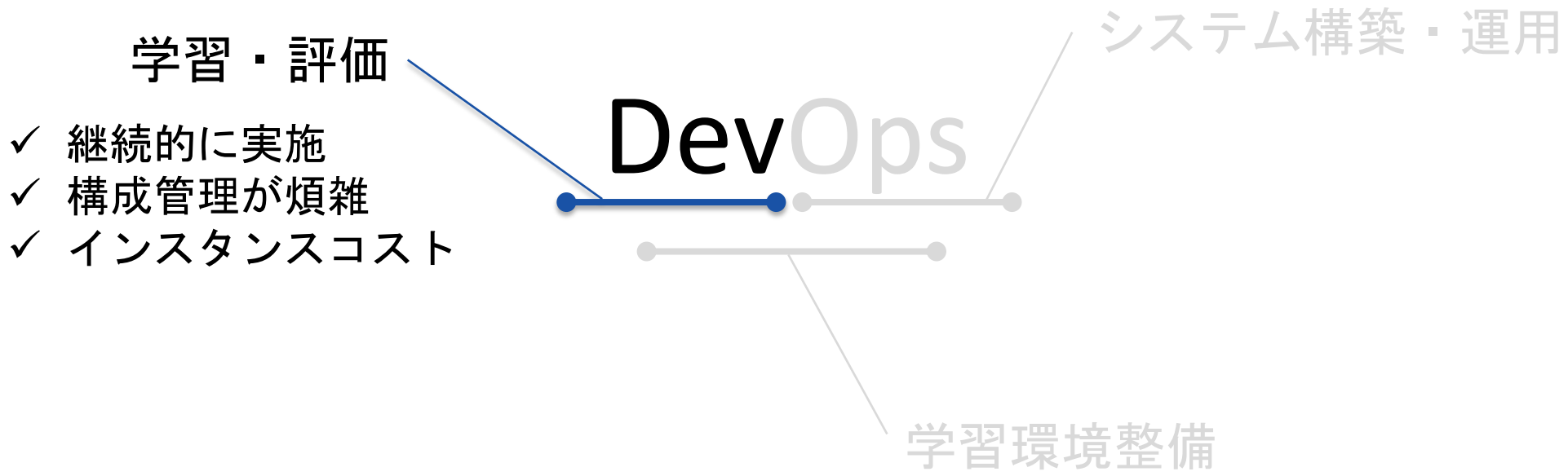
SageMaker / Jobs



✓ ベースのSW stackを選択
✓ DL Frameworkのバージョンを選択
✓ (もちろん詳細な設定も可能)

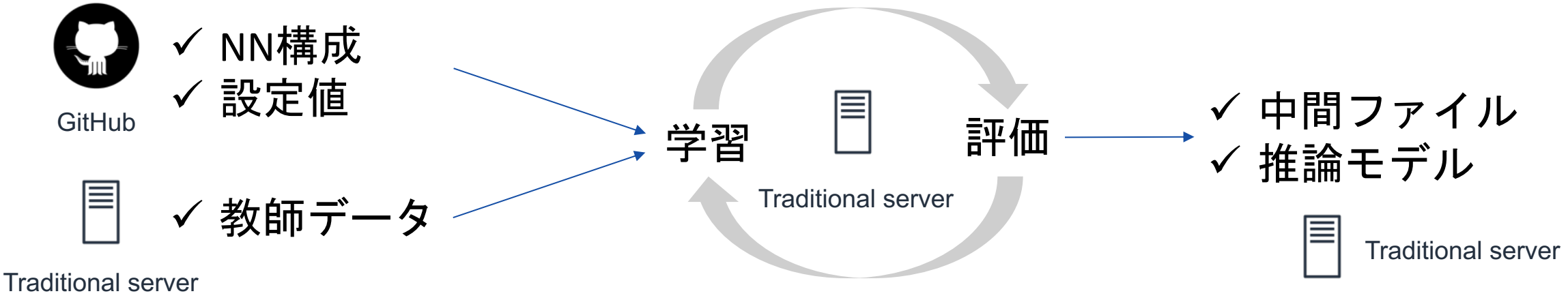
学習・評価に
注力できる

機械学習のDevOps: 学習・評価



学習・評価: Before

DevOps



機械学習の精度向上の難しさ

- ✓ 勾配消失問題
- ✓ 学習効率のためのデータ整備
- ✓ etc. etc....

試行錯誤の連続

構成管理が煩雑

- ✓ 中間ファイルと学習過程の紐付け
- ✓ 前処理した教師データ群の整理
- ✓ 推論モデルとリリース物の紐付け

学習・評価: After

DevOps

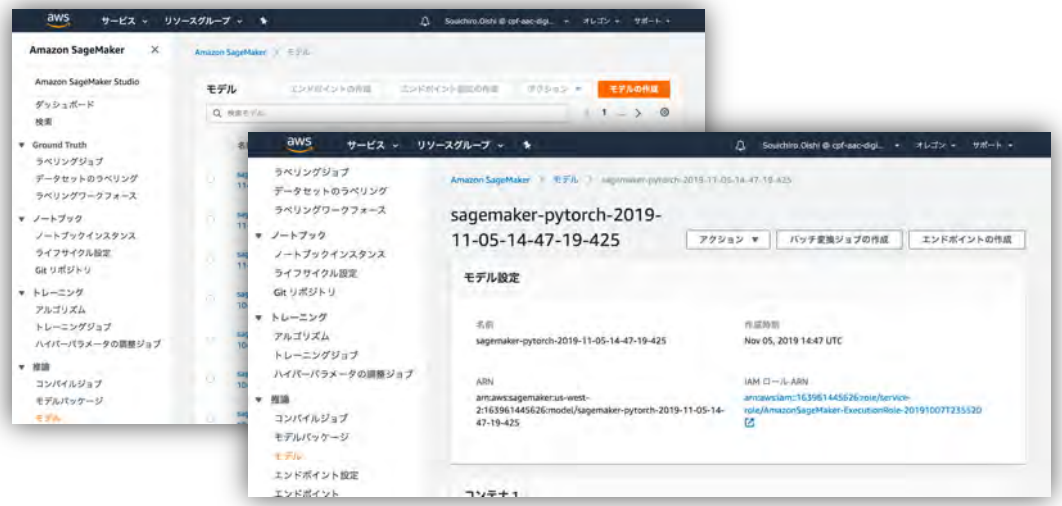
GitHub

✓ NN構成
✓ 設定値

✓ 教師データ

学習 評価

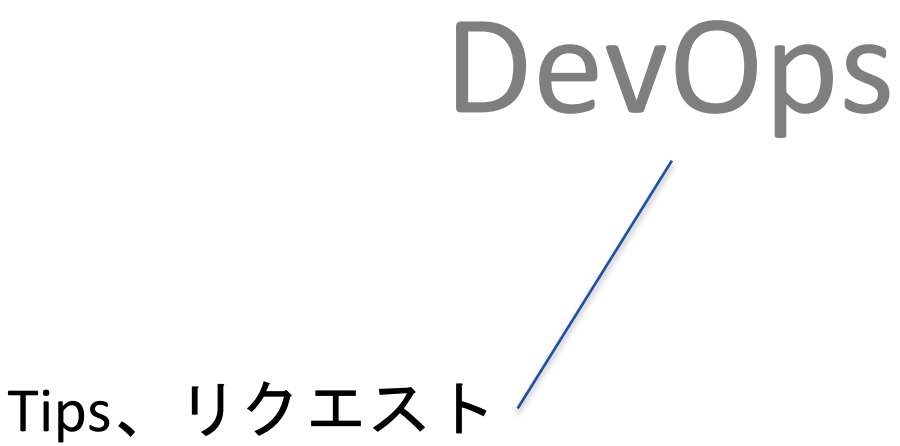
✓ 中間ファイル
✓ 推論モデル



構成管理が容易に

✓ in/outの全データをURIで紐付け
✓ 教師データのバージョン管理も可能
✓ 再デプロイ、ロールバックも容易

機械学習のDevOps: Tips、リクエスト



Tips: 学習環境整備/ローカルでの環境構築

DevOps

Local mode: ローカルPCにSageMakerの環境が構築できる

ローカル環境の構築・維持が必要だが下記ユースケースでは有効

- ✓ 会社の規定などの理由で教師データが社外環境に置けない
- ✓ 学習環境が既に整備されている（ホスティングとの連携）
- ✓ 個人ユース

```
In [5]: from sagemaker.tensorflow import TensorFlow

mnist_estimator = TensorFlow(entry_point='mnist.py',
                             role=role,
                             framework_version='1.12.0',
                             training_steps=100,
                             evaluation_steps=100,
                             train_instance_count=2,
                             train_instance_type='local')

mnist_estimator.fit(inputs)
```

instance_type を ‘local’ とする

Memo: HyperParameter最適化は未対応

```
~/anaconda3/envs/py36sagelocal/lib/python3.6/site-packages/sagemaker/session.py in compile_model(self, input_model_config, output_model_config, role, job_name, stop_condition, tags)
    427
    428         LOGGER.info("Creating compilation-job with name: %s", job_name)
--> 429         self.sagemaker_client.create_compilation_job(**compilation_job_request)
    430
    431     def tune(  # noqa: C901

AttributeError: 'LocalSagemakerClient' object has no attribute 'create_compilation_job'
```

(2020.02.27 現在)

Tips: 学習・評価/スポットインスタンスの利用

DevOps

aws

Amazon SageMaker で、機械学習のトレーニングコストを最大 90% 削減するマネージドスポットトレーニングが利用できるようになりました

投稿日: Aug 26, 2019

Amazon SageMaker において、マネージドスポットトレーニングと呼ばれる、新しいフルマネージドオプションがサポートされました。これは、Amazon EC2 スポットインスタンスを利用して機械学習モデルをトレーニングするものです。スポットインスタンスでは、AWS クラウド内の使用されていないコンピューティング性能を活用できます。結果として、機械学習モデルのトレーニングコストを最適化し、オンデマンドインスタンスに比べて最大 90% 削減できます。

スポットインスタンスは Amazon SageMaker が管理するため、お客様がキャパシティーを継続的にポーリングする必要はありません。また、追加でツールを構築する必要もありません。トレーニングジョブは、スポットキャパシティーが入手できたときに Amazon SageMaker によってできる限り高い信頼性で実行されます。マネージドスポットトレーニングは、トレーニングモデルが SageMaker 内の一般的な ML フレームワーク、SageMaker の組み込みアルゴリズム、カスタム構築モデルを使用して構築されている場合に使用できます。また、マネージドスポットトレーニングの自動モデルチューニングを使用して、機械学習モデルを調整することもできます。

マネージドスポットトレーニングは、Amazon SageMaker で使用できるすべてのインスタンスタイプでサポートされ、Amazon SageMaker が提供されている AWS リージョンすべてで使用できます。詳細については、AWS ドキュメントを参照してください。また、ブログ記事でも詳細をご覧いただけます。

https://aws.amazon.com/jp/about-aws/whats-new/2019/08/amazon-sagemaker-launches-managed-spot-training-saving-machine-learning-training-costs/

```
estimator = PyTorch(entry_point='train.py',
                    source_dir='./src',
                    role=role,
                    framework_version='1.1.0',
                    train_instance_type='ml.p2.xlarge',
                    train_instance_count=1,
                    train_use_spot_instances=True,
                    train_max_run=36000,
                    train_max_wait=36000,
                    hyperparameters={
                        'epochs': 10000,
                        'backend': 'gloo',
                        'upscale_factor': 2
                    },
                    metric_definitions=[
                    {'Name': 'validat
                ])
```

```
train_instance_count=1,
train_use_spot_instances=True,
train_max_run=36000,
train_max_wait=36000,
```

試してみたが...

- ✓ 時間がないときに、なければ待つ、はできない
- ✓ 時間がないときに、中断するかも、は辛かった
- ✓ 時間に余裕があればコストインパクト: 大

リクエスト: システム構築・運用

DevOps

Amazon Elastic Interface

- ✓ CPUインスタンスにGPUを付与できる
- ✓ GPUインスタンスよりもコストダウン可能
- ✓ TensorFlowとApache MXNet、ONNXをサポート

(2020.02.27 現在)

Open Neural Network Exchange (ONNX) フォーマットのサポート

ONNX は、深層学習フレームワークでモデルをトレーニングし、推論のために別のモデルに転送できるようにするオープンフォーマットです。これにより、さまざまなフレームワークの相対的な強みを活用することができます。例えば、ONNX を使用すると、PyTorch の柔軟性によって構築およびトレーニングされたモデルを Apache MXNet に転送して、大規模な推論を効率的に実行できるようになります。ONNX は PyTorch、MXNet、Chainer、Caffe2、Microsoft Cognitive Toolkit に統合されており、TensorFlow を含む他の多くのフレームワ

是非PyTorchもサポートしていただきたいです！

https://aws.amazon.com/jp/machine-learning/elastic-inference/features/

まとめ

手間のかかる機械学習の環境構築や構成管理などを少ない
工数で実現することができ、学習・評価の作業に注力できた。
さらに、デプロイ、トレーニング、ビルドといった単位で
SageMakerの各機能を段階的に導入できることが良かった。