機械学習レンズ

AWS Well-Architected フレームワーク

2020 年 4 月

Archived

This paper has been archived.

The latest version is now available at:

https://docs.aws.amazon.com/ja_jp/wellarchitected/latest/machine-learning-lens/welcome.html

注意

お客様は、本書に記載されている情報を独自に評価する責任を負うものとします。本書は、(a) 情報提供のみを目的としており、(b) AWS の現行製品とプラクティスを表しますが、予告なしに変更されることがあり、(c) AWS およびその関連会社、サプライヤーまたはライセンサーからの契約義務や確約を意味するものではありません。AWS の製品やサービスは、明示または暗示を問わず、一切の保証、表明、条件なしに「現状のまま」提供されます。お客様に対する AWS の責任は、AWS 契約により規定されます。本書は、AWS とお客様の間で締結される一切の契約の一部ではなく、その内容を修正することもありません。

© 2020 Amazon Web Services, Inc. or its affiliates. All rights reserved.

Archived

目次

Archived

要約

このドキュメントでは、AWS Well-Architected フレームワークの機械学習レンズについて説明します。このドキュメントでは、一般的な機械学習 (ML) のシナリオについて説明し、ワークロードがベストプラクティスに従って設計されていることを確認するための重要な要素を示しています。

Archived

はじめに

AWS Well-Architected フレームワークは、AWS でシステムを構築する際に、決定の是非を判断するのに役立ちます。信頼性が高く、安全で、効率が良く、費用対効果が高いクラウド対応システムを設計して運用するために、アーキテクチャに関するベストプラクティスをこのフレームワークに従って学ぶことができます。フレームワークにより、アーキテクチャをベストプラクティスに照らし合わせて一貫的に測定し、改善すべき点を特定する手段を提供します。Well-Architected システムによってビジネスが成功する可能性が大いに高まると弊社は確信しています。

この機械学習レンズでは、AWS クラウドで機械学習のワークロードを設計、デプロイ、設計する方法に焦点を当てます。このレンズは、Well-Architected フレームワークに含まれるベストプラクティスに追加されます。説明を簡潔にするため、このレンズには機械学習 (ML) のワークロードに特化した詳細のみを記載しています。ML ワークロードを設計するときは、「AWS Well-Architected フレームワーク」のホワイトペーパーに記載された最適なベストプラクティスや質問を参考にしてください。

このレンズは、最高技術責任者 (CTO)、アーキテクト、開発者、および運用チームメンバーなど、技術面での役割を担う人物を対象としています。このホワイトペーパーを読むと、AWS で ML ワークロードを設計し運用する際に使われるベストプラクティスと戦略を理解することができます。

Archived

定義

機械学習レンズは、運用上の優秀性、セキュリティ、信頼性、パフォーマンス効率、コスト最適化という 5 つの柱を基本としています。AWS では、ML アプリケーションの堅牢なアーキテクチャを設計できる、ML ワークロードのための複数のコアコンポーネントを提供しています。

機械学習のワークロードを構築する際に評価すべき点は、次の 2 つです。

- 機械学習のスタック
- 機械学習ワークロードのフェーズ

機械学習のスタック

AWS で ML ベースのワークロードを構築する際には、カスタマイズのレベルや ML のスキルレベルと市場投入までのスピードのバランスを考慮して、さまざまな抽象度を選択することができます。

- 人工知能 (AI) サービス
- ML サービス
- ML フレームワーク & インフラストラクチャ

AI サービス

AI サービスのレベルでは、API 呼び出しを使用してワークロードに ML 機能を素早く追加することができる、完全マネージド型のサービスが提供されます。これにより、コンピュータビジョン、音声、自然言語、チャットボット、予測、推奨などの機能を備えた、強力でインテリジェントなアプリケーションを構築できます。このレベルのサービスは、事前にトレーニング済み、または自動的にトレーニングされる機械学習や深層学習モデルに基づいているため、ML の知識がなくても利用できます。

Archived

AWS では、API 呼び出しを通じてアプリケーションと統合できる AI サービスが多数提供されています。たとえば、Amazon Translate はテキストコンテンツの翻訳やローカライズ、Amazon Polly はテキスト読み上げ変換、Amazon Lex は対話型チャットボットの構築に使用できます。

ML サービス

ML サービスのレベルでは、開発者、データサイエンティスト、研究者に機械学習のためのマネージドサービスとリソースを提供します。これらのタイプのサービスを利用すると、基盤となるインフラストラクチャのニーズを心配することなく、データのラベル付け、構築、トレーニング、デプロイ、カスタム ML モデルの運用を行うことができます。インフラストラクチャ管理の差別化につながらない重労働はクラウドベンダーによって管理されるため、データサイエンスチームは得意分野に集中することができます。

AWS の Amazon SageMaker を使用すると、開発者とデータサイエンティストは、あらゆる規模の ML モデルを迅速かつ簡単に構築、トレーニング、デプロイできます。たとえば、Amazon SageMaker Ground Truth では、高精度な ML トレーニングデータセットを迅速に構築でき、Amazon SageMaker Neo では、開発者が ML モデルを一度トレーニングしてから、クラウドやエッジの任意の場所で実行できるようにします。

ML フレームワーク & インフラストラクチャ

ML フレームワークとインフラストラクチャのレベルは、機械学習の専門家を対象としています。このような人たちは、モデルを構築、トレーニング、調整、デプロイするための独自のツールやワークフローを設計することに慣れているほか、フレームワークやインフラストラクチャレベルでの作業にも慣れています。

AWS では、TensorFlow、PyTorch、Apache MXNet などのオープンソースの ML フレームワークを利用できます。このレベルの 深層学習 AMI と深層学習コンテナには、パフォーマンスが最適化された複数の ML フレームワークがプリインストールされています。このような最適化のおかげで、Amazon EC2 P3 や P3dn インスタンスのような強力で ML に最適化されたコン

Archived

ピューティングインフラストラクチャ上でフレームワークを常に起動できるようになり、機械学習ワークロードの速度と効率が向上します。

レベルの組み合わせ

通常、ワークロードでは複数のレベルの ML スタックサービスが使用されます。ビジネスのユースケースに応じて、異なるレベルのサービスやインフラを組み合わせれば、複数の要件を満たし、複数のビジネス目標を達成できます。たとえば、小売ウェブサイトのカスタマーレビューの感情分析に AI サービスを利用したり、マネージド型 ML サービスを使用して自社のデータでカスタムモデルを構築し、将来の売上を予測したりすることができます。

ML ワークロードのフェーズ

一般的な ML ワークロードの構築と運用は反復的なプロセスであり、複数のフェーズで構成されます。これらのフェーズは、一般的なガイドラインである CRISP-DM (Cross Industry Standard Process Data Mining) のオープンスタンダードのプロセスモデルに大まかに従っています。CRISP-DM がベースラインとして使用されているのは、業界で実績のあるツールであり、アプリケーションに依存せず、さまざまな ML パイプラインやワークロードに適用しやすい手法であることが理由です。

エンドツーエンドの機械学習プロセスには、次のようなフェーズがあります。

- ビジネス目標の特定
- ML 問題の骨組み
- データの収集と統合
- データの準備
- データの可視化と分析
- 特徴量エンジニアリング
- モデルのトレーニング

Archived

- モデルの評価
- ビジネスの評価
- 本稼働用のデプロイ (モデルのデプロイとモデル推論)

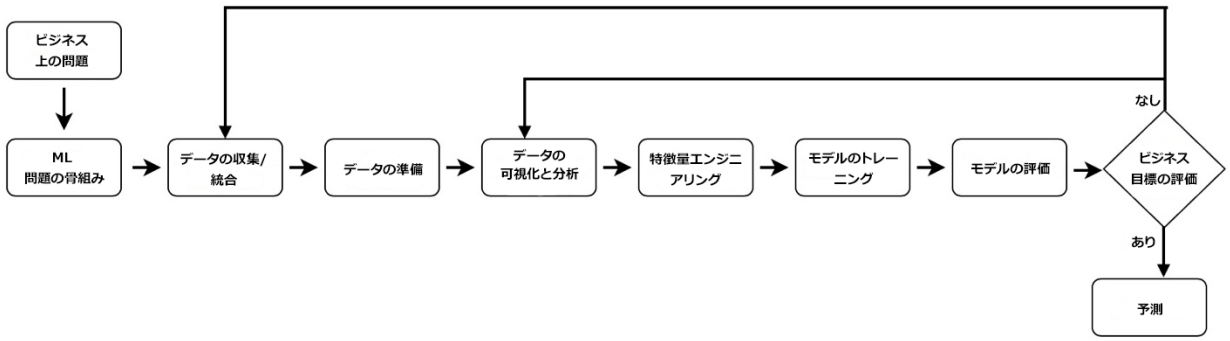

図 *1* - エンドツーエンドの機械学習プロセス

ビジネス目標の特定

ビジネス目標の特定は最も重要な段階です。ML を検討している組織は、解決すべき問題と、その問題を ML で解決することで得られるビジネスバリューを明確に把握する必要があります。具体的なビジネスの目標や成功の基準に対して、ビジネスバリューを測定できることが求められます。これはどのような技術的なソリューションにも当てはまりますが、ML は破壊的な技術であることから、ML ソリューションを検討する際に、このステップは特に難しくなります。

成功の基準を決めたら、組織がその目標に向かって実際に実行する能力を評価します。目標は達成可能なものであり、本稼働への明確な道筋を示すものでなければなりません。

ビジネス目標を達成するためには、ML が適切なアプローチであることを検証する必要があります。目標を達成するために利用できるすべての選択肢を評価します。採用するアプローチを決定する際には、各アプローチの最終結果の精度、コスト、スケーラビリティを評価します。

ML ベースのアプローチを成功させるためには、トレーニングしようとしているアルゴリズムに適用可能かつ関連性の高い高品質のデータを豊富に揃えることが不可欠です。データの可

Archived

用性を慎重に評価し、適切なデータソースが利用可能でアクセス可能であることを確認します。たとえば、ML モデルのトレーニングやベンチマークにはトレーニングデータが必要ですが、ML ソリューションの価値を評価するためにはビジネスからのデータも必要です。

以下のベストプラクティスが適用されます。

- ビジネス要件を理解する
- ビジネスに関する質問を作成する
- プロジェクトの ML の実現可能性とデータ要件を判断する
- データ取得、トレーニング、推論、間違った予測にかかるコストを評価する
- 同様の領域で実績のある、または公開済みの作業を確認する (利用できる場合)
- 許容可能なエラーを含む、主要なパフォーマンスメトリクスを判断する
- ビジネスに関する質問に基づいて機械学習のタスクを定義する
- 重要かつ必須の機能を特定する

ML 問題の骨組み

このフェーズでは、何を観測し、何を予測するべきか (ラベルまたはターゲット変数) というビジネス上の問題が機械学習の問題の骨組みになります。ML においては、何を予測し、パフォーマンスとエラーのメトリクスをいかに最適化すべきかを決定することが重要なステップとなります。

たとえば、製造業社が利益を最大化する製品を特定する場合を考えてみます。このビジネス目標を達成するためには、どれだけの製品を生産するかを適切に決定することがひとつの鍵となります。このシナリオでは、過去と現在の売上から、製品の将来の売上を予測します。将来の売上を予測することが解決すべき問題となりますが、その問題の解決にも ML を利用できます。

以下のベストプラクティスが適用されます。

- プロジェクトの成果の成功を判断するための基準を定義する

Archived

- プロジェクトの正確性、予測のレイテンシー、インベントリ値の最小化など、観測可能かつ定量化可能なパフォーマンスメトリクスを確立する
- 入力、必要な出力、最適化されるパフォーマンスメトリクスの観点から ML の問題を定式化する
- ML が実現可能で適切なアプローチであるかどうかを評価する
- データソーシングおよびデータのラベル付けの目標と、それを達成するための戦略を作成する
- 解釈しやすく、デバッグをより管理しやすいシンプルなモデルから始める

データ収集

ML ワークロードにおいて、データ (入力とそれに対応する必要な出力) は次の 3 つの重要な働きをします。

- システムの目標の定義: 出力形式、入出力ペアによる各出力と各入力との関係
- 入力を出力に関連付けるアルゴリズムのトレーニング
- トレーニング済みのモデルのパフォーマンスを測定し、パフォーマンスの目標が達成されたかどうかを評価する

まずは、ML モデルに必要なデータを特定し、モデルのトレーニング用データの収集に使用できるさまざまな手段を評価します。

企業が大量のデータを収集、分析するようになるにつれ、従来型のオンプレミスソリューションを使用したデータストレージ、データ管理、分析は時代遅れになる可能性があります。クラウドベースのデータレイクは、あらゆる規模ですべての構造化データと非構造化データを保存できる、一元化されたリポジトリです。構造化しなくてもデータをそのまま保存できるほか、ダッシュボードやビジュアライゼーションからビッグデータ処理、リアルタイム分析、ML まで、さまざまなタイプの分析を実行できるため、より良い意思決定につなげることができます。

Archived

AWS では、静的なリソースからのデータの一括取り込みや、ウェブサイト、モバイルアプリ、インターネット接続デバイスなど、動的に生成される新しいソースからのデータの取り込みなど、さまざまな方法でデータを取り込むことができます。たとえば、Amazon Simple Storage Service (Amazon S3) を使用すると、高度にスケーラブルなデータレイクを構築できます。AWS Lake Formation を使用すると、データレイクを簡単に設定できます。

データを取り込むには、AWS Direct Connect を使用してデータセンターを AWS リージョンに直接プライベートに接続します。AWS Snowball を使用すれば、ペタバイト単位のデータを一括で物理的に転送できます。エクサバイト単位のデータであれば、AWS Snowmobile を使って転送できます。AWS Storage Gateway を使用して既存のオンプレミスストレージを統合したり、AWS Snowball Edge を使用してクラウド機能を追加したりすることもできます。また、Amazon Kinesis Data Firehose を使用すれば、複数のストリーミングデータソースを収集して取り込むこともできます。

以下のベストプラクティスが適用されます。

- データを抽出するために必要なさまざまなソースとステップを詳細化する
- 数量と品質の両方で、データの可用性を確認する
- ダウンストリームでの利用に備えてデータを準備する前に、データを総合的に理解する
- データガバナンスを定義する: 誰がデータを所有しているのか、誰がアクセス権を持っているのか、データの適切な使用方法、要求に応じて特定のデータにアクセスしたり削除したりする機能
- データの系統を追跡することで、後続で処理する場所とデータソースを追跡して把握する
- データの収集と統合にマネージド型の AWS のサービスを使用する
- データレイクなど、一元化されたアプローチを使用してデータを保存する

Archived

データの準備

ML モデルの善し悪しは、トレーニングに使用するデータにかかっています。データを収集した後は、そのデータの統合、注釈、準備、処理が重要になります。本質的に、適切なトレーニングデータは、学習や一般化のために最適な方法で提供されるという特徴を持っています。データの準備は、統計的に有効な小さなサンプルから開始し、データの整合性を常に維持しながら、さまざまなデータ作成戦略を用いて反復的に改善する必要があります。

AWS では、データの注釈、抽出、転送、ロード (ETL) を大規模に行うことができるサービスをいくつか提供しています。

Amazon SageMaker は、データのラベル付けと準備、アルゴリズムの選択、モデルのトレーニングやデプロイのための調整と最適化、予測といった機械学習ワークフロー全体に対応するフルマネージドサービスです。

Amazon SageMaker Ground Truth では、公的機関や民間企業のラベル付け担当者を簡単に利用できるほか、一般的なラベル付けタスク用の組み込みのワークフローとユーザーインターフェイスを提供します。 機械学習モデルを使用して未加工データに自動的にラベル付けを行い、手動によるラベル付けの何分の一かのコストで迅速に高品質なトレーニングデータセットを作成します。アクティブラーニングモデルが確信を持ってラベル付けできない場合にのみ、データが人間にルーティングされます。 このサービスでは、動的なカスタムワークフロー、ジョブ連鎖、ジョブ追跡を利用でき、過去のラベル付けジョブの出力を新しいラベル付けジョブの入力として使用することによって、後続の ML によるラベル付けジョブの時間が短縮されます。

AWS Glue は、ETL パイプラインの自動化に使用できる、完全マネージド型の抽出、変換、ロード (ETL) サービスです。AWS Glue は、Glue データカタログを使用してデータを自動的に検出してプロファイルし、ソースデータをターゲットスキーマに変換する ETL コードを推奨・生成し、ETL ジョブを完全マネージド型のスケールアウトされた Apache Spark 環境上で実行して、データを目的地にロードします。また、複雑なデータフローの設定、オーケストレーション、モニタリングも可能です。

Archived

Amazon EMR は、動的にスケーラブルな Amazon EC2 インスタンス間で大量のデータを簡単かつ迅速に処理できるマネージド型 Hadoop フレームワークを提供します。また、Apache Spark や HBase、Presto、Flink といった他の一般的なフレームワークを EMR で実行することや、Amazon S3 や Amazon DynamoDB といった他の AWS データストア内のデータを操作することもできます。

データの準備は、機械学習モデルを構築するために使用されるトレーニングデータだけでなく、モデルのデプロイ後にモデルに対する推論を行うために使用される新しいビジネスデータにも適用されます。通常、トレーニングデータに適用するのと同じシーケンスのデータ処理ステップが推論リクエストにも適用されます。

Amazon SageMaker Inference Pipeline は、パイプラインをデプロイできるため、未加工の入力データを渡して事前処理、予測、事後処理を実行し、推論リクエストをリアルタイムとバッチの両方で処理できます。推論パイプラインを使用すると、既存のデータ処理機能を再利用できます。

以下のベストプラクティスが適用されます。

- データ準備では、統計的に有効なサンプルデータの小さなセットから始める
- さまざまなデータ準備戦略を反復的に試す
- データクリーニングプロセス中にフィードバックループを実装し、データの準備ステップで異常が発生したらアラートを提供する
- データの整合性を継続的に適用する
- マネージド型 ETL サービスを活用する

データの可視化と分析

データを理解する際に重要となるのが、パターンの識別です。通常、このようなパターンは、表のデータのみを見ているだけではわかりません。適切な可視化ツールを使用すれば、データを短期間でより深く理解できます。表やグラフを作成する前に、何を表示するかを決めなけ

Archived

ればなりません。たとえば、グラフであれば、主要業績評価指標 (KPI)、関係性、比較、分布、構成などの情報を伝えることができます。

AWS では、大規模なデータの可視化や分析に使えるサービスをいくつか提供しています。

Amazon SageMaker には、データの可視化や分析に利用できる、ホストされた Jupyter ノートブック環境が用意されています。オープンソースのウェブアプリケーションである Project Jupyter を使用すると、視覚化やナラティブテキストの作成の他、データクリーニング、データ変換、数値シミュレーション、統計的モデリング、データの可視化などを行うことができます。

Amazon Athena は、ANSI SQL 演算子と関数を使用して Amazon S3 内のデータをクエリするために使用できる、完全マネージド型インタラクティブクエリサービスです。Amazon Athena はサーバーレスなため、クエリの需要に合わせてシームレスに拡張できます。

Amazon Kinesis Data Analytics には、ストリーミングデータを分析し、リアルタイムで分析する機能があります。これにより、実用的なインサイトを獲得できます。このサービスは、受信データのボリュームとスループットに合わせて自動的に拡張します。

Amazon QuickSight は、ダッシュボードや可視化を提供するクラウド型のビジネスインテリジェンス (BI) サービスです。このサービスは、数百人のユーザーをサポートするために自動的に拡張し、ストーリーボード作成のための安全な共有とコラボレーションを提供します。さらに、このサービスには、異常の検出、予測、What-if 分析を提供する ML 機能が組み込まれています。

以下のベストプラクティスが適用されます。

- データをプロファイルする (カテゴリ別、順序、量的な可視化)
- ユースケース (データサイズ、データの複雑さ、リアルタイムとバッチなど) に合ったツールやツールの組み合わせを選択する
- データ分析パイプラインを監視する
- データの前提を検証する

Archived

特徴量エンジニアリング

視覚化や分析を通じてデータを調査し、理解した後に、特徴量エンジニアリングを行います。データの固有の属性はすべて特徴と見なされます。たとえば、顧客離れを予測するという問題の解決策を設計するには、一定期間にわたって収集した顧客データから始めます。顧客データでは、顧客の所在地、年齢、所得レベル、最近の購入などの特徴量 (属性とも呼ばれる) を取得します。

特徴量エンジニアリングとは、機械学習や統計的モデリングを用いて予測モデルを作成する際に、変数を選択して変換するプロセスです。通常、特徴量エンジニアリングでは、特徴の作成、特徴の変換、特徴の抽出、特徴の選択を行います。

- **特徴の作成**では、直近の問題と関連するデータセット内の特徴を特定します。
- **特徴の変換**では、不足している特徴や有効でない特徴の置き換えを管理します。特徴のデカルト積の形成、非線形変換 (数値変数のカテゴリへのビン分割など)、ドメイン固有の特徴の作成などの手法があります。
- **特徴の抽出**は、既存の特徴から新しい特徴を作成するプロセスであり、通常は特徴の次元を減らすことを目的としています。
- **特徴の選択**では、データセットから無関係な特徴や冗長な特徴をフィルタリングします。通常、分散または相関のしきい値を観測して、削除する特徴を決定します。

Amazon SageMaker では、Spark や scikit-learn プリプロセッサを利用した Jupyter ノートブック環境を提供しており、特徴量エンジニアリングやデータの変換に利用できます。Amazon SageMaker からは、AWS Glue や Amazon EMR などの ETL サービスを利用して特徴の抽出や変換のジョブを実行することもできます。さらに、Amazon SageMaker Inference Pipeline を使用し、既存のデータ処理機能を再利用できます。

Amazon SageMaker Processing では、Amazon SageMaker のすべてのセキュリティおよびコンプライアンス機能を備えた、完全マネージド型の環境で特徴量エンジニアリング (およびモデル評価) の分析ジョブを大規模に実行できます。Amazon SageMaker Processing では、組み込みのデータ処理コンテナを使用したり、独自のコンテナを持ち込んでマネージド型インフラ

Archived

ストラクチャで実行するカスタムジョブを送信したり、柔軟な対応が可能になります。送信後は、Amazon SageMaker がコンピューティングインスタンスを起動し、入力データを処理して分析し、完了したらリソースを解放します。

以下のベストプラクティスが適用されます。

- ドメインのエキスパートの力を借りて、特徴の実現可能性と重要性を評価する
- 冗長な特徴と無関係な特徴を除去する (データのノイズを減らし、相関関係を減らす)
- コンテキスト全体で一般化できる特徴から始める
- モデルの構築時に反復処理する (新しい特徴、特徴の組み合わせ、新しいチューニング目標)

モデルのトレーニング

この段階では、問題に適した機械学習アルゴリズムを選択して、ML モデルをトレーニングします。トレーニングでは、アルゴリズムに学習用のトレーニングデータを提供し、モデルパラメータを設定してトレーニングプロセスを最適化します。

通常、トレーニングアルゴリズムでは、トレーニングエラーや予測精度などのいくつかのメトリクスを計算します。これらのメトリクスは、モデルが効率的に学習し、未知のデータについての予測が適切に一般化行されているかどうかを判断するのに役立ちます。アルゴリズムによって報告されるメトリクスは、ビジネスに関する問題や使用した ML の手法によって異なります。たとえば、分類アルゴリズムは、真または偽陽性および真または偽陰性をキャプチャする混同行列によって測定でき、回帰アルゴリズムは、二乗平均平方根誤差 (RMSE) によって測定できます。

ML アルゴリズムとその結果として得られるモデルアーキテクチャの動作を制御するために調整できる設定は、ハイパーパラメータと呼ばれます。ML アルゴリズムにおけるハイパーパラメータの数や種類は、モデルごとに異なります。一般的に使用されるハイパーパラメータの例としては、学習率、エポック数、隠れ層、隠れユニット、活性化関数などがあります。ハイ

Archived

パーパラメータのチューニング (最適化) とは、最適なモデルアーキテクチャを選択するプロセスのことです。

Amazon SageMaker には、Amazon S3 に用意して保存しておいたトレーニングデータを使ってトレーニングできる、定評のある組み込みアルゴリズムがいくつか用意されています。また、独自のカスタムアルゴリズムを Amazon SageMaker でトレーニングすることもできます。カスタムアルゴリズムは Amazon ECS と Amazon ECR を使ってコンテナ化する必要があります。

アルゴリズムを選択したら、API 呼び出しを使用して Amazon SageMaker でトレーニングを開始します。トレーニングは、単一のインスタンスの他、インスタンスの分散型クラスターで行うことができます。トレーニングプロセスに必要なインフラストラクチャ管理は Amazon SageMaker で管理されるため、差別化につながらない面倒な作業の負担が軽減します。

Amazon SageMaker では、ハイパーパラメータチューニングジョブによるモデルの自動チューニングも可能です。設定が完了すると、ハイパーパラメータチューニングジョブは、アルゴリズムと指定したハイパーパラメータの範囲を使用して、データセットで多くのトレーニングジョブを実行し、モデルの最適なバージョンを見つけ出します。次に、選択されたメトリクスでで測定し、最高パフォーマンスのモデルを得たハイパーパラメータを選択します。Amazon SageMaker の自動モデルチューニングは、組み込みアルゴリズム、カスタムアルゴリズム、ML フレームワーク用の Amazon SageMaker の事前構築済みコンテナで使用できます。

Amazon SageMaker Debugger は、定期的な間隔でトレーニングジョブの状態をキャプチャするデータを監視、記録、分析し、ML トレーニングプロセスを可視化します。また、トレーニング中にキャプチャしたデータをインタラクティブに調査する機能や、トレーニング中に検出されたエラーに対するアラート機能も備わっています。たとえば、勾配値が大きすぎるまたは小さすぎるなどの一般的なエラーを自動的に検出してアラートを送信します。

Amazon SageMaker Autopilot は、データの前処理、アルゴリズムの選択、ハイパーパラメータのチューニングを自動的に処理することで、ML のトレーニングプロセスを簡素化します。 トレーニングデータを表形式で提供するだけで、分類モデルや回帰モデルを構築できます。この機能では、データプリプロセッサ、アルゴリズム、アルゴリズムパラメータ設定のさまざまな組み合わせで複数の ML ソリューションを調査し、最も正確なモデルを見つけ出しま

Archived

す。Amazon SageMaker Autopilot は、ネイティブにサポートされる高性能アルゴリズムの中から最適なアルゴリズムを選択します。さらに、モデル品質を最適にするため、これらのアルゴリズムのさまざまなパラメータ設定を自動で試行します。その後、本番環境に最適モデルを直接デプロイしたり、複数の候補を評価して精度、遅延、モデルサイズなどのバランスを調整できます。

AWS Deep Learning AMI と **AWS Deep Learning Containers** を使用し、複数のオープンソースの ML フレームワークをインフラストラクチャのトレーニングに利用できます。AWS 深層学習 AMI には、TensorFlow、PyTorch、Apache MXNet、Chainer、Gluon、Horovod、Keras などの定評のある深層学習フレームワークやインターフェイスがプリインストールされています。AMI やコンテナは、ML のパフォーマンスに最適化された強力なインフラストラクチャ上で起動できます。

Amazon EMR には分散クラスター機能があり、クラスターのローカルまたは Amazon S3 に保存されているデータに対してトレーニングジョブを実行するオプションもあります。

以下のベストプラクティスが適用されます。

- モデルをトレーニングする前にモデルテスト計画を生成する
- トレーニングに必要なアルゴリズムの種類を明確に理解する
- トレーニングデータがビジネスに関する問題を代表するものであることを確認する
- トレーニングのデプロイにマネージド型サービスを使用する
- 段階的トレーニングや転移学習戦略を適用する
- 目標メトリクスで測定した結果が大幅に改善していない場合には、すぐにトレーニングジョブを中止し、過剰適合を回避してコストを削減する
- モデルのパフォーマンスは時間の経過とともに低下する可能性があるため、トレーニングメトリクスを注意深く監視する
- マネージド型サービスを活用してモデルチューニングを自動化する

Archived

モデルの評価とビジネスの評価

モデルのトレーニングが完了したら、そのパフォーマンスと精度を評価して、ビジネス目標を達成できるかどうかを判断します。さまざまな方法で複数のモデルを生成し、それぞれのモデルの効果を評価できます。たとえば、モデルごとに異なるビジネスルールを適用し、さまざまな対策を適用して各モデルの適合性を判断できます。また、モデルが特異度ではなくより高い感度が必要か、または感度ではなく高い特異度が必要かについても評価できる場合があります。マルチクラスモデルの場合、各クラスのエラー率を個別に評価します。

モデルの評価には、履歴データ (オフライン評価) またはライブデータ (オンライン評価) を使用できます。オフライン評価では、トレーニングされたモデルは、ホールドアウトセットとして確保されたデータセットの一部を使って評価されます。このホールドアウトデータは、モデルのトレーニングや検証には使用されず、最終的なモデルでのエラー評価のみに使用されます。ホールドアウトデータの注釈には、評価が意味のあるものになるように、高い精度が必要です。追加のリソースを割り当てて、ホールドアウトデータの精度を検証します。

モデルトレーニングに使用される AWS サービスにも、この段階での役割があります。モデルの検証は、Amazon SageMaker、AWS Deep Learning AMI、Amazon EMR を使用して実行できます。

評価結果に基づいて、データ、アルゴリズム、またはその両方を微調整できます。データを微調整する際には、データクレンジング、準備、特徴量エンジニアリングの概念を適用します。

以下のベストプラクティスが適用されます。

- 成功を測定する方法を明確に理解する
- プロジェクトに対するビジネス上の期待値に対するモデルのメトリクスを評価する
- 本番環境のデプロイ (モデルのデプロイとモデル推論) を計画して実行する

モデルのトレーニング、調整、テストが完了したら、モデルを本番環境にデプロイし、モデルに対する推論 (予測) を行うことができます。

Archived

Amazon SageMaker には、デプロイや推論のための幅広いオプションが用意されているため、本番環境の ML モデルをホスティングするための AWS のサービスとして推奨されます。

モデルトレーニングと同様に、API 呼び出しを使用して Amazon SageMaker でモデルをホストできます。モデルを 1 つのインスタンスで、または複数のインスタンスにまたがってホストできます。同じ API を使用して自動スケーリングを設定できるため、ML モデルのさまざまな推論要求に対応できます。モデルのホストに必要なインフラストラクチャ管理は Amazon SageMaker で完全に管理されるため、差別化につながらない手間のかかる作業を軽減できます。

Amazon SageMaker Inference Pipeline では、推論パイプラインをデプロイできるため、未加工の入力データを渡し、事前処理、予測、完全な事後処理を実行し、推論リクエストをリアルタイムとバッチで処理できます。推論パイプラインは、ML フレームワーク、組み込みのアルゴリズム、あるいは Amazon SageMaker で使用できるカスタムのコンテナから構成されます。SparkML で利用できる機能トランスフォーマー一式と Scikit-learn のフレームワークコンテナを使って、機能のデータ処理および特徴量エンジニアリングのパイプラインを構築したり、これらを推論パイプラインの一部としてデプロイしてデータ処理コードを再利用したりできるなど、ML の処理の管理が簡単になります。これらの推論パイプラインは完全マネージド型であり、データサイエンスプロセスの一部として事前処理、予測、事後処理を組み合わせることができます。

Amazon SageMaker Model Monitor は、本番環境の ML モデルを継続的に監視します。ML モデルの本番環境へのデプロイ後に、実際のデータがモデルのトレーニングに使用したデータと異なってくると、モデルの品質にばらつきが生じ、最終的には精度の低いモデルになってしまうことがあります。Model Monitor は、時間の経過とともにモデルのパフォーマンスを低下させる可能性のあるデータドリフトなどの逸脱を検出し、ユーザーに修復アクションを実行するよう警告します。

Amazon SageMaker Neo を使用して ML モデルを一度トレーニングすると、そのモデルをクラウドやエッジの任意の場所で実行できます。Amazon SageMaker Neo は、コンパイラとランタイムで構成されます。コンパイル API は、さまざまなフレームワークからエクスポートされたモデルを読み取ってフレームワークに依存しない表現に変換し、最適化されたバイナリコー

Archived

ドを生成します。各ターゲットプラットフォームのランタイムは、コンパイル済みモデルをロードして実行します。

Amazon Elastic Inference では、Amazon EC2 と Amazon SageMaker インスタンスに低コストの GPU を使用したアクセラレーションをアタッチでき、深層学習の推論実行コストを削減できます。スタンドアロンの GPU インスタンスはモデルトレーニング用に設計されており、通常、推論用としてはサイズが大きすぎます。トレーニングジョブは数百のデータサンプルを同時にバッチ処理しますが、ほとんどの推論はリアルタイムでの単一の入力を処理するため、GPU コンピューティングをわずかしか消費しません。Amazon Elastic Inference を使うと、Amazon EC2 または Amazon SageMaker インスタンスタイプに、コードを変更せずに適切なサイズの GPU を使用した推論アクセラレーションをアタッチすることで、この問題を解消できます。

Elastic Inference は TensforFlow や Apache MXNet といったいくつかの深層学習フレームワークでネイティブにサポートされていますが、Open Neural Network Exchange (ONNX) を使用してモデルをエクスポートし、MXNet にインポートすると、他の深層学習フレームワークでも使用できます。

以下のベストプラクティスが適用されます。

- 本番稼働中のモデルパフォーマンスをモニタリングし、ビジネスにおける期待と比較する
- トレーニング中と本番稼働中のモデルパフォーマンスの違いをモニタリングする
- モデルパフォーマンスで変化が検出されたら、モデルを再トレーニングする (たとえば、新しい競合他社の出現により販売予測が変化し、今後の予測が変わる可能性がある)
- データセット全体の推論を取得する場合は、ホスティングサービスの代わりにバッチ変換を使用する
- 本番稼働用バリアントを活用し、A/B テストで新しいモデルのバリエーションをテストする

Archived

一般的な設計の原則

Well-Architected フレームワークは、クラウドにおける機械学習ワークロードの適切な設計を可能にする一般的な設計の原則を提供します。

- **高品質データのデータセットを利用可能にすることで俊敏性を実現する**

  データサイエンスのワークロードには、配信パイプラインのすべてのフェーズでライブデータまたはバッチデータへのアクセスが必要です。データ検証と品質管理でデータへのアクセスを有効にするメカニズムを実装します。

- **シンプルなものから開始し、実験を通して進化する**

  少ない機能から始めると、複雑なモデルから開始したために機能の影響を追跡できなくなるという間違いを回避できます。シンプルなモデルを選択し、プロセス全体で一連の実験を実行します。

- **モデルトレーニングと評価をモデルホスティングから疎結合化する**

  モデルトレーニング、モデル評価、モデルホスティングの各リソースを分離することで、データサイエンスのライフサイクルにおける特定の段階に最適なリソースを選択します。

- **データドリフトを検出する**

  経時的にデータドリフトを管理するには、モデルが本稼働状態になった後で推論の精度を継続的に測定します。通常、ML で使用されるデータは複数のソースから取得され、そのデータの形状と意味は、アップストリームのシステムやプロセスの変更に伴い変化する可能性があります。適切なアクションを実行できるように、こうした変化を検出するメカニズムを確保します。

- **トレーニングと評価パイプラインを自動化する**

Archived

自動化により、自動モデルトレーニングとモデルアーティファクトの作成をトリガーできるため、複数のエンドポイント環境に一貫してデプロイできます。モデルの再トレーニングアクティビティがトリガーされるように自動化を適用すると、手作業が軽減され、人為的エラーが減少し、モデルパフォーマンスの継続的な改善に役立ちます。

- **上位の抽象化を選択して成果を迅速に達成する**

  適切な AI/ML サービスを選択する際は、まず、適性に対する大まかなサービスを評価します。その後、ビジネス目標を迅速に満たして差別化につながらない高負荷の作業を除外し、開発コストを削減するメカニズムを評価する必要があります。

シナリオ

以下に、AWS での機械学習ワークロードの設計とアーキテクチャに影響する一般的なシナリオをいくつか紹介します。各シナリオには、設計の一般的な推進要因と、各シナリオの実装方法を示すリファレンスアーキテクチャが含まれています。

AWS の AI サービスを使用してインテリジェントなアプリケーションを構築する

機械学習スタックにおける AWS の AI サービスレイヤーは、最小限の開発労力とわずかな作業時間で、既存または新しいアプリケーションに AI 機能を追加したい組織に適しています。このレイヤーのサービスは、フルマネージド型ですぐに使用できるコンピュータビジョン、音声、自然言語、チャットボット 機能を提供します。

開発者がこれらのサービスを使用する場合、ML プロセスのデータ準備、データ分析、モデルトレーニング、評価フェーズを管理する必要がなくなります。代わりに、シンプルな API 呼び出しを使用してこれらの機能をアプリケーションに統合できます。

Archived

Amazon Comprehend は ML を使用して非構造化テキストデータからインサイトや関係性を見つけ出す自然言語処理 (NLP) サービスです。サービスは最初にテキストの言語を識別します。次に、キーフレーズ、場所、人物、ブランド、イベントを抽出します。このサービスではトークン分割と品詞を使用してテキストを分析します。また、テキストの肯定性または否定性の度合いが理解できるため、トピックごとにテキストファイルのコレクションを自動的に整理できます。また、Amazon Comprehend の AutoML 機能を使用して、組織のニーズに合ったエンティティおよびテキスト分類モデルのカスタムセットを独自に調整して構築することもできます。

Amazon Lex は、音声とテキストを使用してアプリケーションに会話型インターフェイスを構築するためのサービスです。Amazon Lex では、音声をテキストに変換する自動音声認識 (ASR) と、テキストの意図を認識する自然言語理解 (NLU) という高度な深層学習機能を利用できます。これらの機能により、ユーザーにとって非常に魅力的で、リアルな会話を再現するアプリケーションを構築できます。

Amazon Polly は、テキストをリアルな音声に変換するサービスです。このサービスを使用して発話するアプリケーションを作成し、まったく新しいカテゴリの音声対応製品を構築できます。Amazon Polly は、高度な深層学習技術を使用して人間の声のような音声を合成するテキスト読み上げサービスです。

Amazon Rekognition を使用すると、アプリケーションに画像と動画の分析を簡単に追加できます。Amazon Rekognition API に画像または動画を提供すると、サービスは物体、人物、テキスト、シーン、活動を識別したり、不適切なコンテンツを検出したりできるようになります。Amazon Rekognition は、提供される画像や動画について非常に正確な顔分析と顔認識も提供しています。ユーザー検証、人数のカウント、公共安全などのさまざまなユースケースで顔を検出、分析、比較できます。

Amazon Transcribe は、アプリケーションに音声のテキスト変換機能を簡単に追加できる自動音声認識 (ASR) サービスです。Amazon Transcribe API を使用すると、Amazon S3 に保存されている音声ファイルを分析し、文字に起こされた音声のテキストファイルをサービスから返すことができます。また、ライブ音声ストリームを Amazon Transcribe に送信し、リアルタイムで文字に起こされたテキストのストリームを受信することもできます。

Archived

Amazon Translate は、高速で高品質な言語翻訳を低価格で提供するニューラル機械翻訳サービスです。ニューラル機械翻訳は、深層学習モデルを使用して、従来の統計ベースやルールベースの翻訳アルゴリズムよりも正確で自然な翻訳を提供する形態の自動言語翻訳です。Amazon Translate を使用すると、世界中のユーザー向けにウェブサイトやアプリケーションなどのコンテンツをローカライズできるほか、大量のテキストを効率的に簡単に翻訳できます。

AWS の AI サービスからの回答には、特定の結果に対する AI サービスの信頼度を表す信頼スコアも含まれます。すべての ML システムは本質的に確率的であるため、信頼スコアは、システムが結果にどの程度の信頼を置いているかの指標と考えることができます。AI サービスを使用する場合は、特定のユースケースで許容できる適切なしきい値を設定してください。マルチクラスモデルの場合は、クラスのエラー率に基づいて設定されたクラスごとのしきい値を使用します。たとえば、Amazon Rekognition を使用してイベントでの集団の関心を計測する場合は信頼スコアのしきい値を小さくする必要があっても、医療画像の分析に同じサービスを使用する場合は高いしきい値が必要になる可能性があります。医療画像分析など、結果の影響力が大きいドメイン固有のユースケースの場合、医療専門家による二次的な検証が必要になる場合もあります。

AI サービスはサーバーレスの従量課金制であるため、ビジネスに合わせてサービスを拡張し、エントリフェーズやピーク時以外の時間帯にコストを抑えることができます。AI サービスは、サーバーレスという特性により、AWS Lambda を使用したイベント駆動型アーキテクチャの候補として理想的です。AWS Lambda では、実質的にあらゆるタイプのアプリケーションやバックエンドサービスに対して、管理タスクを実行せずにコードを実行できます。使用したコンピューティング時間に対してのみお支払いいただきます。コードが実行中でなければ料金はかかりません。

ユースケースの例として、カスタマーエクスペリエンスとエンゲージメントの向上をビジネス目標にして、小売店の顧客人口統計データをキャプチャおよび分析します。顔画像をキャプチャして処理する際は、そのデータを保護する安全策を実装し、そのデータを使用する前に適切な信頼レベルを適用する必要があります。図 2 に示すリファレンスアーキテクチャは、顔分析に Amazon Rekognition、顔属性データの分析に Amazon Athena、分析の可視化に Amazon QuickSight を使用して実装しています。

Archived

顔分析技術の使用は、公民権を保護する法律など、すべての法律に準拠する必要があります。AWS のお客様には、この技術を使用するにあたり、適用されるすべての法律に従う責任があります。AWS 適正利用規約 (AUP) は、Amazon Rekognition を含む AWS のサービスを法律に違反する形で使用することを禁止しています。また、AUP に違反するお客様は、当社のサービスを利用できなくなります。

リファレンスアーキテクチャ

図 2 - 顧客人口統計の分析ソリューション

このリファレンスアーキテクチャには、以下のプロセス概要が含まれます。

- 画像を一時的に保存する Amazon S3 バケットを作成し、バケットの暗号化を有効にしてイメージを保護します。AWS IAM を使用して S3 バケットへのアクセスを制限します。アップロードプロセスには書き込み専用アクセス権限 (パブリック読み取りなし)、AWS Lambda 関数には読み取り専用アクセス権限が付与されます。CloudTrail を使用して、S3 バケットのデータイベントを別の S3 バケットに記録できるようにし、バケットに関連するすべてのアクティビティのログを収集できるようにします。
- 小売店でキャプチャされた顧客の画像は Amazon S3 バケットにアップロードされるため、処理後に画像が自動的に削除されるようにライフサイクルポリシーを確立する必要があります。

Archive

- Amazon S3 にアップロードされた各画像により AWS Lambda 関数がトリガーされ、顔分析を使用して年齢、性別、感情などの人口統計データを理解できます。Lambda 関数は Amazon Rekognition サービスを呼び出して、画像から顔属性を抽出します。この属性には、顧客の年齢、性別、感情 (喜んでいる、落ち着いている、怒っているなど) が反映されています。推論に関する情報も、信頼レベルとともに属性に含まれます。
- 人口統計データは 2 つ目の Amazon S3 バケットに .csv 形式で保存されます。安全に保存された一意のキーを使用して .csv ファイルを暗号化します。AWS Key Management Service (AWS KMS) などのサービスを使用してこれらのキーを管理および保存できます。
- Amazon Athena は、この csv ファイルから人口統計データを読み取ってロードし、クエリを実行します。Amazon Athena では、Amazon S3 で AWS KMS を使用することで、ソースデータとクエリ結果の両方に対してデータの暗号化をサポートします。信頼レベルを適切に使用するために、Amazon Athena のビューを使用し、十分な信頼度を持つユースケースのビューのみに検索を制限します。
- Amazon QuickSight でお客様のインサイトをダッシュボードに構築します。AWS IAM を使用して Amazon QuickSight ダッシュボードおよび Amazon Athena クエリへのアクセスを適切な担当者に制限し、すべてのアクセスが安全に記録されるようにします。

この例で対象とするオブジェクトは画像であり、Amazon Rekognition を使用して画像を分析します。顔画像の保護、画像の自動削除、推論に使用される信頼レベルの使用と記録のために、保護対策が講じられています。信頼度の低い結果を除外し、信頼レベルが正しく使用されるようにします。このアーキテクチャは、適切な AI サービスを使用して、テキストや音声などのさまざまなオブジェクトタイプを分析するために使用できます。たとえば、Amazon Transcribe を使用して音声ファイルを文字に起こし、Amazon Comprehend を使用して非構造化テキストを分析できます。

高度化

個々の AI サービスは、画像分析や文字起こしなどの特定のタスクを処理するように事前にトレーニングされていますが、これらのサービスのサーバーレスという特性により、AWS Step Functions を使用して複数のサービスを調整し、高度なソリューションを構築できます。AWS

Archived

メディア分析ソリューションはその 1 例です。このソリューションを使用して、既存のメディアファイルから検索可能なカタログを簡単に分析、理解、構築できます。以下のリファレンスアーキテクチャでは、複数の AI サービス (Amazon Rekognition、Amazon Transcribe、Amazon Comprehend) を使用して、メディアからメタデータを分析および抽出します。抽出されたメタデータはインデックス化されて Amazon Elasticsearch に保持されるため、メディアコンテンツ全体を検索できます。

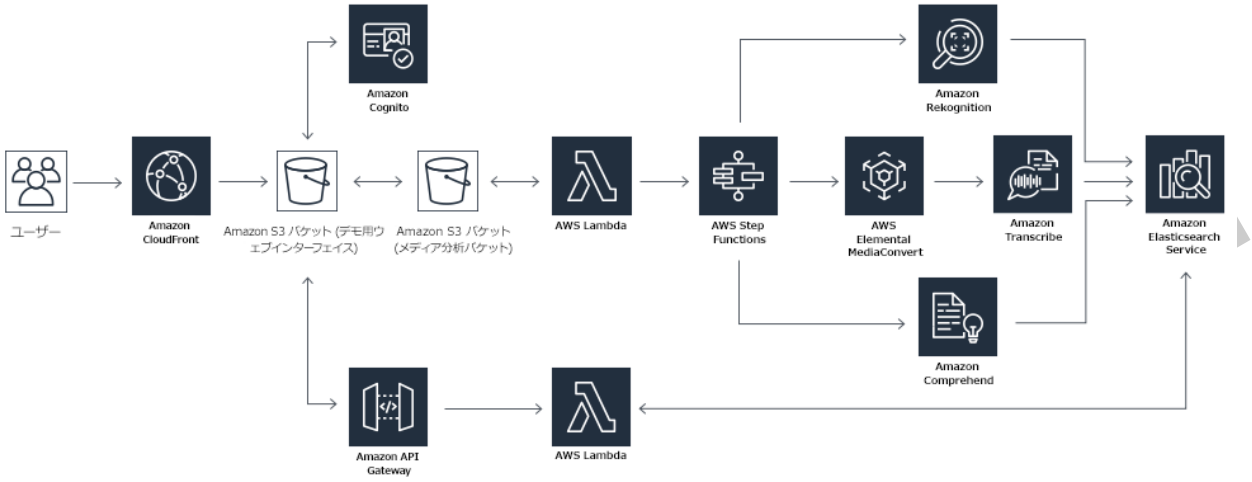

図 3 - メディア分析ソリューションのリファレンスアーキテクチャ

このリファレンスアーキテクチャには、以下のプロセス概要が含まれます。

- Amazon S3 バケットにウェブインターフェイスをデプロイすると、シンプルなウェブインターフェイスで小さなメディアファイルの分析をすぐに開始できます。Amazon CloudFront を使用して、Amazon S3 バケットのコンテンツへのアクセスを制限します。
- アップロードされたメディアファイルは、Amazon API Gateway の RESTful API、API リクエストを処理する AWS Lambda 関数へと流され、Amazon Cognito ユーザープールはメディアファイルとの安全なやり取りを実現します。
- AWS Step Functions ステートマシンは、メディア分析プロセスを調整します。2 つ目の Lambda 関数は、Amazon Rekognition、Amazon Transcribe、Amazon Comprehend などのマネージド型 AI サービスを使用して分析とメタデータの抽出を実行します。

- MP4 動画ファイルがアップロードされると、AWS Elemental MediaConvert が音声を抽出し、Amazon Transcribe および Amazon Comprehend で分析できるようにします。
- メタデータの結果は S3 バケットに保存され、Amazon Elasticsearch Service (Amazon ES) クラスターでインデックス化されます。

画像分析、言語翻訳、文字起こしなどの特定のユースケースを解決する AI サービスにより、機械学習や深層学習に関する広範な知識がなくても、強力でインテリジェントな機能を構築できます。その結果、ビジネス目標に向けた実験と評価が迅速化され、市場投入までの時間が短縮されます。この例の場合、エラーの影響は小さいため、どの ML アプローチに対しても低い信頼レベルを使用できます。

データで AI サービスを使用

これまで説明した AI サービスは事前トレーニング済みのモデルに基づいていますが、AWS ではお使いのデータでトレーニングされた ML モデルを返す AI サービスも提供しています。

Amazon Personalize は、顧客が提供するユーザー項目のインタラクションデータに基づいて、アプリケーション向けにカスタマイズされた専用のパーソナライゼーションの推奨事項を作成できるフルマネージド型サービスです。 アプリケーション内でタイムリーに推奨される動画や、適切なタイミングで送信されるパーソナライズされた通知 E メールなど、お使いのデータに基づいてパーソナライズされたエクスペリエンスにより、顧客に関連性の高い体験が提供され、多くの場合、企業の収益の向上につながります。

Amazon Forecast は、顧客が提供する履歴データに基づいて精度の高い予測を行うフルマネージド型サービスです。 このサービスでは深層学習を使用して複数のデータセットから学習し、データに最適なさまざまなアルゴリズムを自動で試行できます。製品需要の予測、クラウドコンピューティングの使用量、財務計画、サプライチェーン管理システムでのリソース計画など、さまざまなユースケースに使用できます。

Archived

マネージド型 ML サービスを使用してカスタム ML モデルを構築する

マネージド型サービスのアプローチでは、所有するデータを使用して ML モデルを構築およびデプロイし、ビジネス価値の予測的モデルおよび規範的モデルを作成できます。マネージド型 ML サービスを使用する場合、開発チームとデータサイエンスチームは、エンドツーエンドの ML プロセスに対するデータ準備、データ分析、モデルトレーニング、モデル評価、モデルホスティングの各段階を管理する必要があります。

Amazon SageMaker は、データをラベリングし準備する、アルゴリズムを選択する、モデルをトレーニングする、デプロイに向けモデルを調整、最適化する、予測する、アクションをとるという ML ワークフロー全体に対応するフルマネージド型サービスです。開発者やデータサイエンティストが、差別化要素ではないインフラストラクチャ管理の負担を負うことなく ML モデルを構築できるように、Amazon SageMaker には以下の機能が備わっています。

- **トレーニングデータを収集し準備**

  Amazon SageMaker Ground Truth を使用してデータにラベルを付け、多くの一般的な ML 問題に対して複数の事前構築済みノートブックを活用します。

- **機械学習アルゴリズムのサポート**

  複数の組み込みの高性能アルゴリズムから選択するか、独自のアルゴリズムを使用します。また、ユースケースに適したアルゴリズムを AWS Marketplace で探すこともできます。

- **モデルのトレーニング**

Archived

高性能トレーニングクラスターをセットアップ、管理、終了する API 呼び出しを使用して、独自のデータで ML モデルをトレーニングします。単一のインスタンスを使用するようにトレーニングクラスターを設定するか、分散トレーニングをサポートするように複数のインスタンスを選択します。Amazon SageMaker Debugger は、トレーニング実行でのデータの取得と分析を自動化し、トレーニングプロセスをリアルタイムに把握します。

- **モデルの最適化**

  Amazon SageMaker でモデルをトレーニングしてから、Amazon SageMaker Neo を使用して他の ML フレームワーク用に最適化します。

- **本番環境でのモデルのデプロイ**

  API 呼び出しを使用して、選択した Auto Scaling 対応インフラストラクチャにトレーニング済みモデルをデプロイします。

- **デプロイされたモデルを監視**

  本番環境で ML モデルを継続的に監視して、モデルのパフォーマンスを低下させる可能性のあるデータドリフトなどの逸脱を検出し、修復アクションを自動化します。

AWS Lambda はイベント駆動型アーキテクチャをサポートし、データの取り込みから予測に至る ML プロセスの複数の段階を連携させるツールです。

リファレンスアーキテクチャ

Amazon SageMaker、Amazon Kinesis Data Streams、Amazon S3、AWS Lambda を使用したエンドツーエンドの ML プロセスの自動化については、*AWS* で *Amazon SageMaker* とデータレイクを使用した予測データサイエンスソリューションに関する以下のリファレンスアーキテクチャ (図 4) で説明されています。

Archived

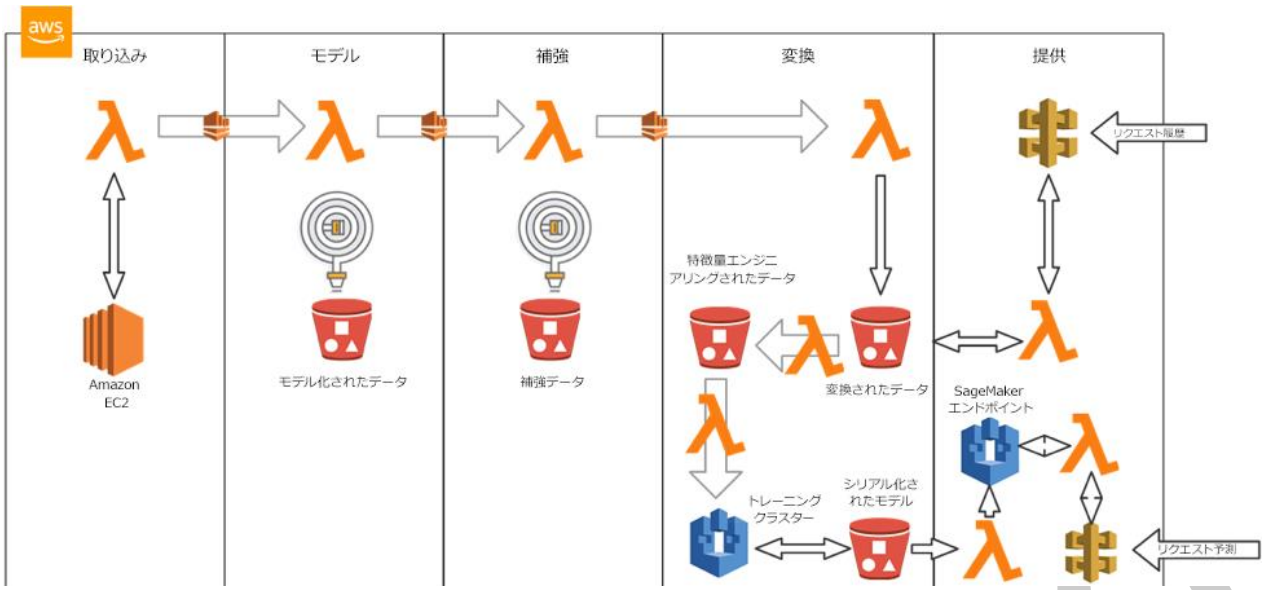

図 *4 - AWS* で *Amazon SageMaker* とデータレイクを使用した予測データサイエンス

このリファレンスアーキテクチャには、以下の概要要素が含まれます。

- Amazon S3 は、未加工データ、モデル化データ、拡張データ、変換データを保持するデータレイクとして使用されます。
- Amazon Kinesis Data Streams では、取り込み、モデル化、拡張、変換の各段階にわたり新しいデータをリアルタイムで処理できます。
- データ変換コードは AWS Lambda でホストされ、消費および ML モデルトレーニング用に未加工データを準備し、データの入出力を変換します。
- AWS Lambda は新しいモデルの REST エンドポイントを構築、管理、作成する Amazon SageMaker API 呼び出しを自動化します。API 呼び出しはスケジュールに基づいて実行されるか、データレイク内のデータ変更によりトリガーされ実行されます。

このアーキテクチャは、顧客データを使用する ML モデルの自動化されたトレーニングと改善を継続的に提供します。差別化につながらない手間のかかるインフラストラクチャ管理は必要ありません。

Archived

データ変換コードは AWS Lambda でホストされます。データ変換コードは、Amazon SageMaker ノートブックインスタンスでも実行できます。ただし、大規模なデータ変換の場合は特に、これらのオプションがすべての状況で正しい選択肢であるとは限りません。

データ処理のマネージド型 ETL サービス

クレンジング、検出、大規模な特徴量エンジニアリングなどのデータ処理アクティビティは、他の便利なユーティリティの中でも特に、データ検出の SQL サポートを提供する Apache Spark などのツールに最適です。AWS で、Amazon EMR は Spark クラスターの管理を容易にし、スポットインスタンス料金でコストを最小限に抑え、伸縮自在なスケーリングなどの機能を有効にします。

Amazon SageMaker ノートブックは、外部の Amazon EMR クラスターに接続し、Apache Spark を使用して伸縮自在でスケーラブルなクラスターでのデータ処理を可能にします。その後、Amazon SageMaker トレーニングおよびデプロイメント API を使用してモデルをトレーニングし、デプロイできます。

たとえば、消費者行動の深い理解に基づいて消費者にターゲットを絞ったマーケティングのビジネスユースケースを想定してください。Amazon Pinpoint は、E メール、テキスト、SMS などの複数のエンゲージメントチャネルを通じて消費者にターゲットを絞ったメッセージを送信できるマネージド型サービスです。ターゲットを絞ったキャンペーンの例として、プロモーションアラートと顧客定着キャンペーン、注文確認やパスワードリセットメッセージなどのトランザクションメッセージなどがあります。ただし、メッセージの送信先として正しい顧客や顧客層を特定することが重要です。ML を使用すれば、過去の消費者購買パターンに基づいて将来の購買行動を予測できます。その後、予測された購買行動を使用し、Amazon Pinpoint でターゲットを絞ったキャンペーンを配信できます。

Archived

リファレンスアーキテクチャ

このリファレンスアーキテクチャは、ML のさまざまな段階で Amazon EMR、Apache Spark、Amazon SageMaker を使用し、Amazon Pinpoint でターゲットを絞ったマーケティングメッセージを送信する方法を示しています。

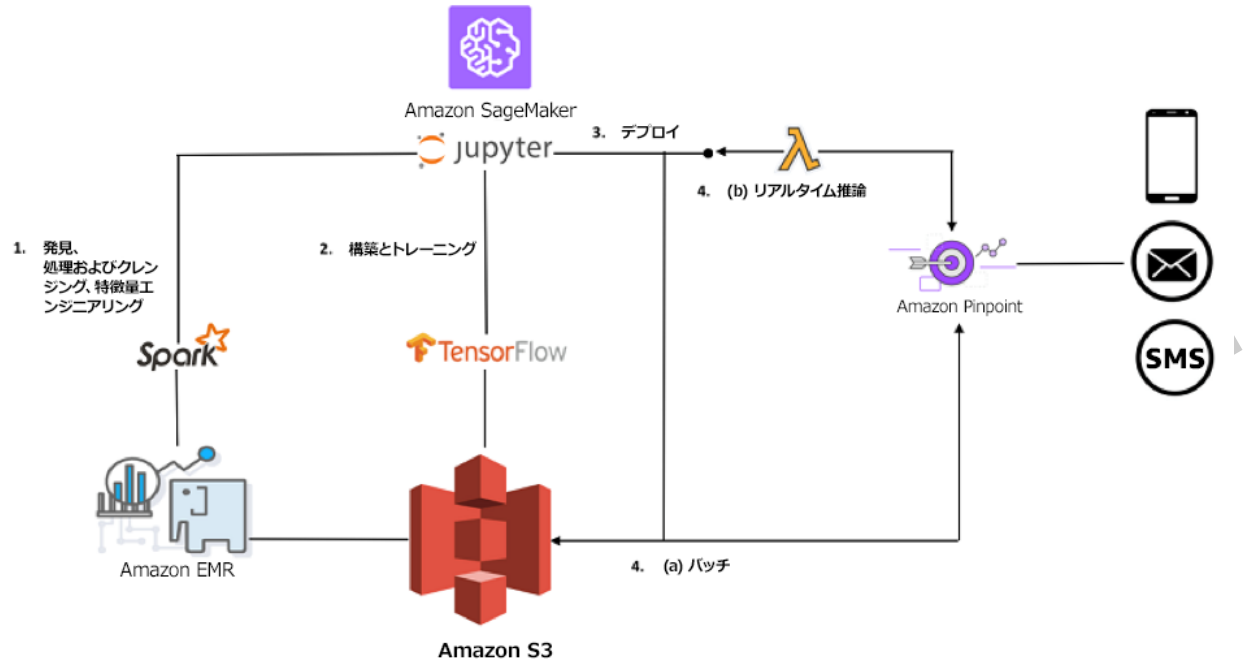

図 *5 - Amazon SageMaker の ML による Amazon Pinpoint* キャンペーンドライブ

このリファレンスアーキテクチャには、以下の概要要素が含まれます。

- 大量データボリュームを保持するデータレイクとして Amazon S3 を使用します。
- 外部 Amazon EMR クラスターに対して実行するように Amazon SageMaker ノートブックを設定します。データクレンジング、処理、検出、および特徴量エンジニアリングは、EMR クラスターで Apache Spark を使用して行われます。変換されたデータは Amazon S3 に保存されます。
- Amazon SageMaker は、変換されたデータを使用、分散型トレーニング機能を活用して、カスタムモデルをトレーニングします。

- Amazon SageMaker は、トレーニングされたモデルの Auto Scaling API エンドポイントを作成します。
- API エンドポイントを使用して、バッチ推論とリアルタイム推論を行います。
- 予測をバッチで処理し、データレイクでカタログ化します。マーケティングチームは Amazon Pinpoint にデータをインポートしてキャンペーンを開始できます。

エッジおよび複数のプラットフォームでの Machine Learning

ML モデルのトレーニングには、クラウドで利用できる強力なコンピューティングインフラストラクチャが必要です。ただし、通常、これらのモデルに対する推論に必要な計算能力ははるかに少なくなります。場合によっては、エッジデバイスなど、クラウドへの接続が限られている場合や接続がない場合でも、推論を行う必要があります。鉱業分野はこのタイプのユースケース例です。エッジデバイスがローカルイベントにすばやく応答できるようにするには、推論結果を低レイテンシーで取得できることが重要です。

AWS IoT Greengrass により、エッジデバイスで機械学習を利用できます。AWS IoT Greengrass を使用し、クラウドで作成、トレーニング、最適化されたモデルを使用して、デバイス上での ML 推論をローカルで簡単に実行できます。ML モデルは Amazon SageMaker、AWS Deep Learning AMI、または AWS Deep Learning Containers を使用して構築され、Amazon S3 に保持されて、エッジデバイスにデプロイされます。

図 6 は、AWS IoT Greengrass と AWS クラウドでの ML モデルトレーニング間のインタラクションを示しています。

Archived

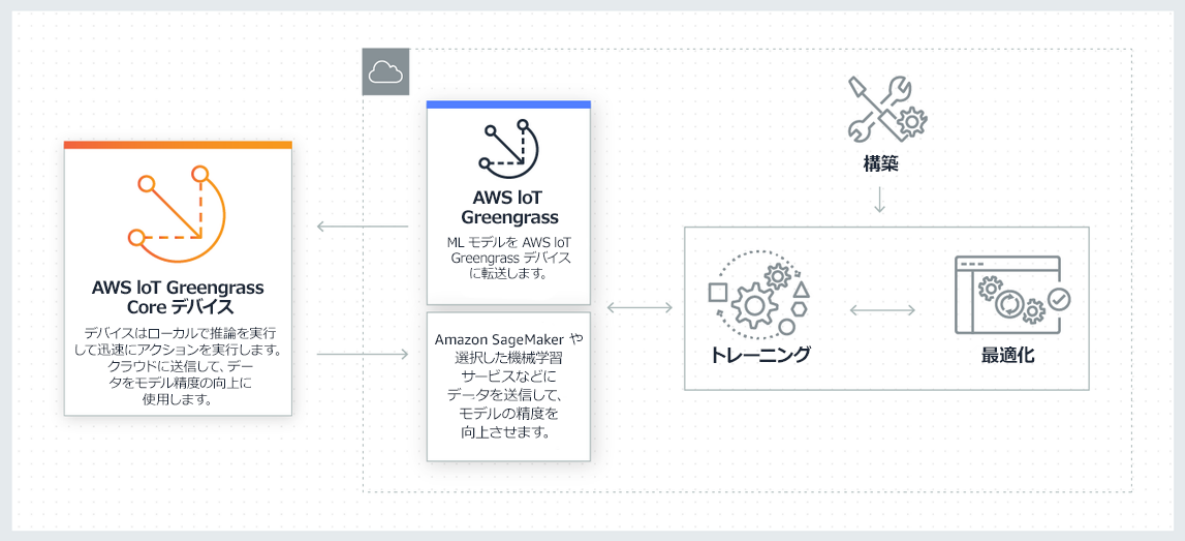

図 *6* - クラウドでの *AWS IoT Greengrass* と *ML* モデル

AWS IoT Greengrass を実行している接続されたデバイスで推論をローカルで実行し、レイテンシーとコストを削減します。すべてのデバイスデータをクラウドに送信して ML 推論と予測を行う代わりに、デバイスで直接推論を実行できます。これらのエッジデバイスで予測が行われるため、結果を取得して分析し、異常値を検出できます。その後、分析されたデータをクラウドで Amazon SageMaker に送り返し、再分類してタグ付けすることで ML モデルを改善できます。

クラウドで構築、トレーニング、最適化された ML モデルを使用し、デバイスでローカルに推論を実行できます。たとえば、Amazon SageMaker でシーン検出分析用の予測モデルを構築し、最適化して任意のカメラで実行して、その後デプロイし、不審なアクティビティを予測してアラートを送信できます。AWS IoT Greengrass で実行している推論から収集されたデータは、Amazon SageMaker に送り返すことができます。Amazon SageMaker では、タグ付けして使用することで、ML モデルの品質を継続的に向上させることができます。

リファレンスアーキテクチャ

エッジで鳥の種類を識別するユースケースのリファレンスアーキテクチャについては図 7 を参照してください。このアーキテクチャでは、オブジェクト検出モデルが Amazon SageMaker でトレーニングされ、エッジデバイスにデプロイされます。カスタムオブジェクト検出は、MRI での腫瘍の発見、異常作物の識別、鉄道ホームの監視など、幅広い業界やユースケースの重要

Archived

な手段となっています。このユースケースに使用されるエッジデバイスは、深層学習対応のビデオカメラである AWS DeepLens です。

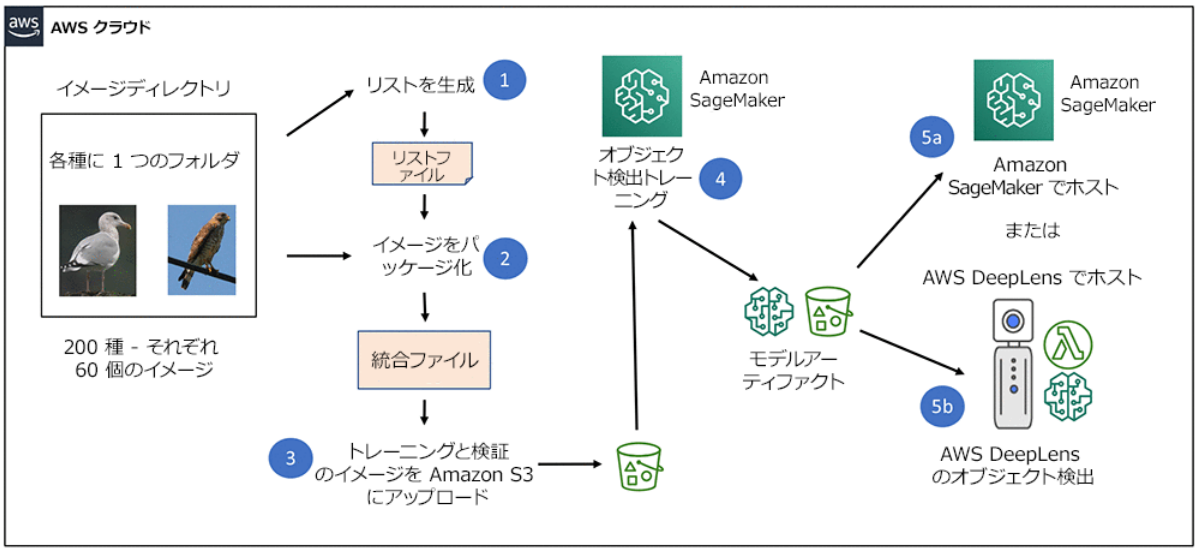

図 7 - エッジアーキテクチャでの鳥の種類の識別

このリファレンスアーキテクチャには、以下の要素が含まれます。

- 鳥の画像データセットを収集、理解、準備する
- Amazon SageMaker の組み込みアルゴリズムを使用してオブジェクト検出モデルをトレーニングする
- Amazon SageMaker エンドポイントを使用してモデルをホストする
- AWS DeepLens のエッジにモデルをデプロイする
    - AWS DeepLens にデプロイする前にモデルアーティファクトを変換する
        - AWS DeepLens で AWS Lambda 関数からモデルを最適化する
- AWS DeepLens でモデル推論と鳥の種類の識別を実行する

AWS DeepLens は、前のアーキテクチャで使用されたエッジデバイスの 1 つです。エッジおよびクラウドで ML モデルを Intel や NVIDIA などの複数のハードウェアプラットフォームにデプロイすることはできますが、ML モデルは MXNet、Tensor、PyTorch など、トレーニングに使

Archive

用するフレームワークと緊密に連携しているため、必ずしも実用的ではありません。ML モデルをトレーニングした特定のプラットフォーム以外のプラットフォームにデプロイする場合は、最初にモデルを最適化する必要があります。ML フレームワークとプラットフォームの数が増えるにつれ、追加のプラットフォーム用にモデルを最適化する労力が増え、著しく時間がかかることがあります。

Amazon SageMaker Neo は、この問題に対処するためのコンパイラとランタイムという 2 つのコンポーネントから構成されます。コンパイラは、モデルを効率的で一般的な形式に変換します。この形式は、従来の汎用フレームワークの 100 分の 1 未満のリソースしか使用しないコンパクトなランタイムでデバイスを実行します。Amazon SageMaker Neo ランタイムは、基盤となるハードウェア向けに最適化されており、ML 推論の高速化に役立つ特定の命令セットを使用します。モデルは、ホームセキュリティ用途のカメラやアクチュエーターのようなリソースに制約のあるデバイスでも実行できるように、メモリのフットプリントが 10 分の 1 未満になるように最適化されます。

モデルのデプロイアプローチ

トレーニングされた ML モデルは、コンシューマーが簡単に呼び出し、そこから予測を取得できるような方法でホストされる必要があります。ML モデルのコンシューマーは、組織の外部または内部のいずれかになります。ML モデルのコンシューマーは通常 ML プロセスを理解しておらず、リアルタイムまたはバッチモードで予測できるシンプルな API のみを必要とします。

Amazon SageMaker は、モデルをデプロイするモデルホスティングサービスと、推論を提供する ML モデルを利用できる HTTPS エンドポイントを提供します。

Amazon SageMaker ホスティングサービスを使用したモデルのデプロイプロセスは、3 つのステップで構成されます。

1. Amazon SageMaker でモデルを作成します。

   ベストプラクティスを適用し、モデルがビジネス要件を満たすことを確認してから次に進みます。

Archived

2. HTTPS エンドポイントのエンドポイント設定を作成します。

   本番環境用バリアントで 1 つ以上のモデルの名前と、Amazon SageMaker が起動し、各本番環境用バリアントをホストする ML コンピューティングインスタンスを指定します。エンドポイント設定では、異なる重みとインスタンス設定 (本番環境用バリアント) を使用して、複数のモデルを同じエンドポイントにアタッチできます。エンドポイントの存続中は、いつでも設定を更新できます。

3. HTTPS エンドポイントを作成します。

   Amazon SageMaker は ML コンピューティングインスタンスを起動し、エンドポイント設定の詳細に指定されたとおりにモデルをデプロイして、HTTPS エンドポイントを提供します。その後、モデルのコンシューマーはエンドポイントを使用して推論できます。

Amazon SageMaker モデルエンドポイント設定の本番環境用バリアント機能により、複数の ML モデルをさまざまなインフラストラクチャでホストし、それぞれが推論リクエストのサブセットまたはすべてを処理できます。本番環境用バリアントを活用して、デプロイのリスクを最小限に抑えることができます。

すべてのバリアントで、モデルエンドポイントのレスポンスにモデルバージョンを含むようにします。モデル推論で問題が発生した場合やモデルの説明可能性が必要な場合、特定のモデルバージョンを知ることでソースへの変更を追跡できます。

標準デプロイ

標準モデルのデプロイでは、Amazon SageMaker エンドポイントは単一の本番環境用バリアントで設定されます。本番環境用バリアント設定によりモデルをホストするインスタンスのタイプと数が指定されます。すべての推論トラフィックは、エンドポイントでホストされている単一のモデルで処理されます。

以下は、標準デプロイの本番環境用バリアント設定例です。

Archived

```
ProductionVariants=[{
  'InstanceType':'ml.m4.xlarge',
  'InitialInstanceCount':1,
  'ModelName':model_name,
  'VariantName':'AllTraffic'
}])
```

ブルー/グリーンデプロイ

ブルー/グリーンデプロイ手法は 2 つの同じ本稼働環境を提供します。この手法は、モデルの新バージョンを本番環境にデプロイする場合に使用します。

図 8 に示すように、この手法には同一の環境が 2 つ必要です。

- **バージョン n** を実行するライブ本番環境 (ブルー)
- **バージョン n+1** を実行するこの環境の完全なコピー (グリーン)

ブルー環境 (バージョン n) がライブトラフィックを処理する一方で、合成トラフィックを使って次のリリース (バージョン n+1) をグリーン環境でテストします。このテストでは、新しいモデルが技術メトリクスとビジネスメトリクスの両方を満たすことを検証する必要があります。グリーン環境で n+1 バージョンのテストがすべて成功すると、ライブトラフィックがグリーン環境に切り替わります。次に再度グリーン環境でメトリクスを検証しますが、今回はライブトラフィックを使用します。このテストで問題が見つかった場合は、トラフィックをブルー環境に戻します。一定期間にわたり問題が発生しない場合は、ブルー環境を削除します。

Archived

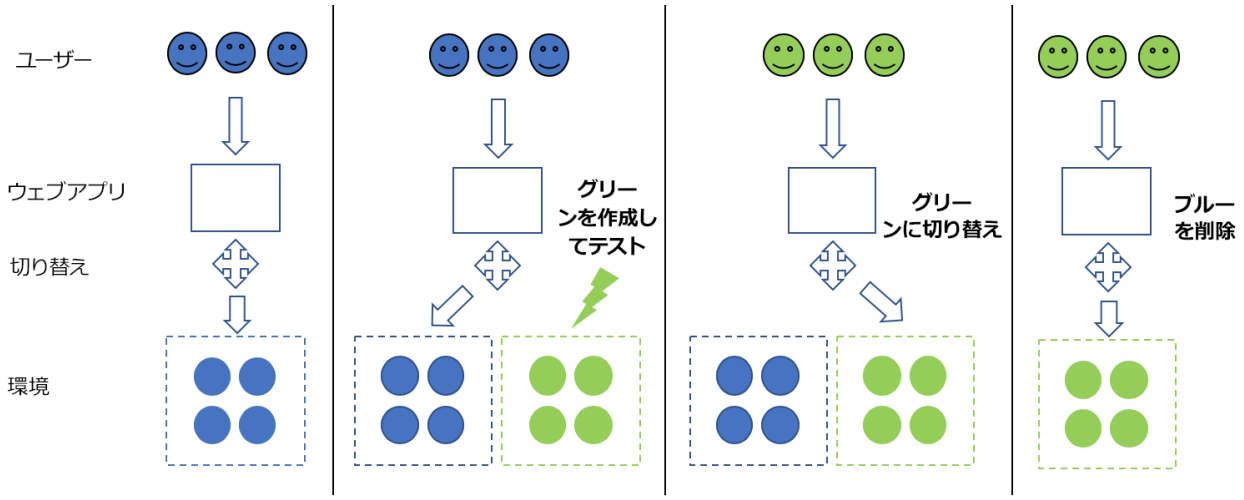

図 *8* - ブルーグリーンデプロイ手法

Amazon SageMaker でブルーグリーンデプロイを行うには、以下のような手順があります。

1. 既存のライブモデルと新しいモデルに同一の本番環境用バリアントを使用して、新しいエンドポイント設定を作成する。
2. 既存のライブエンドポイントを新しいエンドポイント設定で更新する。

   Amazon SageMaker は、新しい本番環境用バリアントに必要なインフラストラクチャを作成し、ダウンタイムなしで重みを更新する。
3. API コールを使用して、トラフィックを新しいモデルに切り替える。
4. 新しい本番環境用バリアントのみを使用して新しいエンドポイント設定を作成し、エンドポイントに適用する。

   Amazon SageMaker は、過去の本番用バリアントのインフラストラクチャを終了する。

Archived

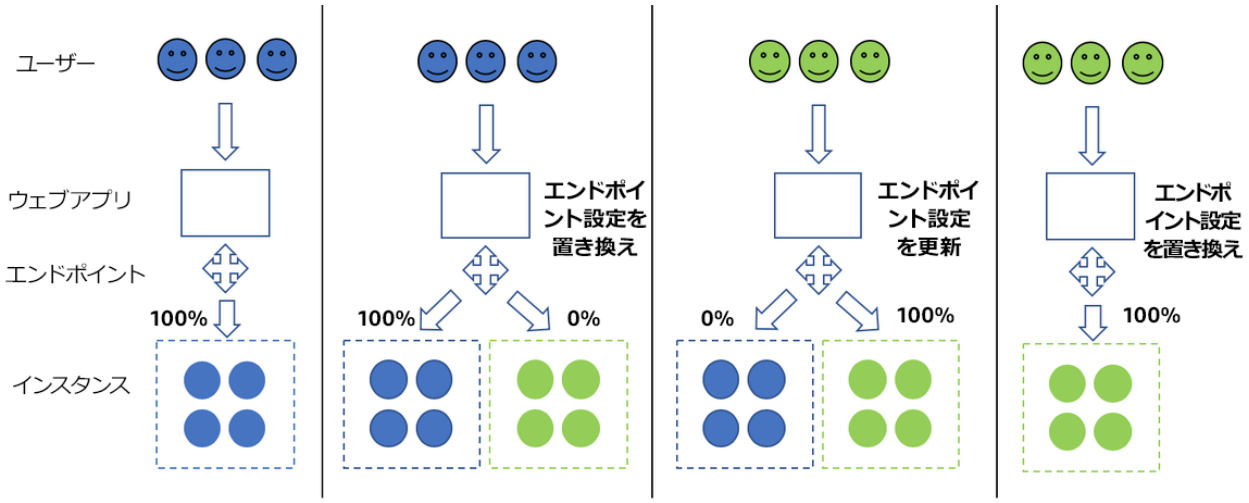

図 9 - *Amazon SageMaker* 本番環境用バリアントを使用したブルー/グリーンモデルのデプロイ

カナリアデプロイ

カナリアデプロイでは、まず少人数のユーザーへのデプロイから始めることで、新しいリリースを検証するリスクを最小限に抑えることができます。新しいリリースのテストが満足できる結果になるまで、他のユーザーは、旧バージョンを使用します。その後で、新しいリリースを全ユーザーに段階的に展開していきます。

Archived

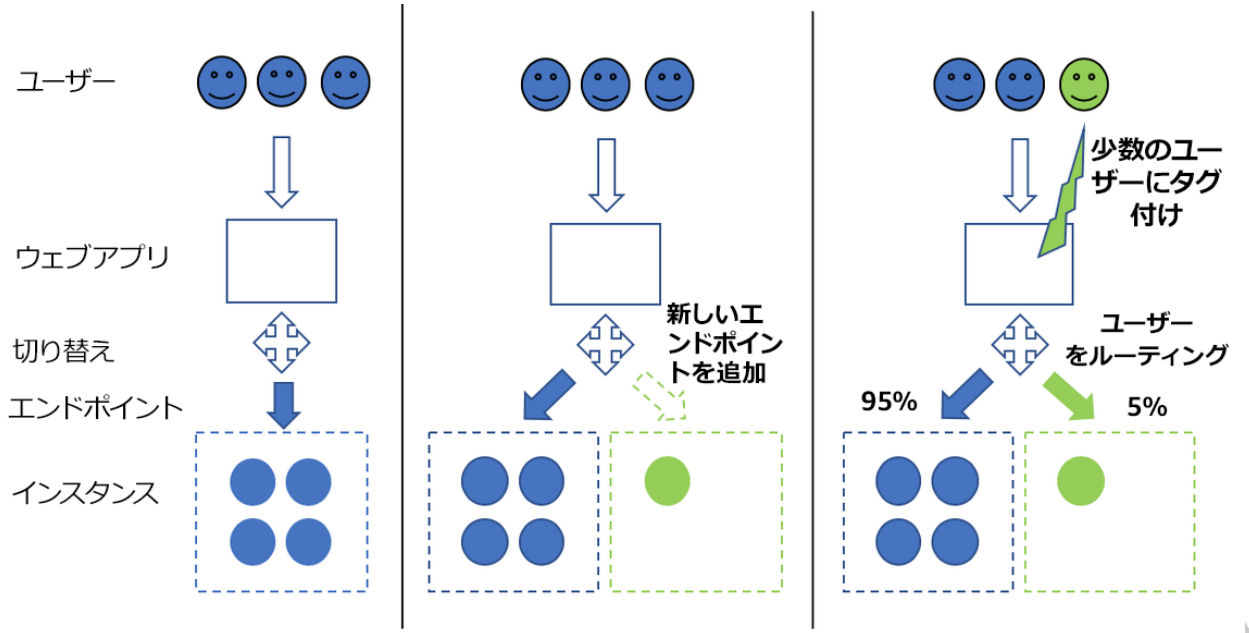

図 *10 - Amazon SageMaker* の本番環境用バリアントを使用したカナリアデプロイ: 初期ロールアウト

新しいモデルが期待どおりに実行できることを確認したら、エンドポイントを必要に応じて増減しながら、段階的にすべてのユーザーに展開します。

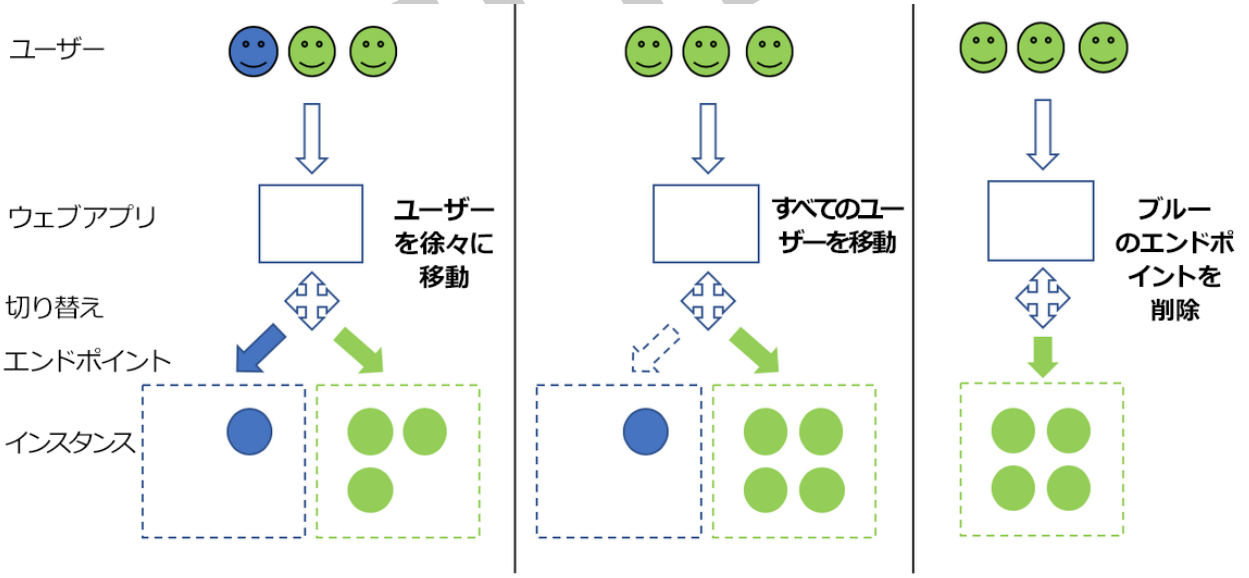

図 *11 - Amazon SageMaker* の本番環境用バリアントを使用したカナリアデプロイ: 完全ロールアウト

A/B テスト

A/B テストは、クリックスルー率やコンバージョン率などの大まかなメトリクスをモニタリングしながら、同じ機能のパフォーマンスをバージョン間で比較する手法です。このコンテキストでは、さまざまなユーザーに対してさまざまなモデルを使用して推論を実施し、結果を分析するということを意味します。このさまざまなモデルとは、同じアルゴリズム (ビルトインの Amazon SageMaker アルゴリズムまたはカスタムアルゴリズム) で構築されますが、別々のハイパーパラメータ設定を使用します。

A/B テストはカナリアテストに似ていますが、A/B テストのほうがユーザーグループは大きく、期間も長くなります (通常は数日数週間)。このタイプのテストでは、Amazon SageMaker エンドポイント設定に本番用バリアントを 2 種類 (モデル A 用とモデル B 用) 使用します。まず、両モデルのトラフィックが均等 (50/50) になるように設定し、同一のインスタンスを使用するように設定されていることを確認します。同じ重みの初期設定で両モデルのパフォーマンスをモニタリングした後は、トラフィックの重みを段階的に変更してモデルのバランスを変える (60/40、80/20 など) か、ウェイトをシングルステップで変更して、1 つのモデルがすべてのライブトラフィックを処理するようにします。

A/B テスト用本番環境バリアント設定のサンプルは以下のとおりです。

```
ProductionVariants=[
  {
    'InstanceType':'ml.m4.xlarge',
    'InitialInstanceCount':1,
    'ModelName':'model_name_a',
    'VariantName':'Model-A',
    'InitialVariantWeight':1
  },
  {
    'InstanceType':'ml.m4.xlarge',
    'InitialInstanceCount':1,
    'ModelName':'model_name_b',
    'VariantName':'Model-B',
    'InitialVariantWeight':1
  }
])
```

Archived

Well-Architected フレームワークの柱

優れた機械学習のワークロードソリューションを設計するには、次のポイントが重要です。それぞれの柱では、ML ワークロードの定義、ベストプラクティス、質問、検討事項、主要な AWS のサービスなど、機械学習レンズの詳細についてご説明します。

ML ワークロードを設計するときは、「AWS Well-Architected フレームワーク」のホワイトペーパーに記載された最適なベストプラクティスや質問を参考にすることをお勧めします。

運用上の優秀性の柱

「運用上の優秀性」の柱には、システムの運用およびモニタリング、インサイトの獲得を通してビジネス価値を生み出し、サポートプロセスと手順を継続的に改善する能力が含まれます。

設計の原則

クラウドには、ML ワークロードの運用面を最適化する能力を高めるための原則が数多くあります。このワークロードをうまく運用する能力は、ML ワークロードを迅速に実用化するために不可欠です。

AWS の「運用上の優秀性」のベストプラクティスは、ML ワークロードがクラウドで効率的に稼働するように設計されています。すべての AWS ワークロードに適用する運用上の優秀性に関する標準のプラクティスは、「運用上の優秀性の柱: AWS Well-Architected フレームワーク」のホワイトペーパーを参照してください。ML ワークロードの「運用上の優秀性」を最適化する設計の原則には、次のようなものがあります。

- **組織横断的なチームを設立する**: ワークロードを本稼働環境に移行できるよう、ドメインの専門知識がある組織横断的なプロジェクトチームをつくります。ML ワークロードの開発、デプロイ、サポートに関わるすべての関係者を集めてください。

Archived

- **エンドツーエンドのアーキテクチャと運用モデルを早期に特定する:** ML 開発ライフサイクルの初期段階で、モデルのトレーニングとホスティングを行うエンドツーエンドのアーキテクチャと運用モデルを特定します。これにより、ML ワークロードの開発、デプロイ、管理、統合に必要なアーキテクチャと運用上の検討事項を早期に特定できます。
- **ML ワークロードのモニタリングと測定を継続的に行う:** モデルで実施したトレーニング、ホスティング、予測に関する主なメトリクスを特定し、定期的に収集します。これにより、システムのメトリクス、モデルレイテンシー、データドリフトの検出などの重要な評価基準に基づいて、デプロイされたモデルの健全性を継続的に監視できます。
- **モデルの再トレーニング戦略を策定する:** デプロイされたモデルのパフォーマンスと有効性は、時間が経つと変わることがあります。モデルバージョンのパフォーマンスと有効性がビジネス目標を達成する条件を示すメトリクスを特定し、しきい値に応じて、モデルの再トレーニングが必要なことを表示して、それらのアクティビティをトリガーするアラートを作成します。アラートを使うと、現モデルの無効化、旧モデルバージョンへの復帰、新しい Ground Truth データに基づく新モデルの再トレーニング、データサイエンスチームによるモデルの再トレーニング戦略の調整などのアクティビティをトリガーできます。
- **機械学習の検出アクティビティと結果を文書にまとめる:** データサイエンスによる検出と調査のタスクから、機械学習モデルの生成と進化に関する背景情報とインサイトを得ることができます。ソースコントロールで管理およびバージョニングされたコードパッケージに、これらのアクティビティを文書化します。
- **機械学習のインプットデータとアーティファクトをバージョン管理する:** インプットデータとアーティファクトをバージョン管理すると、ML ワークロードの旧バージョンのアーティファクトを再現できます。モデルアーティファクトに加え、モデル作成に使用されたトレーニングデータやトレーニングソースコードなどのインプットをバージョン管理します。他にも、使用されたアルゴリズム、機能エンジニアリングのソースコード、ホスティング設定、推論コードとデータ、後処理ソースコードも対象です。

Archived

- **機械学習デプロイパイプラインを自動化する:** 開発環境から本番環境へのモデル移行を定義するパイプラインを使用して、ML デプロイパイプラインにおける人間のタッチポイントを最小化し、ML モデルがコンスタントにデプロイを繰り返すように設定します。ユースケースの要件を達成してビジネスの課題を解決できるデプロイ戦略を策定し、実行します。必要に応じて、手動の品質ゲートをパイプラインに追加し、モデルを対象環境にデプロイする準備が完了しているかどうかを評価します。

定義

クラウドにおける「運用上の優秀性」には 3 つのベストプラクティスがあります。

- 準備
- 運用
- 進化

ベストプラクティス

準備

運用上の優秀性を準備するには、ワークロードとそれに期待される動作を理解する必要があります。ML ワークロードの運用準備のためには、次のような視点で評価する必要があります。

- 運用の優先順位
- 運用に適した設計
- 運用開始の準備

> MLOPS 01: 機械学習ワークロードの運用とサポートのために、チームでどのような準備をしましたか?

Archived

ML のワークロードは、サポートの観点とは異なることがよくあります。これは、ML モデルを統合しデプロイする必要があるチームが ML ワークロード固有の運用面についてよく理解していない場合があるためです。ML モデルを本番環境に効果的に統合し、ビジネス目標を達成するためのベストプラクティスとは、チーム間で相互に協力すること、そして、基本的な習熟度レベルで、機械学習ワークロードのサポートと保守を行うすべてのリソースをトレーニングすることです。

ML ワークロードで考慮する必要がある運用上の要件には、拡張機能やレイテンシーのモデル化機能など、データサイエンティストが対応できないものがよくあります。それとは対照的に、運用担当者では測定値を評価できない、キャプチャが必要な特定のモデルの振る舞いもあります (たとえば、時間経過によるモデルの有効性など)。

ML ワークロードを統合および運用するためのチームアプローチを検討するうえで、重要なプラクティスは次のとおりです。

- モデルと API を開発するチームと、ML ワークロードをサポートするチームまたは監査責任のあるチームで、お互いに概要のトレーニングを実施する。
- モデルと API を本番ソリューションに効果的に統合できるように、組織横断的なチームをつくる。これにより、ML ワークロードを本番ソリューションへ統合およびデプロイするのを阻害する一般的な要因を回避できます。

MLOPS 02: モデル作成アクティビィをどのようにドキュメント化していますか?

ML モデル開発のライフサイクルは、モデルのバージョンを完成する前に必要になる実験量などの点で、アプリケーション開発ライフサイクルとは大きく異なります。モデルのバージョンをサポートし、適切に活用できるようにするため、モデル生成プロセスを文書化します。具体的には、設定仮説、モデルに必要なデータの事前/事後処理、およびモデルバージョンをシステムやアプリケーションに統合する際に必要な処理です。

このプロセスの文書化によって、モデルの統合とサポートを担当する関係者がモデルに関連する情報を共有できるようになります。このドキュメントをソースコントロールリポジトリのよ

Archived

うな安全でバージョン管理された場所に保存することで、モデルの作成と変更に関連する知的財産を蓄積できます。

AWS では、Amazon SageMaker Notebooks と Amazon SageMaker Studio がマネージド型のノートブック環境を提供しており、データサイエンティストはそこで開発プロセスや実験を文書化できます。これらのノートブックはソースコントロールシステムと統合され、デプロイされるモデル用に作成するドキュメントの標準的な部分になります。

MLOPS 03: モデルの経緯をどのように追跡していますか?

アルゴリズムごとに、異なるアルゴリズムとハイパーパラメータで ML モデルを繰り返し開発すると、モデルのトレーニング実験やモデルバージョンの数が増加します。これらのモデルのトラッキングとモデルの系譜の追跡は、監査やコンプライアンスのためだけでなく、モデルのパフォーマンスが劣化した根本原因を分析するためにも重要です。

さらに、モデルの系譜とデータの系譜を同期させることが重要です。これはデータ処理コードのバージョンとモデルバージョンの両方を生成しながら、各モデルバージョンのトレーニング用にデータのパイプラインをすべて文書化し、モデルのエラーのデバッグやコンプライアンスを監査する必要があるためです。

AWS では、Amazon SageMaker Experiments を使用して ML モデルのイテレーションを整理してトラッキングできます。Amazon SageMaker Experiments は、モデルの入力パラメータ、設定値、出力アーティファクトを自動キャプチャし、実験として保存します。これにより、モデル開発の各イテレーションで作成および利用される多数のバージョンの入力および出力アーティファクトを管理する目的で、手動トラッキングやカスタマイズのトラッキングソリューションを構築する必要はなくなります。このアプローチにより、チームは多くの実験から最適な結果を得られるモデルを簡単に選択してデプロイできます。

MLOPS 04: ML ワークロードの開発とデプロイのパイプラインをどのように自動化していますか?

Archived

そのデザインの一部として、ML ワークロードのデプロイ、更新、更新、運用の方法を定義した運用アーキテクチャを作成します。Infrastructure-as-Code (IaC) と Configuration-as-Code (CaC) に共通するプラクティスを組み込むことで、デプロイに一貫性を持たせ、高い信頼性で複数の環境にわたりリソースを正確に再作成できるようになります。さらに、さまざまなフェーズやターゲット環境で ML のワークロードの動きをコントロールされた方法でオーケストレーションする自動化されたメカニズムを用意することで、ワークロードを更新する際のリスクを軽減できます。

ML ワークロード (MLOps) に、継続的インテグレーションと継続的デリバリー (CI/CD) のプラクティスを組み込み、品質ゲートとトレーサビリティなどの自動化を実現します。たとえば、CI/CD パイプラインはソースとアーティファクトのバージョン管理から始まりますが、これは標準的な変更管理活動をサポートすると同時にデバッグ活動の信頼性を高めます。ソース、データ、アーティファクトのバージョン管理のプラクティスを ML ワークロードに適用すると、デプロイ時のバージョンまで遡るトレーサビリティにより運用デバッグアクティビティを改善できます。さらに、バージョン管理は、変更に失敗した場合や、新モデルが必要な機能を実現できなかった場合に、ある既知の作業バージョンまでロールバックできます。

CI/CD パイプライン全体にロギングとモニタリングを実装すると、より高いレベルの環境へのデプロイを許可または拒否する品質ゲートを追加する基盤を構築できます。ベストプラクティスには、パッケージのコンテナの脆弱性チェックなどの標準的な品質ゲートや、パイプラインに含まれる ML 特有の品質ゲートなどがあります。これらの品質ゲートでは、お客様のビジネスユースケースで特定された固有のメトリクスを使用してモデルを評価する必要があります。これらには、適合率と再現率、F1、正解率を評価するメトリクスが含まれる場合があります。品質ゲートを追加すると、セキュリティリスク、モデルパフォーマンスや精度メトリクスの低下などの、運用上の課題を示す状態が確認された場合に、現在デプロイされているモデルが新バージョンのモデルに置き換わらないようにすることができます。

AWS では、API エンドポイントを介して Amazon Polly などの AI サービスを提供しています。この結果、この分野には固有のベストプラクティスはありません。モデルのトレーニングおよびデプロイがすべて完了しているためです。API エンドポイントと連動するコードとシステムに関する開発およびデプロイの自動化は、標準的な AWS のベストプラクティスに従う必要

Archived

があります。Amazon Personalize などの AWS の AI サービスの中には、お客様が提供するトレーニングデータに基づいてモデルをトレーニングするものがあります。これらのサービスでモデルの作成および更新を行う際は、データを安全に保護するために本ホワイトペーパーに記載されているベストプラクティスに従ってください。

AWS で独自の ML モデルを構築してトレーニングする場合、開発とデプロイのパイプラインは、AWS のサービスとサードパーティーの統合を組み合わせて自動化されます。モデルデプロイの自動パイプライン作成に使用する適切なサービスやツールを決定するには、デプロイ戦略、モデル特性、モデルのトレーニング戦略を特定することが重要です。

ML ワークロードは、使用する AWS の ML サービスによってさまざまです。ただし、パイプライン作成の一般的なガイドラインでは、AWS CodePipeline などのオーケストレーションレイヤーと、パイプラインで各ステージを実行するロジックを組み合わせて使用することが含まれます。関数ベースのロジックを作成して実行するには、サーバー管理の必要がなく運用オーバーヘッドが小さい AWS Lambda を使用します。次の図は、AWS におけるデプロイのパイプライン事例です。ただし、このデプロイは、前述した要因によって変わります。

Archived

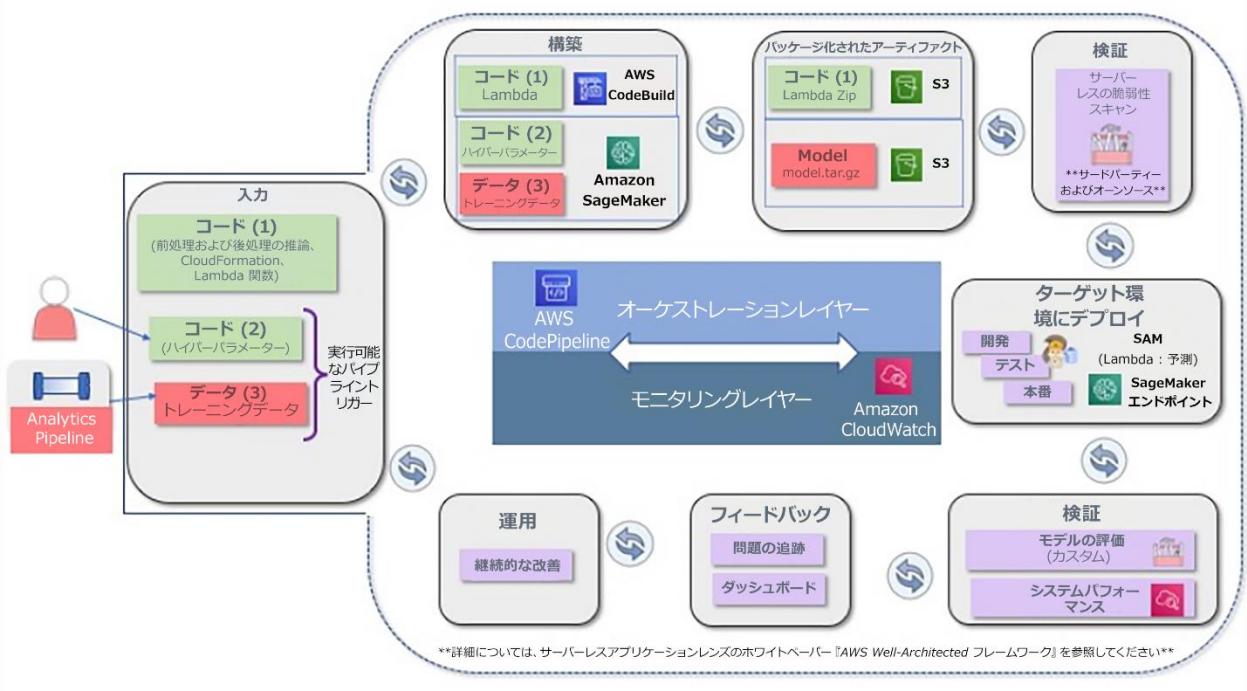

図 *12 - AWS* の機械学習用 *MLOps CI/CD* パイプライン事例

運用

> MLOPS 05: モデルがホストするアクティビティのモニタリングとロギングはどのように行いますか?

モデルエンドポイントをホストして予測を行う場合、そのエンドポイントにモニタリングとアラートを設定し、潜在的な問題や改善の機会を特定して対応する必要があります。モデルエンドポイントには、基盤となるコンピューティングリソースの運用状況を測定するメトリクスのモニタリングとアラートにくわえ、エンドポイントのホストの運用状況のモニタリングも設定するべきです。

AWS では、エンドポイントのコンピューティングリソースの運用状況を管理する標準プラクティスとして、「AWS Well Architected: 運用上の優秀性」のホワイトペーパーで定義されているものが必要です。Amazon SageMaker は、コアシステムのメトリクスを自動でモニタリングし、またホストされたモデルの自動スケーリング機能を設定する機能によって、エンドポイントをオンデマンドでサポートするコンピューティング基盤を動的に調整します。この機能によって、エンドポイントは運用オーバーヘッドを削減しつつ動的に需要に対応できます。

自動スケーリングをサポートするコンピューティングリソースのモニタリングのほか、Amazon SageMaker にはエンドポイントの使用状況や運用状況をモニタリングするためのエンドポイントメトリクスを出力する機能があります。Amazon SageMaker Model Monitor は、本番稼働中の ML モデルをモニタリングする機能で、データの品質問題が発生した場合にアラートを出します。ベストプラクティスとしては、サービスを使用してモデル予測エンドポイントメトリクスの集約と分析を行う仕組みを作ることが推奨されます。サービスには、ビルトイン機能の Kibana でダッシュボードやビジュアリゼーションが使える Amazon Elasticsearch などがあります。さらに、ホストメトリクスをバージョン管理されたインプットにさかのぼることができるトレーサビリティ機能によって、現在の運用パフォーマンスに影響する可能性がある変更を分析できます。

Archived

進化

> MLOPS 06: 新しいデータや更新データで ML モデルを再トレーニングするタイミングはどのように判断しますか?

ML ワークロードは、初期段階では価値が高い予測値を提供しますが、その同じモデルによる予測の精度は時間とともに低下します。これは「ドリフト」という現象によることが多く、時間の経過による正解データの変化など、さまざまな要因が影響した結果である可能性があります。モデルの予測はビジネスの意思決定につながるため、これは既存モデルのパフォーマンスに間接的に影響する可能性があります。特定の配送物に関するリスクを予測する小売業のシナリオを例に挙げます。トレーニングデータには、過去に破損があった配送物のデータが含まれています。この企業がビジネスの意思決定にこのモデルの使用を開始すると、破損があった製品のインスタンスが減少するため、データに間接的に影響を与えます。

存在する最新のデータを使ってモデルが効果的に学習と予測ができるように、新しいデータや更新されたデータによるモデルの再トレーニングが必要となることがよくあります。追加データを効果的に ML モデルに組み込むには、既存モデルのパフォーマンスと定義されたメトリクスの比較分析を行い、モデル分散があるしきい値に達したときにアラームや再トレーニングイベントをトリガーしたり、新しい既知データに基づいて時間の経過とともにモデルをプロアクティブに再トレーニングしたりするメカニズムを実装する必要があります。

追加データを検討するためのベストプラクティスは、次のとおりです。

- モデルのパフォーマンスと精度を示すメトリクスを定義します。
- それらのメトリクスを定期的に取得するメカニズムを構築して分析を行い、メトリクスのしきい値に基づいたアラートを出します。たとえば、あるモデル予測までさかのぼってダウンストリームの結果の確認、データ取得およびトラッキングを行い、エラー率などのメトリクスを経時的に計算するシステムが必要になることがあります。

Archived

- モデルの再トレーニングが適切かどうかを評価します。追加の正解データが利用可能または取得可能かどうか、追加データをラベル付けする必要があるのかどうかを判断します。新しいデータを使用してトレーニングを定期実行する、新しいデータを再トレーニングのトリガーとする、メトリクスのしきい値を使用して再トレーニングを評価するなど、既知のワークロードの特性に基づく再トレーニングの初期戦略を立てます。この戦略では、変化量、再トレーニングコストおよび本番モデル更新の潜在的な価値のトレードオフを評価する必要があります。定義された戦略に基づいて、再トレーニングの自動化を設定します。

AWS では、Amazon Translate などの AI サービスは新しいデータで自動的にトレーニングされるので、AWS によって更新されたモデルを活用して、経時的にモデルのパフォーマンスを改善できます。

AWS で ML サービスを使用して独自モデルの構築およびトレーニングを行うとき、AWS はさまざまな機能を通して、新しいデータによるモデルの継続的な再トレーニングをサポートします。準備したトレーニング用データは Amazon S3 に保存してください。再トレーニングのシナリオは以下のとおりですが、これはワークロードの特性に基づいて検討する必要があります。

- **モデルドリフト (メトリクス主導型再トレーニング):** データの分布が元の学習データから大きく乖離したり、サンプルにないデータが増加したりするような、バリエーションの影響を受けやすい ML ワークロードでは、定義されたメトリクスや新たに用意されたデータに基づいてモデルの再トレーニングをトリガーする自動メカニズムを設定します。AWS では、データのドリフトを把握するメカニズムの 1 つとして、データ分散の変化を検出する Amazon SageMaker Model Monitor があります。ドリフトの検出は、AWS CloudWatch のメトリクスを利用すると可能になります。CloudWatch により再トレーニングのジョブは自動でトリガーされます。

Archived

- **追加のトレーニングデータ:** AWS には、Amazon S3 バケットへの新しいデータの PUT に基づいて再トレーニングを自動でトリガーするメカニズムがあります。コントロールされたモデルの再トレーニング開始方法として好ましいのは、ソースとなる Amazon S3 バケットの変更に基づくイベントトリガーを使用して ML パイプラインを設定することです。S3 バケットに新しいトレーニングデータがあることを検出するには、CloudTrail と CloudWatch Events を組み合わせることで、AWS Lambda 関数や AWS Step Functions のワークフローをトリガーして、トレーニングパイプラインの再トレーニングタスクを開始できます。次の図は、AWS CodePipeline と ML サービスを表している例です。

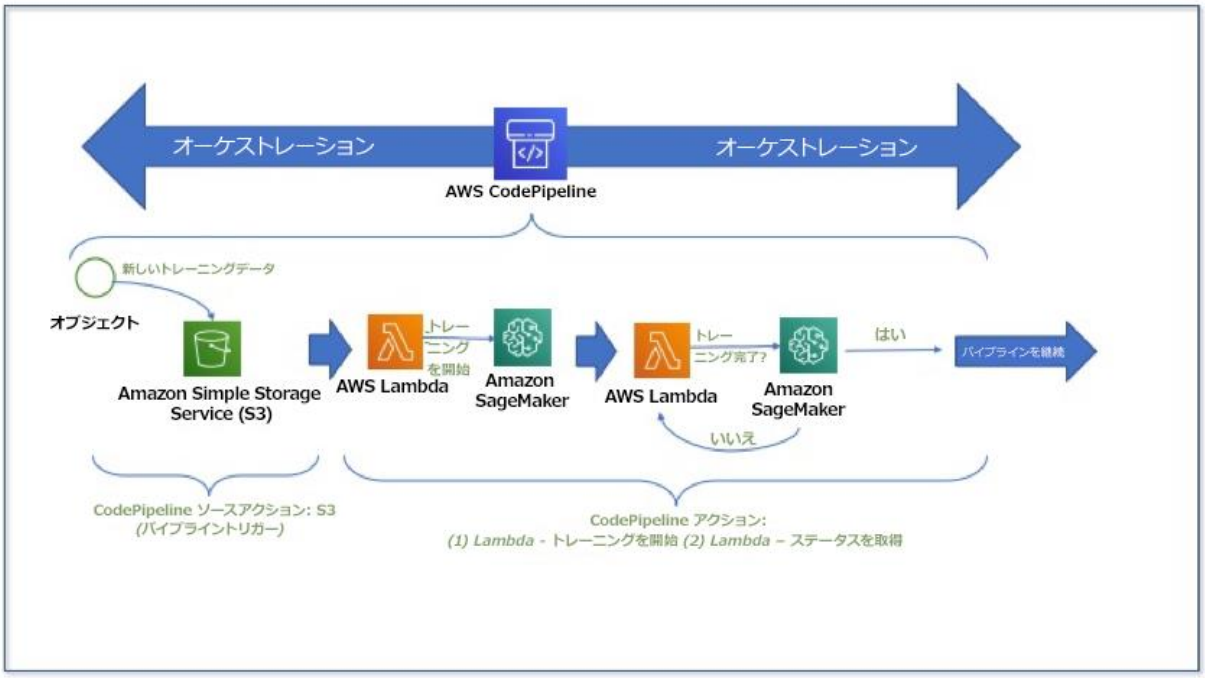

図 *13 - ML* サービスの新しいトレーニングデータをイベントトリガーとする例

あるいは、AWS のサービスの API と統合された、デプロイとオーケストレーションを行う Jenkins などのサードパーティー製ツールを使用して、新しいデータが利用可能になった際のモデルの再トレーニングを自動化することもできます。

モデルに追加データを組み込む方法を定義するときは、モデルをバージョン管理してトレーニングのすべての過去データを元の形式で維持するか、またはトレーニングデータの過去バージョンを簡単に再現できるようにしてください。これにより、モデルのアーティファクトが誤って削除された場合でも、バージョン管理されたアーティファクトの作成に使用されたすべてのコンポーネントの統合バージョンを使用して、同じモデルアーティファクトを再現できます。

> MLOPS 07: モデル開発、モデルトレーニング、モデルホスティングのイテレーション間の学習はどのように組み込めばよいですか?

学習を取り込むには、成功した開発実験、失敗の分析、運用アクティビティを共有し、説明できるような継続的なフィードバックの仕組みを持つことが重要です。これによって、それ以降に実施する ML ワークロードのイテレーションを改善し続けることができます。

学習についての重要な検討事項は、次の側面によるモデル評価です。

- **ビジネス評価:** ビジネス目標に対するモデルの成功を検証するためには、ベースラインとなるビジネスメトリクスを設定するだけでなく、その情報を継続的に収集し、長期間にわたってモニタリングするメカニズムが必要です。たとえば、もしビジネス目標が特定の顧客を広告キャンペーンのターゲットにして商品の売上を伸ばすことである場合、商品の売上、ターゲット顧客、商品を購入した顧客など、成功の KPI (Key Performance Indicator) を継続的に測定するためのベースラインを設定し、運用の仕組みを作る必要があります。
- **モデル評価:** ML 問題と捉える事象に対してモデルの成功を検証するためには、エンドツーエンドのパイプラインでモデルのパフォーマンスに関連する主要なメトリクスを取得する必要があります。これには、トレーニングや検証エラーなどのトレーニングメトリクスと、予測精度など継続的に発生するホストモデルのメトリクスがあります。個々のメトリクスは、ユースケースとビジネス KPI に基づいて決定される必要があります。

Archived

- **システム評価:** ML ワークロードのフェーズのサポートに使用するシステムレベルのリソースを検証するためには、コンピューティング、メモリ、ネットワークなどのシステムレベルのリソースを継続的に収集し、モニタリングすることが重要です。ML ワークロードの要件は、フェーズごとに変化します。たとえば、トレーニングジョブではメモリ負荷が高く、推論ジョブではコンピューティング負荷が高くなります。

  AWS では、この分野の一般的なプラクティスに加えて、SageMaker のノートブックインスタンスを活用して、モデル開発ライフサイクルのドキュメントと包括的な説明を作成するデータサイエンスの調査活動を把握することもできます。これは、本番モデルのサポートを成功させるだけでなく、モデルの進化に伴い、複数のデータサイエンティストや開発者の行動の可視化、トレーサビリティを実現するうえでも重要となります。さらに、収集した主要な運用メトリクスを一元的に可視化することで、チームで継続的にレビューを行い、経時的に運用の遡及分析を行うことができます。

リソース

運用の卓越性のベストプラクティスの詳細については、以下のリソースを参照してください。

ドキュメントとブログ

- Amazon SageMaker と Apache Airflow でエンドツーエンドの機械学習ワークフローを構築する
- AWS Step Functions を使用した Amazon SageMaker モデルの自動的および継続的なデプロイ
- Step Functions で Amazon SageMaker を管理する
- AWS CodePipeline と AWS Lambda を使用したパイプラインの作成

ホワイトペーパー

- 運用上の優秀性の柱 – AWS Well-Architected フレームワーク
- サーバーレスアプリケーションレンズ – AWS Well-Architected フレームワーク

Archived

セキュリティの柱

セキュリティの柱とは、リスクの評価とその軽減戦略を通してビジネス価値を提供しながら、情報、システム、資産を保護する能力のことを指します。

設計の原則

Well-Architected のセキュリティ設計原則全体に加えて、ML セキュリティの具体的な設計の原則があります。

- **ML システムへのアクセスを制限する:** システムを設計するときは、ML システムへのアクセスレベルを考慮する必要があります。ML モデルとそのモデルのトレーニングに使用するデータセット双方へのアクセスを制限して、データやモデルの汚染を避けるべきです。推論エンドポイントは、権限のある関係者のみが ML モデルに対して推論を行えるように保護する必要があります.
- **データガバナンスの確保:** ML に使用するデータは複数のソースから収集することがあり、組織のさまざまなチームで利用可能である必要があります。本番データはデータサイエンスの開発活動だけでなく、トレーニングモデルにも必要です。高品質なデータセットにチームがエンドツーエンドでアクセス可能にするには、データセットの完全性、セキュリティ、可用性を保証するデータガバナンス戦略が必要です。ガバナンスとアクセスコントロールの機能があるデータレイクソリューションを導入すると、開発者とデータサイエンティストには、調査活動やトレーニングモデルに使用する高品質データへのコントロールされたアクセスが付与されます。また、データは流出やミューテーションから保護する必要があります。組織の異なるチームがデータに対して実行できるアクションと、データの送信先を管理します。

Archived

- **データ系統を適用する**: ML プロセスではフェーズごとにさまざまなソースからのデータを使用するため、データの原型や経時的な変遷をモニタリング、トラッキングします。データ系統によって可視化が可能となり、データ処理や機械学習のエラーを根本原因にさかのぼるプロセスを簡略化できます。このデータにアクセス可能なユーザーとそのデータの使用方法は、厳密に管理します。存続中、データがどのように管理されてきたかを実証するには、予防的な制御、監査、モニタリングが必要です。
- **規制コンプライアンスの適用:** ML システムに関する規制上の課題には、HIPAA や GDPR の説明にあるようなプライバシーへの配慮があります。ML システムはそれらのフレームワークで確立された規制ガイドラインを遵守する必要があります。また、連邦準備制度理事会の SR 11-7 ガイドラインのような財務リスク管理上の課題が該当することもあります。従来のようなアルゴリズムが静的であるモデルと異なり、ML/AI アルゴリズムを活用したモデルは時間とともに変化する可能性があるため、常に注意を払って確実に規制機関に準拠する必要があります。

定義

クラウド内でのセキュリティには、5 つのベストプラクティス領域があります。

- Identity and access management
- 発見的統制
- インフラストラクチャの保護
- データ保護
- インシデント対応

ベストプラクティス

Identity and Access Management

MLSEC 01: サーバーレス API へのアクセスをどのようにコントロールしますか?

Archived

一般的に、ML ワークロードの構築には複数のチームが関与し、各チームは ML フェーズを 1 つ以上担当します。 ML プロセスの各フェーズで使うリソースには、データ、アルゴリズム、ハイパーパラメーター、トレーニング済みモデルのアーティファクト、インフラストラクチャなどがあります。すべてのリソースへのアクセスは、最小権限のアクセスで厳密に制御する必要があります。 たとえば、特徴量エンジニアリングの担当チームが、モデルのトレーニングやデプロイを実施しない場合は、それを実施する権限を持つべきではありません。 同様に、本番環境へのモデルデプロイを担当する運用チームは、トレーニングデータへのアクセス権限や変更権限を持つべきではありません。 ワークロードによっては、チームメンバーが ML ワークロードの複数のフェーズにわたって作業を担当し、その役割の責任を果たすために適切なアクセス許可が必要になるケースもあります。

AWS では、さまざまなリソースやサービスへのアクセス権を AWS IAM で制御します。認証には ID を使用しますが、データへのアクセス、データおよびアルゴリズムの変更、トレーニングジョブの起動、モデルのデプロイを、誰 (人間) と何 (プロセス) が実行できるかといった詳細な制御は、IAM のユーザー、グループ、ロール、ポリシーを使用して実装されます。

デプロイ済みモデルへのアクセスは、意図した正当な消費者のみに制限します。AWS 環境の中に存在する、またはその環境にアクセスする一時的な IAM 資格情報を取得できるモデルの利用者には、デプロイ済みモデルのエンドポイントを起動に必要な、最小権限のアクセス許可が付与された IAM ロールを使用してください。 環境外のコンシューマーには、API ゲートウェイとホストモデルのエンドポイントの組み合わせを使用し、安全な API を介してアクセスが提供されます。

発見的統制

ML に適用される発見的統制分野のセキュリティについてのベストプラクティスは、「AWS Well-Architected フレームワーク」のホワイトペーパーを参照してください。

インフラストラクチャの保護

ML に適用されるインフラストラクチャの保護分野のセキュリティについてのベストプラクティスは、「AWS Well-Architected フレームワーク」のホワイトペーパーを参照してください。

Archived

データ保護

> MLSEC 02: ML ワークロードで使用される機密データへのアクセスをどのように保護し、監視していますか?

ML プロセスにおいて、データはあらゆる段階で使用されます。プロジェクトの初期段階でビジネス目標の特定後、さまざまなデータソースのアクセス性と可用性を評価し、利用可能なデータとやりとりします。プロジェクトの ML 部分を開始する前に、通常は集中型データレイクが既に存在しているか、構築されています。データレイクにある保管中のデータや、ML プロセスのさまざまな段階を通過するデータを保護しましょう。組織内のすべてのチームがすべてのデータにアクセスできる必要はありません。 データを分類して、最も細かな権限の細分性でさまざまなデータ部分へのアクセス権を制御し、データへのアクセスを継続的に監視します。

AWS では、集中型データレイクが Amazon S3 で AWS Lake Formation を使って実装されます。Amazon S3 でのデータレイクの保護と監視は、さまざまなサービスと機能の組み合わせを使って達成され、転送中および保管中のデータを暗号化し、詳細な AWS IAM ポリシー、S3 バケットポリシー、S3 アクセスログ、Amazon CloudWatch、AWS CloudTrail などのアクセスを監視します。 Building Big Data Storage Solutions (Data Lakes) for Maximum Flexibility では、これらのさまざまな機能を使った安全なデータレイクの構築について説明しています。

AWS IAM によるアクセスコントロールの実装に加えて、Amazon Macie を使って Amazon S3 のデータを保護し、分類します。Amazon Macie は、機械学習によって AWS 内の機密データを自動的に検出、分類、保護するフルマネージド型のセキュリティサービスです。このサービスは、個人識別情報 (PII) や知的財産などの機密データを識別し、こういったデータへのアクセスや移動の状況を可視化します。Amazon Macie では、データアクセスのアクティビティで継続的に異常を監視し、不正アクセスの危険や不注意によるデータ漏えいのリスクが検出された場合に、詳細なアラートを生成します。

探索またはトレーニングのためにデータレイクからコンピューティングインスタンスにデータを移動する場合、移動先のコンピューティングインスタンスへのアクセスも、厳しく制御で

Archived

きます。 この場合も、コンピューティングインフラストラクチャの転送中および保管中のデータを暗号化します。

データの準備および特徴量エンジニアリングの段階では、複数のオプションを使って AWS でのデータ探索のセキュリティを確保できます。Amazon SageMaker がホストする管理されたノートブック環境か、Amazon EMR Notebooks でデータを探索できます。 また、Amazon Athena や AWS Glue などのマネージドサービスを使ってもデータを探索できます。Amazon S3 のデータレイクからデータを移動する必要はありません。 また、2 つのアプローチを組み合わせて使うこともできます。 Amazon SageMaker ノートブックインスタンスにホストされている Jupyter ノートブックを使って、小規模なデータサブセットの探索、可視化、特徴量エンジニアリングを実行します。そして Amazon EMR や AWS Glue などのマネージド型 ETL サービスを使って、特徴量エンジニアリングを拡張します。

Amazon SageMaker ノートブックインスタンスにホストされている Jupyter ノートブックを使う場合は、ノートブックインスタンスを Amazon VPC にデプロイします。これにより、ネットワークレベルの制御を使って、ホストされたノートブックへの通信を制限できます。さらに、ノートブックインスタンスが送受信するネットワーク呼び出しを VPC フローログでキャプチャして、ネットワークレベルでのさらなる可視化と制御が可能になります。ノートブックを VPC にデプロイすることで、Amazon RDS や Amazon Redshift データウェアハウスのリレーショナルデータベースなどの VPC 内からアクセス可能なデータソースとシステムにクエリすることもできます。IAM を使うと、ノートブックインスタンスのウェブベースの UI へのアクセスをさらに制限し、VPC 内からのみアクセスできるようになります。

VPC 内のノートブックインスタンスから Amazon S3 のデータレイクに保存されたデータに通信するには、VPC インターフェイスエンドポイントの接続を使用します。これにより、ノートブックインスタンスと Amazon S3 間の通信が AWS ネットワーク内で完全かつ安全に実行されるようになります。ノートブックインスタンスで保管中のデータを暗号化するには、AWS KMS で管理されたキーを使って、Amazon SageMaker ノートブックインスタンスにアタッチされた EBS ボリュームを暗号化します。

Archived

Jupyter ノートブックサーバーには、基盤となるオペレーティングシステムへのウェブベースのアクセスがあります。これにより、開発者やデータサイエンティストは、追加のソフトウェアパッケージや Jupyter カーネルをインストールして環境をカスタマイズできます。デフォルトでは、ユーザーはローカルルート権限を取得する権限を有し、基盤となる EC2 インスタンスを完全に制御できます。このアクセスを制限することによって、ユーザーがルートアクセス権限を取得できないようにしつつ、ローカルユーザーの環境は制御できるようにします。

ルートアクセス権限へのアクセス制限に加えて、ライフサイクル設定を使って Jupyter ノートブックインスタンスを管理します。ライフサイクル設定は、ノートブックインスタンスの初回作成時またはノートブックインスタンスの起動時に、root として実行されるシェルスクリプトです。これによりカスタムツール、パッケージ、モニタリングをインストールできるようになります。ライフサイクル設定は複数のノートブックインスタンス間で変更および再利用ができるため、一度変更を行えば、再起動をすることで、管理されたノートブックインスタンス全体に新しい設定を適用できます。これにより、IT、運用、セキュリティチームは必要な制御ができると同時に、開発者やデータサイエンティストのニーズをサポートできます。

モデルをトレーニングする場合、単一のノートブックインスタンスが提供するよりも、より高いコンピューティング能力が多くの場合必要となります。AWS では、Amazon SageMaker を使ってトレーニングインスタンスのクラスターでモデルをトレーニングできます。Amazon SageMaker はトレーニングジョブの実行に使われる基盤となるインフラストラクチャをプロビジョニングし、データに対してアルゴリズムを実行してトレーニングされたモデルを生成します。

VPC でトレーニングインスタンスのクラスターを起動すると、ネットワークレベルの制御をトレーニングインスタンスに適用し、VPC エンドポイント経由で Amazon S3 や AWS ECR などの AWS のサービスへのアクセスを許可できます。セキュリティグループを使って、VPC 内の AWS のサービスにホストされていないデータソースへのトレーニングジョブのアクセスを制限します。また、プロキシサーバーとセキュリティグループを使って、VPC 以外のネットワークアクセスを制御します。KMS で管理された暗号化キーを使って、トレーニングノードの EBS ボリューム上のデータを暗号化し、トレーニング中の機密データの保護を強化します。

Archived

Amazon SageMaker コントロールプレーンの VPC エンドポイントを使って、VPC と Amazon SageMaker コントロールプレーン間のプライベート通信を有効にし、トレーニングジョブを管理、モニタリングします。

インスタンスのクラスターでモデルをトレーニングする場合、このプロセス中にアルゴリズムによってやりとりされる情報にも対応します。 分散型トレーニングジョブの一環で、TensorFlow などのフレームワークが係数などの情報を共有することは一般的です。これはトレーニングデータではなく、アルゴリズムが相互に同期し続けるために必要な情報です。このデータはデフォルトでは暗号化されていない場合があります。分散型トレーニングジョブの一部で、トレーニングジョブのノード間通信を暗号化するよう Amazon SageMaker を設定します。これで、これらのノード間で転送されるデータは、転送中に暗号化されるようになります。

ホストされた Jupyter ノートブック環境とトレーニングクラスターの保護に加えて、ML アルゴリズムの実装を保護することも不可欠です。Amazon SageMaker ではコンテナテクノロジーを使って、アルゴリズムとモデルのトレーニングおよびホストを行っています。これで Amazon SageMaker と他の ML パートナーが、アルゴリズムとモデルをコンテナとしてパッケージ化し、コンテナが ML プロジェクトの一部で使用できるようになります。さらに、Amazon SageMaker で使うテクノロジー、言語、フレームワークをパッケージ化できます。独自のコンテナを作成するときは、AWS Elastic Container Repository (ECR) でホストされているプライベートコンテナレジストリにそれを公開し、KMS で管理されたキーを使って AWS ECR でホストされているコンテナを暗号化します。

トレーニング中、Amazon SageMaker はコンテナを AWS ECR から取得し、トレーニングインスタンスで実行できるように準備します。小規模なデータセットの場合、Amazon SageMaker はトレーニング用の「ファイル」モードをサポートします。このモードでは、S3 バケットからのトレーニングデータをトレーニングインスタンスにアタッチされた EBS ボリュームにダウンロードします。これにより、アルゴリズムはローカルファイルシステムからトレーニングデータを読み取れるため、Amazon S3 と直接統合する必要がありません。 コンテナを使い Amazon S3 からオブジェクトをコピーすることによって、Amazon SageMaker はトレーニング中やホスティング中に、アルゴリズムとモデルをネットワーク分離できます。

Archived

ただし、トレーニングデータセットが大きい場合、トレーニングジョブ開始前にローカルファイルシステムにコピーをすることは効率的ではありません。 このような場合は、Amazon SageMaker の「パイプ」モードを使用し、Amazon S3 から直接データをトレーニングインスタンスにストリーミングします。つまり、トレーニングジョブを迅速に開始でき、速やかに完了できる他、必要なディスク容量を削減し、Amazon SageMaker で ML モデルをトレーニングする全体コストを軽減できます。

Amazon SageMaker によってトレーニング中に生成されたログは、AWS CloudWatch Logs に記録されます。AWS KMS で管理された暗号化キーを使って、AWS CloudWatch Logs によって取り込まれたログデータを暗号化します。

データ保護はトレーニングデータのみならず、推論に使われる本番環境/ライブデータにおいても重要です。例としては、AWS AI サービスまたは Amazon のホストモデルのエンドポイントに対して行われる推論の API 呼び出しがあります。 これらの API 呼び出しの HTTPS リクエストには署名が必要です。これにより、リクエスタの ID が検証され、リクエストペイロードデータは転送中に潜在的なリプレイ攻撃から保護されます。 AWS コマンドラインインターフェイス (AWS CLI) または AWS SDK の 1 つを使って API 呼び出しを行う場合、これらのツールはツール設定時に指定したアクセスキーで、リクエストに自動的に署名をします。ただし、カスタムコードを書いて HTTPS リクエストを AWS に送信する場合は、リクエストに署名する機能を実装する必要があります。

さらに、Amazon Translate や Amazon Comprehend などの AWS AI サービスには、AWS および関連する ML や AI テクノロジーの継続的な開発と改善のためにデータを使用するための規定があります。AWS サポートに連絡して、これらの目的でのデータの使用をオプトアウトできます。アカウントがオプトアウトされたという確認を受け取り、指示された手順に従うと、コンテンツは保存されなくなり、AWS AI サービスや Amazon ML/AI テクノロジーの開発や改善には使用されなくなります。

MLSEC 03: トレーニングされた ML モデルはどのように保護していますか?

Archived

ML モデルのトレーニングに使うデータの保護に加えて、トレーニングプロセスによって生成されたモデルアーティファクトへのアクセスも保護します。モデルをホストし、モデルのコンシューマーがモデルに対して推論を安全に実行できるようにします。 ML モデルのコンシューマーは内部または外部のアプリケーションやユーザーとなる場合がありますが、通常は、シンプルなエンドポイントまたは予測ができる API を介して統合されます。

AWS では、トレーニングフェーズの最後に生成された ML モデルは、通常 Amazon S3 に保持されます。VPC 内でトレーニングしたモデルをプライベート VPC エンドポイントを使って、Amazon S3 にアップロードします。これにより、AWS ネットワーク内で Amazon S3 に安全にモデルが転送されます。Amazon SageMaker を使ってモデルをトレーニングすると、サービスが転送中および保管中のモデルアーティファクトとその他のシステムアーティファクトを暗号化します。

Amazon SageMaker は推論のコンピューティングノードのクラスターに、トレーニングされたモデルをデプロイしてホストし、それに対し推論を実行するエンドポイント (HTTPS URL) を提供します。Amazon SageMaker でホストされるエンドポイントは、リアルタイム推論とバッチ変換予測の両方をサポートしています。どちらの場合も、ホストされたエンドポイントによって、同じ VPC ベースのネットワーク保護、モデルをホストするコンテナのネットワーク分離、推論ノードの EBS ボリュームの暗号化が可能になります。

Amazon SageMaker でホストされるエンドポイントは、IAM を使ってモデルと呼び出しを保護するセキュリティを強化します。これにより、モデルに対して推論を実行できる IAM ユーザー、IAM ロール、ソース VPC、IP がモデルを制御できます。さらに、AWS PrivateLink を使って、サービスとして他のコンシューマーにモデルを安全に共有できます。

Amazon SageMaker はトレーニングの一部としてキャプチャされたログと同様に、モデル推論アクティビティを AWS CloudWatch Logs に記録します。この場合も、AWS CloudWatch Logs によって取り込まれたログを、KMS で管理された暗号化キーを使って暗号化してください。これにより、推論中のモデルのアクティビティログが取得し、セキュリティおよび監査実行の要件に合った情報を必要に応じて詳細に提供できるようになります。

Archived

ML モデルのコンシューマーは多くの場合、モデルをホストしている環境外のアプリケーションからモデルを予測します。たとえば、ウェブアプリケーションはインターネットに接続されたエンドポイントに対して推論を行う場合があります。次の図は Amazon SageMaker でホストされているモデルにアクセスするためのサーバーレスアーキテクチャを示します。このアーキテクチャでは、エンドユーザーが API ゲートウェイに直接アクセスし、AWS Lambda と Amazon SageMaker モデルのエンドポイントは保護されたプライベートネットワーク内で運用されています。

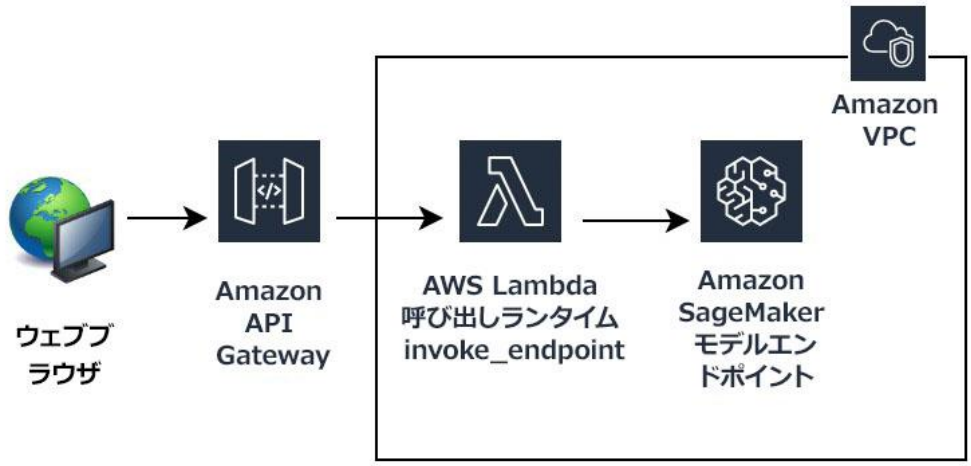

図 *14* - 推論のサーバーレスアーキテクチャ

このアーキテクチャのステップ概要:

1. コンシューマーアプリケーションは、リクエストパラメータ値を使って API ゲートウェイの API を呼び出します。
2. API Gateway はパラメータ値を Lambda 関数に渡します。Lambda 関数は値を解析し、Amazon SageMaker モデルエンドポイントに送信します。
3. モデルは予測を実行し、予測値を AWS Lambda に返します。Lambda 関数は返された値を解析し、API Gateway に送信します。
4. API Gateway は推論値でクライアントに応答します。

このアーキテクチャが使われているユースケースの詳細については、Call an Amazon SageMaker model endpoint using Amazon API Gateway and AWS Lambda を参照してください。

インシデント対応

ML に適用されるインシデント対応分野のセキュリティについてのベストプラクティスは、「AWS Well-Architected フレームワーク」のホワイトペーパーを参照してください。

主要な AWS のサービス

AWS でのデータセキュリティと監視に関する主要な AWS のサービスは次のとおりです。

- AWS IAM
- Amazon Virtual Private Cloud (Amazon VPC) と VPC エンドポイント
- Amazon SageMaker

リソース

AWS のセキュリティのベストプラクティスの詳細については、以下のリソースを参照してください。

ホワイトペーパー

- AWS セキュリティのベストプラクティス
- 最大の柔軟性があるビッグデータストレージソリューション (データレイク) の構築

ドキュメントとブログ

- OWASP セキュアコーディングのベストプラクティス
- Amazon SageMaker のための認証とアクセスコントロール
- Amazon API Gateway と AWS Lambda を使った Amazon SageMaker モデルエンドポイントの呼び出し
- Amazon SageMaker エンドポイントのサーバーレスフロントエンドの構築

Archived

信頼性の柱

信頼性の柱には、インフラストラクチャまたはサービスの障害からの復旧、必要に応じた動的なコンピューティングリソースの獲得、設定ミスや一時的なネットワークの問題などによる障害の軽減などのシステムの能力が含まれます。

設計の原則

クラウドには、システムの信頼性強化に役立つ多くの原則があります。標準的なプラクティスについては、「信頼性の柱: AWS Well-Architected フレームワーク」ホワイトペーパーを参照してください。また、ML ワークロードの信頼性向上に特化した原則もいくつかあります。

- **自動化によるモデル入力の変更の管理:** ML ワークロードには、モデルのトレーニングに使用されるデータの変更を管理する追加要件があり、障害や人為的ミスの発生時にまったく同じバージョンのモデルを再作成できるようになっています。自動化でバージョンと変更を管理することで、信頼性と一貫性のある方法で復旧できます。
- **1 回のみのトレーニングで環境をまたいだデプロイが可能:** 同じバージョンの ML モデルを複数のアカウントまたは環境にデプロイする場合、アプリケーションコードに適用される 1 回のみの構築と同じプラクティスをモデルトレーニングに適用する必要があります。特定のバージョンのモデルを 1 回だけトレーニングし、出力モデルアーティファクトを使って、複数の環境にデプロイすることによって、複数の環境にあるモデルに予期しない変更が発生しないようにする必要があります。

定義

クラウド内で信頼性に関するベストプラクティスには 3 つの領域があります。

- 基盤
- 変更管理

Archived

- 障害の管理

ベストプラクティス

基盤

このサブセクションに該当する ML ワークロードに固有の基本的なプラクティスはありません。「AWS Well-Architected フレームワークの信頼性の柱」ホワイトペーパーで説明されているプラクティスを使って、基本的な機能を確認する必要があります。

変更管理

> MLREL 01: 機械学習モデルと予測エンドポイントへの変更をどのように管理しますか?

ML ワークロードの場合、モデルに加えられた変更や予測エンドポイントへの変更について、トレーサビリティを確保できるメカニズムを構築することが重要です。これにより、より迅速なトラブルシューティングが可能になり、新しいモデルが想定どおりに動作していない場合に、以前のバージョンのモデルに戻せます。モデルのデプロイされたバージョンは、限れられたリソースへの読み取り専用アクセスを持つアーティファクトリポジトリで保護されている、バージョン管理された特定のモデルアーティファクトにさかのぼって追跡できる必要があります。モデルアーティファクトはビジネスで定義された保持期間中保持されます。

オーバーヘッドと手動での介入を減らすために、モデルまたはエンドポイントへの変更は、パイプラインを通じて自動的に行われる必要があります。これにはビジネスの必要性に応じた変更管理追跡システムとの統合を含みます。バージョン管理されたパイプライン入力とアーティファクトによるトレーサビリティによって、変更を追跡し、変更に失敗した場合には自動でロールバックできるようになります。

モデルに変更をデプロイするには、標準の A/B テスト戦略の使用をお勧めします。この戦略では、定義された割合のトラフィックが新しいモデルに転送され、残りのトラフィックは古いモデルに転送されます。この場合、ロールバックには DNS を古いバージョンに戻すことも含まれます。ロールバックまたはロールフォワードが必要なタイミングを効果的に特定するために、モ

Archived

デルのパフォーマンスを評価するメトリクスを導入して、ロールバックまたはロールフォワードアクションが必要なときにアラートが出るようにする必要があります。ロールバックまたはロールフォワードのアーキテクチャ設計には、各モデルについて以下の評価が重要です。

- モデルアーティファクトはどこに保存されますか?
- モデルアーティファクトはバージョン管理されていますか?
- 各バージョンにはどのような変更が含まれますか?
- デプロイされたエンドポイントの場合、デプロイされるモデルのバージョンは何ですか?

リソースをさかのぼって追跡し、モデルを自動的にデプロイするトレーサビリティメカニズムを構築することによって、信頼性のあるロールバックと復元の機能が実現します。さらに、モデルを複数の環境にデプロイする際の信頼性を確保するには、1 回のみのトレーニング戦略を使って、デプロイプロセスでの不注意によるばらつきを削減する必要があります。

AWS でモデルを作成するときは、既存のサービス機能を使用し、基準を導入して、ML モデルを以前のバージョンに復元できるようにしておくことをお勧めします。

Amazon Transcribe などの AWS の AI サービスの場合、AWS がエンドポイント予測の作成に使われるデプロイされたエンドポイントのバージョン管理を行います。AWS はエンドポイントのサービスとしてのホスティングに関連する変更管理を担当します。

AWS の ML サービスと AWS の ML フレームワークとインターフェイスには共通の基準があり、変更管理のトレーサビリティと、ロールフォワードとロールバック機能を提供しています。モデルアーティファクトをバージョン管理されたオブジェクトとして Amazon S3 に保存して、モデルの耐久性と可用性を確保します。モデルトレーニングとモデルホスティングに使用されるコンテナイメージは、AWS Elastic Container Registry (ECR) など耐久性があり安全なイメージリポジトリに保存するようにします。さらに、モデルアーティファクトとコンテナイメージの整合性を守るため、IAM ロールベースのアクセスを使ってモデルアーティファクトへのアクセスを制限し、リソースに適用されるポリシーに最小権限を実装します。アーティファクトをコードとして作成するために使用されるすべての設定を、AWS CodeCommit などのマネージド型ソース管理システムに保存します。

Archived

さらに、アーティファクトのトレーサビリティがあることで、特定のバージョンにロールフォワードまたはロールバックができます。デプロイされたモデルアーティファクトのバージョン間の変更に重きを置いた、モデルアーティファクトのバージョン履歴のマニフェストを作成、運用します。これを行うには変更マニフェストデータを永続的なストアに保存します。または、エンドツーエンドでモデルの開発とデプロイを制御する自動デプロイパイプラインを通じても最適に実現できます。これは後述の**障害の管理**で説明しています。可能な場合は、本番環境用バリアントを介して SageMaker にネイティブの A/B テスト機能を使い、複数のモデルでの変更を評価し、変更にすばやく対応します。

AWS での ML フレームワークとインターフェイスには、再利用可能な設計基準を作成する多くの機能が備わっており、AWS CloudFormation テンプレートと AWS 開発者ツールという形でワークロードの自動化と復元可能性を向上させます。「AWS Well-Architected フレームワークの信頼性の柱」ホワイトペーパーに記載のとおり、ロールバックとロールフォワード機能をサポートするデプロイ戦略を作成して、複数のモデルにまたがる変更を評価し、変更にすばやく対応します。

自動ロールバックとロールフォワード機能を実装すると、変更の失敗、システム障害やモデルパフォーマンス低下から復旧できます。この機能には、明確に定義されたバージョン管理戦略と合わせて、問題が検出されたときに変更を追跡し、元に戻すメカニズムが必要です。モデル評価に必須のメトリクスがすべて定義され、収集されていることを確認します。Amazon CloudWatch などのモニタリングシステムでメトリクスを収集し、モデルのバージョンが想定どおりに動作していない場合に、ロールバックイベントをトリガーするようアラームを設定します。

MLREL 02: ML モデルへの変更は、ワークロード全体でどのように連携されますか?

既存のワークロード機能の中断を最小限に抑えて、またはまったく中断することなく ML モデルへの変更を反映するには、インターフェイスアプリケーションを設計するときに、依存するシステムとアプリケーションの統合方法を考慮することが重要です。柔軟なアプリケーションと API 設計は、インターフェイスアプリケーションから変更を抽出するのに役立ちます。さ

Archived

らに、依存するシステムやアプリケーションに変更を伝え、連携するための戦略も欠かせません。定義済みの変更管理戦略に従って変更を反映し、影響のあるチームに連絡してそれらの変更を追跡できるようにします。

アプリケーションレベルの変更の管理と制御と同様の方法で、モデルの新しいバージョンのデプロイを管理します。ML モデルの変更管理戦略には、変更の伝達方法とデプロイ方法を考慮に含めて、サービスの中断やモデルのパフォーマンスと精度の低下を回避する必要があります。モデルの新しいバージョンをデプロイする戦略には、ターゲット環境にデプロイする前後に実行する検証アクティビティも含める必要があります。

変更管理が一貫性を持って正しく実行されるようにするベストプラクティスは、すべての変更を CI/CD パイプラインを通じて実行し、最小権限の原則に従ったアクセスコントロールを使用してデプロイプロセスを実施することです。手動または自動化された品質ゲートと自動化を組み合わせてデプロイを制御することにより、デプロイ前に依存するシステムと効果的に変更を検証できます。

> MLREL 03: 予測のモデルをホストするエンドポイントはどのようにスケーリングしますか?

モデルのエンドポイントを自動的にスケールできる機能を実装することが重要です。これにより、ワークロードの需要の変化に応じて確実に予測を処理できます。エンドポイントをスケールするには、エンドポイントをモニタリングし、現在の需要を支えるためのリソースの追加または削除をトリガーするしきい値を特定する必要があります。スケーリングのトリガーを受け取ったときには、そのエンドポイントをサポートするバックエンドリソースをスケールするソリューションの準備ができている必要があります。エンドポイントに対して負荷テストを実行して、効果的なスケーリングと予測の確実な処理ができるかを検証できるようにします。

AWS では、この領域におけるエンドポイントのスケーリングと責任は、使用する AI/ML サービスによって異なります。Amazon Comprehend、Amazon Polly、Amazon Translate などの AWS の AI サービスの場合、エンドポイントは AWS によって自動的に管理、スケーリングされます。Amazon SageMaker などの AWS の ML サービスの場合は、アベイラビリティーゾーン

間の自動スケーリング機能はサービス内で設定できます。高可用性を実現するには、複数のアベイラビリティーゾーンにまたがる自動水平スケーリングを本番環境用バリアントすべてに設定します。設定完了後、障害テストを実行してエンドポイントが障害から回復でき、可用性の要件に従うことができるかを確認することが重要です。

AWS の ML フレームワークとインターフェイスの場合は、自動スケーリング機能および負荷分散機能をセットアップします。モデルが EC2 インスタンスでホストされている場合も、EC2 インスタンスまたは AWS Fargate でホストされている ECS または EKS のコンテナを使っている場合も同様です。自動スケーリング機能を使って、異常のあるインスタンスを自動的に置き換えて EC2 インスタンスを自己修復することもできます。この領域の基本的なベストプラクティスについては、「AWS Well Architected フレームワークの信頼性の柱」を参照してください。

障害の管理

> MLREL 04: トレーニングされた ML モデルに障害が起きた場合やモデルが不注意によって損失した場合、どのように復旧しますか?

トレーニング済みの ML モデルは、パッケージ化されたアーティファクトであり、障害や損失が発生した場合に回復可能である必要があります。リソースの障害や損失は、システム障害から人為的エラーまで、さまざまなイベントによって発生する可能性があります。ML モデルの場合、次の障害シナリオを念頭に置き、復旧目標と比較して、適切な戦略がデプロイされていることを確認する必要があります。人為的なエラーによってモデルアーティファクトを誤って削除した場合、または基盤となるストレージが利用できなくなった場合、そのアーティファクトを簡単に復元または再作成できますか?

モデルアーティファクトを不注意による削除から守るには、アーティファクトの使用に関して最小限必要な権限のみを許可し、特権ユーザーによる削除に備え MFA などの追加メカニズムを実装し、定義済みの災害対策戦略の要件に応じてアーティファクトのセカンダリコピーを保存することで、モデルアーティファクトが保護されるようにします。並行して、アーティファクトのバージョン管理戦略を導入すると、特定のバージョン管理されたアーティファクトの復旧が可能になります。副次的に、モデルアーティファクトの特定バージョンの再作成が可能にな

ることによって、障害や損失からの保護も強化されます。同じ保護メカニズムとバージョン管理戦略を適用して、データとトレーニングコードを含む入力をモデル化します。

AWS ではモデルの障害と復元可能性に関連するベストプラクティスは、使用する AWS のサービスによって異なります。Amazon Polly などの AWS の AI サービスの場合は、AWS が保護と管理を行う構築済みモデルを使用します。Amazon SageMaker などの AWS の ML サービスの場合は、Amazon S3 にモデルアーティファクトを作成して保存します。この場合、AWS IAM が提供するアクセスコントロールを活用して、リソースの保護のためにモデルの入力とアーティファクトを保護できます。さらに、Amazon S3 のバージョン管理と、オブジェクトのタグ付けがされたモデルアーティファクトのバージョン管理とトレーサビリティを組み合わせたものなどのメカニズムを使って、障害発生時の復元可能性を確保します。

> MLREL 05: モデルホスティングリソースに障害が起きた場合やリソースが不注意によって損失した場合、どのように復旧しますか?

ML ワークロードのあらゆるコンポーネントを復旧できれば、ソリューションはリソースの障害や損失に対応できるようになります。リソースの障害や損失は、システム障害から人為的エラーまで、さまざまなイベントによって発生する可能性があります。ML ワークロードの場合、次の障害シナリオを念頭に置き、復旧目標と比較して、適切な戦略がデプロイされていることを確認する必要があります。モデルエンドポイントが誤って削除された場合、そのエンドポイントを再作成して、エンドポイントを特定のバージョンに復元できますか?

AWS では障害の管理に関連するベストプラクティスは、使用する AWS のサービスによって異なります。Amazon Polly などの AWS の AI サービスは AWS によってホスト、スケーリング、管理がされるため、お客様はエンドポイント復旧の責任を負っていません。AWS の ML サービス、AWS の ML インフラストラクチャと ML フレームワークのベストプラクティスは、モデル予測のホスティングを担当するエンドポイントを、ビジネスが定義した特定のバージョンまたは特定の時点に完全に復旧できるようにしておくことです。モデルエンドポイントを復旧できるようにするには、そのエンドポイントの作成に使用されたすべてのコンポーネントと設定

Archived

が、管理しているバージョン管理戦略に含まれている必要があります。これにより、利用不可能になったコンポーネントの完全な復旧が可能になります。

たとえば、SageMaker でエンドポイントを再作成するには、まったく同じバージョンのモデルアーティファクト、コンテナイメージ、エンドポイント設定などの複数の SageMaker コンポーネントが必要となります。特定のモデルアーティファクトのバージョンを再作成するには、そのモデルアーティファクトの作成に使用されたトレーニングデータのバージョン管理戦略と、アルゴリズム把握しておく必要もあります。障害発生時にパイプライン内のコンポーネントを再作成できるようにするには、依存リソースもすべて、バージョン管理します。バージョン管理に加えて、徹底してデプロイをドキュメント化しているマニフェストにバージョン管理されたアーティファクトすべてを含めること、バージョン管理されたアーティファクトすべてを、セキュリティの柱で説明されている最小権限の原則を使って保護することが重要です。

リソース

信頼性のベストプラクティスの詳細については、以下のリソースを参照してください。

ホワイトペーパー

- 信頼性の柱 - AWS Well-Architected フレームワーク

パフォーマンス効率の柱

パフォーマンス効率の柱では、コンピューティングリソースを効率的に使ってシステム要件を満たし、需要が変化し技術が発展するのに合わせて効率性を維持する方法について重点を置きます。

設計の原則

クラウドには、システムのパフォーマンス効率の強化に役立つ多くの原則があります。標準的なプラクティスについては、「パフォーマンス効率の柱 :AWS Well-Architected フレームワーク」ホワイトペーパーを参照してください。また、ML ワークロードのパフォーマンス効率の向上に特化した多くの原則もあります。

Archived

- **ML ワークロード用にコンピューティングを最適化する:** ほとんどの ML ワークロードは非常にコンピューティング集約的です。これは、大量のデータとパラメータに対して大量のベクトル乗算とベクトル加算を実行する必要があるためです。特に深層学習では、より大きなキューデプスを備え、より多くの算術論理演算ユニットとレジスタを搭載した、大量の並列処理が可能なチップセットにスケールする必要があります。そのため、深層学習モデルをトレーニングするためのプロセッサタイプとして GPU が推奨されます。適切なコンピューティングリソースの選択に関する詳細については、以下の「MLPER 01」セクションで説明します。
- **モデルのレイテンシーとネットワーク帯域幅のパフォーマンス要件を定義する:** ML アプリケーションによっては、ビジネス要件を満たすためにほぼ瞬時の推論結果が必要になる場合があります。レイテンシーをできる限り小さくするには、最寄りの API エンドポイントへのコストの高いラウンドトリップを排除する必要があります。レイテンシーの低減は、デバイス自体で直接推論を実行することで実現できます。これはエッジでは機械学習と呼ばれます。この要件に当てはまる一般的なユースケースは、工場での予知保全です。エッジでの低レイテンシーとほぼリアルタイムの推論により、障害の兆候を早期に発見できるため、実際に障害が発生する前に、機械修理にかかる膨大なコストを軽減できる可能性があります。
- **システムパフォーマンスを継続的にモニタリングおよび測定する:** モデルに対する予測の構築、トレーニング、ホスティング、実行に関連する主要なメトリクスを特定し、定期的に収集することで、一連の主要な評価基準に基づく全体的な成功を継続的に監視できます。ML ワークロードの各フェーズをサポートするために使用されるシステムレベルのリソースを検証するには、コンピューティング、メモリ、ネットワークなどのシステムレベルのリソースを継続的に収集して監視することが重要です。ML ワークロードの要件はフェーズによって変化します。たとえば、前の 2 つの設計の原則で説明したように、トレーニングジョブはメモリ集約型で、推論ジョブはコンピューティング集約型です。

Archived

定義

クラウドでのパフォーマンス効率を向上させるには、次の 4 つのベストプラクティス領域があります。

- 選択 (コンピューティング、ストレージ、データベース、ネットワーク)
- 確認
- モニタリング
- トレードオフ

高性能なアーキテクチャを選択する際は、データ駆動型のアプローチを選択します。ハイレベルな設計から、リソースタイプの選択と設定まで、アーキテクチャのあらゆる側面に関するデータを収集します。選択した内容を定期的にレビューすることで、絶えず進化し続けている AWS サービスを確実に活用できます。モニタリングを実施すれば、予想したパフォーマンスからのずれを把握し、その対策を講じることができます。さらに、圧縮やキャッシュを使用したり、整合性に関する要件を緩和したりするなど、アーキテクチャにおけるトレードオフを行ってパフォーマンスを向上させることができます。

Archived

ベストプラクティス

選択

> MLPER 01: モデルのトレーニングとホスティングに最適なインスタンスタイプをどのように選択していますか?

一般的な機械学習パイプラインは、一連のステップで構成されます。このプロセスは、トレーニングデータとテストデータの収集から始まり、その後、収集したデータの特徴量エンジニアリングと変換が行われます。最初のデータ準備フェーズの後に、ML モデルのトレーニング、評価、チューニングが行われてから、最終段階となるデプロイ、サービスの提供、ライフサイクルを通じたモデルのパフォーマンスモニタリングへと続きます。コンピューティングニー

ズはフェーズによって異なるため、これらの各フェーズにおける ML ワークロードのパフォーマンスは慎重に検討する必要があります。 たとえば、モデルのトレーニングでは、GPU インスタンスのパワークラスターが必要になる場合がありますが、推論では、CPU インスタンスの自動スケーリングクラスターでパフォーマンス要件を十分に満たす場合があります。

データサイズ、データ型、アルゴリズムの選択により、どの構成が最も効果的であるかが大きく変わる可能性があります。同じモデルを繰り返しトレーニングする場合は、さまざまなインスタンスタイプに対して初期テストを実行し、パフォーマンスとコスト効率性に優れた構成を見つけるよう強くお勧めします。一般的なガイドラインとして、GPU インスタンスはほとんどの深層学習目的に推奨されます。GPU インスタンスでは、新しいモデルのトレーニングを CPU インスタンスよりも高速に実行できるためです。マルチ GPU インスタンスがある場合、または GPU を搭載した多数のインスタンスに対して分散トレーニングを使用する場合は、ほぼ線形にスケールできます。ただし、GPU 上で最も効率的なトレーニングを行うアルゴリズムは、必ずしも効率的な推論のために GPU を必要としない可能性があることに注意してください。

AWS では、さまざまな機械学習 (ML) のユースケースに合わせて最適化された一連のインスタンスタイプが用意されています。インスタンスタイプには、CPU、GPU、FPGA、メモリ、ストレージ、ネットワーク容量のさまざまな組み合わせがあります。また、GPU を使用した推論アクセラレーターを AmazonElastic Inference 経由で Amazon EC2 または Amazon SageMaker インスタンスにアタッチしたり、AWS によってカスタム設計された高性能 ML 推論チップである AWS Inferentia を搭載した Amazon EC2 インスタンスを使用したりすることもできます。これにより、モデルをトレーニングするか、トレーニングされたモデルに対して推論を実行するかにかかわらず、ML のユースケースに合わせて最適化されたリソースを適切に組み合わせて柔軟に選択できます。各インスタンスタイプには 1 つ以上のインスタンスサイズが含まれており、ターゲットワークロードの要件に合わせてリソースをスケールできます。

最後に、XGBoost などの一部のアルゴリズムは、CPU 計算用に最適化されたオープンソースアルゴリズムを実装していますが、AWS Deep Learning AMI (DLAMI) では、Caffe などの一部のフレームワークは GPU サポートでのみ動作し、CPU モードでは実行できません。

Archived

トレーニングとホスティングに使用するインスタンスの選択にかかわらず、インスタンスまたは Amazon SageMaker エンドポイントの負荷テストを実施し、インスタンスまたはエンドポイントがサポートできるピーク負荷と、同時実行数の増加に伴うリクエストのレイテンシーを確認してください。

> MLPER 02: どのようにして最適なパフォーマンスを維持しながら ML ワークロードをスケールしますか?

需要の増加に対応するため、そして最適なパフォーマンスを達成するために ML アーキテクチャをスケールする方法を検討する際には、マネージドサービスエクスペリエンスを通じてモデルをデプロイすることと、自分で ML モデルをデプロイして管理することの違いを区別することが重要です。

AWS では、Amazon SageMaker でマネージド型 ML エクスペリエンスを提供していますが、通常は、MXNet、TensorFlow、Caffe、Chainer、Theano、PyTorch、CNTK フレームワークを提供する Deep Learning AMI (DLAMI) を EC2 インスタンスで使用して、自分でモデルを管理します。このセクションでは、両方のユースケースについて説明するとともに、AWS AI サービスの使用に関する 3 つ目のユースケースについても簡単に説明します。

Amazon SageMaker は、ユーザーの代わりに本番環境のコンピューティングインフラストラクチャを管理し、組み込みの Amazon CloudWatch のモニタリングとログ記録を使用してヘルスチェック、セキュリティパッチの適用、その他の日常的なメンテナンスを行います。Amazon SageMaker Model Monitor は、本番環境の ML モデルを継続的にモニタリングし、時間の経過とともにモデルのパフォーマンスを低下させる可能性があるデータドリフトなどの逸脱を検出し、ユーザーに修復アクションを実行するよう警告します。

さらに、Amazon SageMaker のホスティングは、Application Auto Scaling を使用して、アプリケーションに必要なパフォーマンスに合わせて自動的にスケールします。Application Auto Scaling を使用すると、推論能力を自動的に調整して、予測可能なパフォーマンスを低コストで維持できます。また、エンドポイントの構成を変更することで、ダウンタイムを発生させることなく、手動で EC2 インスタンスの数とタイプを変更できます。

Archived

深層学習ワークロードには、Amazon Elastic Inference (EI) でスケールしてスループットを高め、深層学習モデルに対するリアルタイム推論のレイテンシーを減らすことができます。Amazon Elastic Inference では、GPU を使用した推論アクセラレーションを任意の Amazon EC2 インスタンスにアタッチできます。この機能は、Amazon SageMaker のノートブックインスタンスやエンドポイントでも利用できるため、組み込みのアルゴリズムと深層学習環境を低コストで高速化できます。

深層学習ニューラルネットワークは、複数のプロセッサを活用し、さまざまなタイプと数のプロセッサにワークロードをシームレスかつ効率的に分散するのに最適です。クラウドを介して広範なオンデマンドリソースを利用できるため、事実上無制限のリソースをデプロイし、あらゆる規模の深層学習モデルに取り組むことができます。クラウドでの深層学習では、分散ネットワークを活用することで、深層学習アプリケーションをより迅速に設計、開発、トレーニングできます。

Ubuntu および Amazon Linux 向けの AWS 深層学習 AMI では、ほぼ線形のスケーリング効率を備えた TensorFlow 深層学習モデルの分散トレーニングがサポートされます。AWS 深層学習 AMI には、Horovod 分散トレーニングフレームワークの最適化されたバージョンと統合された TensorFlow の拡張バージョンがあらかじめ組み込まれています。この最適化により、ノードが相互に直接通信できる (集中ノードを経由したり、リングオールリデュースアルゴリズムを使用して平均勾配を計算したりする必要がない) 高パフォーマンスの実装が可能になります。前述の Horovod 分散に加えて、Chainer や Keras の使用など、分散トレーニングをサポートするその他のフレームワークが多数あります。

AWS AI サービスを使用すると、独自のモデルを開発してトレーニングする代わりに、トレーニング済みサービスへの API 呼び出しを通じてアプリケーションのインテリジェンスを高めることができます。AWS AI サービスを使用するアーキテクチャをスケールする場合は、リソースとレート制限 (API リクエスト率など) をモニタリングする必要があります。サービス制限の管理方法に関する詳細は、一般的な Well-Architected フレームワークの信頼性セクションを参照してください。

Archived

リソース

当社のパフォーマンス効率性のベストプラクティスの詳細については、以下のリソースを参照してください。

ドキュメントとブログ

- TensorFlow で行うスケーラブルなマルチノードトレーニング
- Amazon Elastic Inference
- 自動スケーリングを使用して、Amazon SageMaker エンドポイントのロードテストおよび最適化を行う
- DLAMI のインスタンスタイプを選択する

ホワイトペーパー

- パフォーマンスの柱 - AWS Well Architected フレームワーク

Archived

コスト最適化の柱

コスト最適化の柱には、システムを全ライフサイクルにわたり向上・改善する継続的プロセスが含まれます。最初の PoC (実証支援) の初期設計から、製品ワークロードの継続運用まで、コスト意識の高いシステムを構築・運用する際に紙形式のプラクティスを採用すれば、ビジネス上の成果を上げながら、コストの最小化と投資利益率の最大化の両方を達成できるようになります。

設計の原則

クラウドには、システムのコスト最適化の改善に役立つ多くの原則があります。標準的なプラクティスについては、「AWS Well-Architected フレームワーク」ホワイトペーパーを参照してください。また、ML ワークロードのコスト最適化に特化した多くの原則もあります。

- **マネージドサービスを使用して所有コストを削減:** ML ワークロードの各フェーズで適切なマネージドサービスを採用し、「支払いは使用した分だけ」モデルを活用します。たとえば、モデルのチューニングは、通常、コンピューティングと時間のかかるプロセスです。不要な課金を回避するため、分散トレーニングクラスターの作成、トレーニングジョブの実行によるモデルのチューニング、完成したモデルの保持、トレーニング完了時のクラスターの自動破棄を行うマネージドサービスを使用します。
- **小さなデータセットから試す:** ML ワークロードは大規模で高品質のトレーニングデータセットにより恩恵を受けますが、低コストで迅速に反復できるように、小さなコンピューティングインスタンス (またはローカルシステム) で小さなデータセットから開始します。 実験期間が終わったらスケールアップし、分散コンピューティングクラスターで利用可能な完全なデータセットでトレーニングします。 完全なデータセットでトレーニングする場合は、クラスターノードにデータを保存する代わりに、クラスターにデータをストリーミングします。
- **適切なサイズのトレーニングとモデルホスティングインスタンス**: モデルのトレーニングとホスティングでは、実験を行って必要かつ最適なコンピューティング性能を特定します。小さなインスタンスから開始し、最初にスケールアウトしてからスケールアップすることをお勧めします。また、トレーニング中およびホスティング中に CPU と GPU のニーズの違いを測定します。 一部の ML モデルでは、トレーニングのためにパワフルな GPU インスタンスが必要ですが、デプロイされたモデルに対する推論では、GPU のフルパワーは必ずしも必要ありません。
- **消費パターンに基づいて推論アーキテクチャを考慮する**: モデルには、e コマースの不正検出のように、リアルタイム予測のために常時利用可能でなければならないものもあれば、e コマースの予測モデルのように、一定期間ごとに利用できるだけでよいものもあります。最初のケースでは、24 時間 365 日のホスティングモデルのコストが理にかないますが、2 番目のケースでは、モデルをオンデマンドでデプロイし、予測を実行してからモデルを停止することで、大幅なコスト削減を実現できます。

Archived

- **全体的な ROI と機会コストを定義する**: ML の導入コストと、ML の導入に頼らない場合の機会コストを比較します。 データサイエンティストの時間やモデルの市場投入までの時間などの特殊なリソースは、最も高価で制約のあるリソースである可能性があります。 最もコスト効率の高いハードウェアを選択しても、そのハードウェアによって実験や開発の速度が制限される場合には、コストを最適化できない可能性があります。

定義

4 つの分野のベストプラクティスなど、クラウド内でのコストの最適化:

- 費用対効果の高いリソース
- 需要と供給の一致
- 費用認識
- 時間経過を伴う最適化

他の柱と同様、評価する必要があるトレードオフも存在します。たとえば、最適化したいのは市場投入までのスピードですか、それともコストですか? 市場投入のスピードを向上する、新しい機能を導入する、締め切りに間に合わせるといったケースでは、前払いコストの投資を最適化するよりも、スピードを重視して最適化することが最善です。設計上の決定は、実際のデータに導かれずに、急いで行われる場合があります。なぜなら、コスト最適化のベンチマークを行う時間をかけず、過剰な投資を「念のため」行う傾向があるからです。その結果、過剰なプロビジョニングと、あまり最適化されていないデプロイが生じることになります。以下のベストプラクティスでは、デプロイにおける初期段階でのコスト最適化と継続的なコスト最適化について、そのテクニックと戦略的ガイダンスを紹介します。

Archived

ベストプラクティス

費用対効果の高いリソース

MLCOST 01: データのラベル付けのコストをどのように最適化しますか?

ML モデルを構築するには、開発者やデータサイエンティストが ML モデルのトレーニングに使用するデータセットを準備する必要があります。開発者がアルゴリズムを選択してモデルを構築し、それをデプロイして予測を行う前に、人間のアノテーターが手動で何千もの例を確認し、ML モデルのトレーニングに必要なラベルを追加します。このプロセスには時間とコストがかかります。

AWS では、Amazon SageMaker Ground Truth が、Amazon Mechanical Turk、サードパーティーベンダー、またはその従業員を通じて人間のアノテーターによるデータのラベル付けタスクを簡素化します。Amazon SageMaker Ground Truth は、これらの人間の注釈からリアルタイムで学習し、アクティブな学習を適用して残りのデータセットのほとんどを自動的にラベル付けし、人間によるレビューの必要性を減らします。Amazon SageMaker Ground Truth は、人間と ML 機能を組み合わせることで、高精度なトレーニングデータセットを作成し、時間を節約するとともに複雑さを軽減し、すべてを手動で注釈する場合と比較してコストを削減できます。

MLCOST 02: ML の実験中にどのようにコストを最適化しますか?

ノートブックは、少量のデータを調査および実験するための一般的な方法です。機械学習では一般的に、データセットの小さなサンプルをローカルで反復処理してから、データセット全体を分散方式でトレーニングするために拡張します。

AWS では、Amazon SageMaker ノートブックインスタンスに、データの小さなサンプルを調べるために使用できるホスト型の Jupyter 環境が用意されています。ノートブックインスタンスは、アクティブに使用していないときには停止します。実際には、作業をコミットしてから

Archived

停止し、再び必要になったときに再開します。ストレージは保持され、ライフサイクル設定を使用してパッケージのインストールやリポジトリの同期を自動化できます。

モデルのトレーニングを実験するときには、Amazon SageMaker ノートブックの「ローカル」モードを使用し、別のマネージドトレーニングクラスターではなく、ノートブックインスタンス自体でモデルをトレーニングします。新しいトレーニングを待ったり、構築されるクラスターを毎回ホスティングしたりすることなく、作業を反復してテストできます。これにより、マネージドトレーニングクラスターの作成に関連する時間とコストの両方が節約されます。実験は、ノートブックの外部 (ローカルマシンなど) で行うこともできます。ローカルマシンから、SageMaker SDK を使用して AWS でモデルをトレーニングおよびデプロイできます。

実験の際には、AWS Marketplace for Machine Learning を確認することも忘れないでください。ここには、機械学習のアルゴリズムとモデルが絶えず追加されています。AWS Marketplace のモデルは Amazon SageMaker に直接デプロイされ、ML アプリケーションをすばやく構築できます。これにより、モデルの開発に関連するコストと時間の両方が節約されます。

AWS Marketplace for Machine Learning では、開発したモデルを販売することもできます。これにより、社内モデルを収益化し追加の収益源を確保できます。知的財産を保護しながら、カスタムモデルを他の顧客に公開できます。Amazon SageMaker では、基盤となるモデルを公開することなく、安全なエンドポイントを介してモデルにアクセスできます。

> MLCOST 03: ML トレーニング向けに最もコスト効率が優れたリソースはどのように選択しますか?

完全なトレーニングデータを使用して ML モデルをトレーニングする準備ができたときに避けたいのは、データセットが小さい場合を除き、ノートブックインスタンスでトレーニングジョブをローカルモードで実行することです。代わりに、分散トレーニング用の 1 つまたは複数のコンピューティングインスタンスを使用してトレーニングクラスターを起動します。ワークロードに基づいて、トレーニングクラスター内に適切なコンピューティングインスタンスをサイズ設定します。

Archived

AWS で、Amazon SageMaker トレーニング API を使用してマネージドインスタンスのクラスターを作成します。トレーニングクラスターで複数のインスタンスを使用し、分散トレーニングを実行してトレーニング時間を短縮できます。トレーニングが完了すると、トレーニングクラスター内のすべてのインスタンスが自動的に終了します。

性能構成が異なるさまざまなタイプのインスタンスをトレーニングに使用できますが、使用する ML アルゴリズムに従ってトレーニングインスタンスを適切にサイズ設定することが重要です。シンプルなモデルは、ハードウェア並列処理による性能の向上の恩恵を受けられない場合があるため、大きなインスタンスで迅速にトレーニングされない場合があります。GPU 通信オーバーヘッドが大きいため、トレーニングが遅くなることさえあります。小さなインスタンスから開始し、最初にスケールアウトしてからスケールアップすることをお勧めします。また、選択した ML アルゴリズムでチェックポイント機能がサポートされている場合は、Amazon SageMaker でマネージドスポットトレーニングを使用して評価することで、コストを削減します。

トレーニング用に最適化されたインスタンスタイプを選択することに加えて、より迅速にトレーニングできるように、ML フレームワークの最適化されたバージョンを選択することが重要です。AWS では、Amazon EC2 インスタンスファミリー全体にわたる高パフォーマンストレーニングのための最適化を備えた、TensorFlow、Chainer、Keras、Theano などのフレームワークの最適化されたバージョンが用意されています。

大量のトレーニングデータを処理する場合、Amazon SageMaker のパイプモードでは、Amazon SageMaker のファイルモードよりも大幅に高い読み取りスループットが提供されます。ファイルモードでは、モデルのトレーニングを開始する前にローカルの Amazon EBS ボリュームにデータがダウンロードされますが、パイプモードでは、ML モデルのトレーニング時に Amazon S3 から Amazon SageMaker にデータがストリーミングされます。つまり、トレーニングジョブがより早く始まり、より速く完了し、ジョブに必要なディスク容量が少なく済みます。これにより、Amazon SageMaker での ML モデルのトレーニングにかかる全体的なコストが削減されます。

ML モデルの正しいハイパーパラメータセットを特定するには、高いコストがかかる場合があります。このプロセスには、通常、何百もの異なるモデルのトレーニングに関連するグリッド

Archived

検索やランダム検索などの手法が必要です。Amazon SageMaker 自動モデルチューニング (ハイパーパラメータチューニングとも呼ばれます) は、指定したアルゴリズムとさまざまなハイパーパラメータを使用して、データセットで多くのトレーニングジョブを実行することで、モデルの最適なバージョンを見つけます。次に、選択したメトリクスで測定し、最高パフォーマンスのモデルを得たハイパーパラメータを選択します。ハイパーパラメータチューニングでは、限られた数のトレーニングジョブで最適なパラメータセットを迅速かつ効率的に決定できる ML 手法が使用されます。

また、ハイパーパラメータチューニングジョブのウォームスタートにより、チューニングプロセスを迅速化し、モデルのチューニングコストを削減できます。ハイパーパラメータチューニングジョブのウォームスタートにより、チューニングジョブをゼロから開始する必要がなくなります。 代わりに、選択した親ジョブに基づいて新しいハイパーパラメータチューニングジョブを作成できます。これにより、その親ジョブで実行されたトレーニングジョブを事前知識として再利用できるため、モデルチューニングに関連するコストが減ります。

最後に、データを自動的に分析してモデルを構築する Amazon SageMaker AutoPilot を評価し、時間とコストを節約します。Autopilot は、高性能アルゴリズムのリストから最適なアルゴリズムを選択し、それらのアルゴリズムに対してさまざまなパラメータ設定を自動的に試して最高のモデル品質を取得します。

MLCOST 04: ML 推論のコストをどのように最適化しますか?

モデルのトレーニングに基づいて、ワークロードに適したインスタンスタイプを理解することが重要です。まず、レイテンシー、スループット、コストについて検討します。ここでも、小さなものから開始し、最初にスケールアウトしてからスケールアップすることをお勧めします。

コスト削減のために ML コンピューティングインスタンスの自動スケーリングを使用することに加えて、CPU と GPU の違いを測定します。深層学習 ML モデルでは、トレーニングのために高性能な GPU インスタンスが必要ですが、深層学習モデルに対する推論では、通常は GPU の最大性能を必要としません。そのため、これらの深層学習を本格的な GPU でホストすると、GPU を十分に活用できず、不要なコストが発生する可能性があります。また、ML モデ

Archived

ルに必要な推論アーキテクチャについても考慮します。つまり、推論リクエストのバッチが必要であるときにモデルをオンデマンドでデプロイできるかどうか、またはリアルタイム予測のために 24 時間 365 日使用可能である必要があるかどうかを判断します。

AWS では、Amazon SageMaker エンドポイントで自動スケーリングがサポートされるため、リソースの需要と供給を一致させることができます。Amazon SageMaker の自動スケーリングを使用すると、モデルの伸縮性と可用性を確保できるとともに、推論エンドポイントをスケーリングするための適切なメトリクスを選択することでコストを最適化できます。

自動スケーリングでは、推論リクエストの量に合わせてエンドポイントの背後にあるインスタンス数が増減されますが、GPU のコンピューティング能力の一部をインスタンスにアタッチすることで、ホストされるインスタンスのコンピューティング能力を高めることもできます。Amazon Elastic Inference では、低コストの GPU を使用したアクセラレーションを Amazon EC2 と Amazon SageMaker インスタンスにアタッチでき、深層学習の推論実行コストを削減できます。

スタンドアロン GPU インスタンスは、数百のデータサンプルを並列処理する必要があるモデルトレーニングアクティビティに適していますが、通常、少量の GPU リソースを消費する推論にはオーバーサイズです。モデルが異なれば、必要となる GPU、CPU、メモリリソースの容量も異なります。最も要求の厳しいリソースの要件を満たす GPU インスタンスタイプを選択すると、他のリソースの使用率が低くなり、不要なコストが発生することがよくあります。

Amazon Elastic Inference では、全体的な CPU とメモリのニーズに最も適した Amazon EC2 インスタンスタイプを選択し、リソースの効率的な使用と推論実行コストの削減のために必要な推論アクセラレーションの量を個別に設定できます。

一部のアプリケーションは、オンライン/リアルタイム予測を必要とせず、定期的なバッチ予測に適しています。この場合、エンドポイントを 24 時間 365 日ホストする必要はありません。このようなアプリケーションは、Amazon SageMaker バッチ変換に適しています。バッチ変換では、リクエストごとに 1 つの推論ではなく、データセット全体の推論が生成されます。バッチ変換では、推論の実行に必要なすべてのコンピューティングリソースが管理されます。

Archived

これには、インスタンスの起動と、バッチ変換ジョブが完了した後のインスタンスの終了が含まれます。

特定のデータに対して推論を実行するには、データの前処理、後処理、またはその両方が必要になることがあります。このような処理では、最終モデルから目的の推論を生成する前に、複数の中間モデルからの推論を連結する場合があります。この場合、複数のエンドポイントに中間モデルをデプロイする代わりに、Amazon SageMaker の推論パイプラインを使用します。

推論パイプラインは、Amazon SageMaker モデルの 1 つで、データに対する推論のリクエストを処理する一連のコンテナで構成されます。これらの推論パイプラインは完全マネージド型で、前処理、予測、後処理を組み合わせることができます。パイプラインモデルの呼び出しは、一連の HTTPS リクエストとして処理されます。関連するすべてのステップを同じエンドポイントにデプロイすることで、コストを削減し、推論のレイテンシーを減らすことができます。

本番環境に複数のモデルがあり、それぞれ別々のエンドポイントにデプロイされている場合、推論コストはモデルの数に比例して増加します。ただし、共有サービスコンテナを介して提供できる類似したモデルが多数あり、すべてのモデルに同時にアクセスする必要がない場合は、Amazon SageMaker のマルチモデルエンドポイント機能を使用します。これにより、複数のトレーニング済みモデルをエンドポイントにデプロイし、単一のコンテナで提供できます。予測リクエストでターゲットモデル名をパラメータとして指定することで、特定のモデルを簡単に呼び出すことができます。アクセス頻度の低い ML モデルが多数ある場合は、1 つのマルチモデルエンドポイントを使用すると、推論トラフィックを効率的に処理し、コストを大幅に削減できます。

需要と供給の一致

ML ワークロードに適用されるコスト最適化の「需要と供給の一致」分野のベストプラクティについては、「AWS Well-Architected フレームワーク」ホワイトペーパーを参照してください。

Archived

費用認識

ML ワークロードに適用されるコスト最適化の「費用認識」分野のベストプラクティについては、「AWS Well-Architected フレームワーク」ホワイトペーパーを参照してください。

時間経過を伴う最適化

ML ワークロードに適用されるコスト最適化の「時間経過を伴う最適化」分野のベストプラクティについては、「AWS Well-Architected フレームワーク」ホワイトペーパーを参照してください。

リソース

コスト最適化に関するベストプラクティスの詳細については、以下のリソースを参照してください。

ドキュメントとブログ

- Amazon SageMaker Pricing
- Use the Amazon SageMaker local mode to train on your notebook instance
- Making the most of your Machine Learning budget on Amazon SageMaker
- Lowering total cost of ownership for machine learning and increasing productivity with Amazon SageMaker

まとめ

機械学習は、組織が自動化、効率性、イノベーションを実現するためのまたとない機会をもたらします。このホワイトペーパーでは、ML のレンズを通じて Well-Architected フレームワークの 5 本の柱を再検討し、AWS クラウドで信頼性、安全性、効率性、費用対効果の高い ML ワークロードを構築および運用するためのアーキテクチャ上のベストプラクティスを取り上げています。AI サービス、マネージド ML サービス、ML フレームワークを使用する状況でベストプラクティスを説明し、ビジネス目標に最適なオプションの優れた使用方法を提供します。一

Archived

連の質問で構成されるこれらのベストプラクティスを使用して、既存の ML ワークロードまたは提案された ML ワークロードを確認します。

ML ワークロードを運用する際に、部門横断型チームが関与し、エンドツーエンドのパイプラインの自動化を導入するようにします。信頼性を高めるため、自動スケーリングの自己修復機能を活用し、バージョニングを通じてすべてのアーティファクトを追跡することでシステム障害に確実に対応し、稼働中のシステムを自動的に再構築できるようにします。

AWS での ML ワークロードは、各種の ML アーティファクトにアクセスできるユーザーとモノを厳格に制御する認証および認可コントロールを使用して保護される必要があります。安全なアプリケーションは、組織の機密情報資産を保護し、すべてのレイヤーでコンプライアンス要件を満たします。パフォーマンスに優れた ML ワークロードを実現するため、さまざまな ML 要件に合わせて最適化された複数の優れたインスタンスタイプが用意されています。データ駆動型のアプローチを採用し、利用可能なさまざまな組み合わせの CPU、GPU、FPGA、メモリ、ストレージ、ネットワーク容量を考慮し、CPU インスタンスまたは GPU インスタンスのいずれかを選択します。コストを最適化するため、「使用した分のみ支払う」を活用し、データ、トレーニング、推論の需要に応じてリソースをサイズ設定することで不必要な消費を減らします。

機械学習を取り巻く状況は、ツールのエコシステムやプロセスの成長および成熟と共に進化し続けています。このような進化が起こったときに、お客様が ML アプリケーションを適切に設計できるように、このホワイトペーパーを継続的に更新していきます。

Archived

寄稿者

本ドキュメントの寄稿者は以下のとおりです。

- Sireesha Muppala、AI/ML スペシャリスト SA、アマゾン ウェブ サービス
- Shelbee Eigenbrode、AI/ML ソリューションアーキテクト、アマゾン ウェブ サービス
- Christian Williams、機械学習スペシャリスト SA、アマゾン ウェブ サービス

- Bardia Nikpourian、スペシャリスト TAM (AI/ML)、アマゾン ウェブ サービス
- Ryan King, Sr.TPM、AWS マネージドクラウド、アマゾン ウェブ サービス

その他の資料

詳細については、以下を参照してください。

- Managing Machine Learning Projects (AWS ホワイトペーパー)

改訂履歴

日付	説明
2020 年 4 月	初版発行

Archived