Lente para Machine Learning

AWS Well-Architected Framework

Lente para Machine Learning: AWS Well-Architected Framework

Table of Contents

Lente para Machine Learning – AWS Well-Architected Framework

Data de publicação: abril de 2020 (Revisões do documento (p. 59))

Resumo

Este documento descreve a lente para Machine Learning do AWS Well-Architected Framework. O documento inclui cenários comuns de machine learning (ML) e identifica os principais elementos para garantir que as cargas de trabalho sejam projetadas de acordo com as melhores práticas.

Introdução

O AWS Well-Architected Framework ajuda a entender os prós e os contras das decisões que você toma ao criar sistemas na AWS. Ao usar o Framework, você poderá aprender as melhores práticas de arquitetura para projetar e operar sistemas confiáveis, seguros, eficientes e econômicos na nuvem. Ele fornece uma maneira de você avaliar consistentemente suas arquiteturas em relação às melhores práticas e identificar áreas de melhoria. Acreditamos que ter sistemas bem projetados aumenta significativamente a probabilidade de êxito dos negócios.

Sob a Lente para Machine Learning, nos concentramos em como projetar, implantar e arquitetar suas cargas de trabalho de machine learning na Nuvem AWS. Essa lente é somada às melhores práticas incluídas no Well-Architected Framework. Para resumir, incluímos apenas os detalhes dessa lente que são específicos a cargas de trabalho de machine learning (ML). Ao projetar cargas de trabalho de ML, você deve usar as melhores práticas e perguntas aplicáveis do whitepaper sobre o AWS Well-Architected Framework.

Esta lente é destinada a pessoas que ocupam cargos de tecnologia, como Chief Technology Officer (CTO – Diretor de tecnologia), arquitetos, desenvolvedores e membros da equipe de operações. Depois de ler este artigo, você compreenderá as melhores práticas e estratégias a serem usadas ao projetar e operar cargas de trabalho de ML na AWS.

Definições

A lente para Machine Learning é baseada em cinco pilares: excelência operacional, segurança, confiabilidade, eficiência de performance e otimização de custos. A AWS fornece vários componentes básicos para cargas de trabalho de ML que permitem projetar arquiteturas robustas para suas aplicações de ML.

Ao criar uma carga de trabalho de machine learning, você deve avaliar duas áreas.

Tópicos

Pilha de machine learning

Ao criar uma carga de trabalho baseada em ML na AWS, você pode escolher entre diferentes níveis de abstração para equilibrar a velocidade de entrada no mercado com o nível de personalização e de habilidade em ML:

- Serviços de inteligência artificial (IA)
- Serviços de ML
- Estruturas de trabalho e infraestrutura de ML

Serviços de inteligência artificial (IA)

O nível Serviços de IA fornece serviços totalmente gerenciados que permitem adicionar capacidades de ML rapidamente às suas cargas de trabalho usando chamadas de API. Com isso, você pode criar aplicações poderosas e inteligentes com funcionalidades como visão computacional, fala, linguagem natural, chatbots, previsões e recomendações. Nesse nível, os serviços baseiam-se em modelos de machine learning e aprendizado profundo pré-treinados ou treinados automaticamente, ou seja, você não precisa de conhecimento em ML para usá-los.

A AWS oferece muitos serviços de IA que podem ser integrados às suas aplicações por meio de chamadas de API. Por exemplo, você pode usar o Amazon Translate para traduzir ou localizar conteúdo de texto, o Amazon Polly para conversão de texto em fala e o Amazon Lex para criar chatbots de conversação.

Serviços de ML

O nível Serviços de ML fornece serviços e recursos gerenciados de machine learning para desenvolvedores, cientistas de dados e pesquisadores. Esses tipos de serviços permitem rotular dados, criar, treinar, implantar e operar modelos personalizados de ML sem precisar se preocupar com as necessidades subjacentes de infraestrutura. O trabalho pesado genérico de gerenciamento de infraestrutura é gerenciado pelo fornecedor de nuvem, permitindo que suas equipes de ciência de dados concentrem-se no que fazem melhor.

Na AWS, o Amazon SageMaker permite que desenvolvedores e cientistas de dados criem, treinem e implantem modelos de ML com rapidez e facilidade em qualquer escala. Por exemplo, o Amazon SageMaker Ground Truth ajuda você a criar rapidamente conjuntos de dados de treinamento de ML altamente precisos, enquanto o Amazon SageMaker Neo permite que os desenvolvedores treinem modelos de ML uma vez e os executem em qualquer lugar na nuvem ou na borda.

Estruturas de trabalho e infraestrutura de ML

O nível Estruturas de trabalho e infraestrutura de ML é destinado a profissionais especializados em machine learning. Trata-se de pessoas acostumadas a projetar as próprias ferramentas e fluxos de trabalho para criar, treinar, ajustar e implantar modelos, e que estão acostumadas a trabalhar diretamente com estrutura de trabalho e infraestrutura.

Na AWS, você pode usar estruturas de trabalho de ML de código aberto, como TensorFlow, PyTorch e Apache MXNet. Nesse nível, a Amazon Machine Image (AMI – Imagem de máquina da Amazon) e os contêineres do Deep Learning têm várias estruturas de trabalho de ML pré-instaladas que são otimizadas para performance. Essa otimização significa que eles sempre estão prontos para execução na poderosa infraestrutura de computação otimizada para ML, como instâncias P3 e P3dn do Amazon EC2, que proporcionam um aumento de velocidade e eficiência para cargas de trabalho de machine learning.

Combinação de níveis

Frequentemente as cargas de trabalho usam serviços de vários níveis da pilha de ML. Dependendo do caso de uso empresarial, é possível combinar os serviços e a infraestrutura de diferentes níveis para atender a vários requisitos e atingir vários objetivos empresariais. Por exemplo, você pode usar serviços de IA para análise de sentimentos das avaliações de clientes em seu site de vendas e usar serviços gerenciados de ML para criar um modelo personalizado usando seus próprios dados para prever vendas futuras.

Fases das cargas de trabalho de ML

A criação e a operação de uma carga de trabalho comum de ML é um processo iterativo e consiste em várias fases. Identificamos essas fases de maneira ampla com base no modelo de processo de padrão aberto para Cross Industry Standard Process Data Mining (CRISP-DM – Processo padrão intersetorial para mineração de dados) como uma diretriz geral. O CRISP-DM é usado como base por ser uma ferramenta comprovada no setor e ter neutralidade em relação às aplicações, fazendo dela uma metodologia fácil de aplicar em uma grande variedade de cargas de trabalho e pipelines de ML.

O processo completo de machine learning inclui as seguintes fases:

Figura 1 – Processo completo de machine learning

Tópicos

- Avaliação do modelo e avaliação comercial (p. 11)

Identificação da meta empresarial

A Identificação da meta empresarial é a fase mais importante. Uma organização que está avaliando a adoção do ML deve ter uma ideia clara do problema a ser resolvido e do valor empresarial a ser obtido com a solução desse problema usando ML. Você precisa ser capaz de mensurar o valor empresarial em comparação com objetivos empresariais e critérios de sucesso específicos. Embora isso seja válido para qualquer solução técnica, essa etapa é especialmente desafiadora ao considerar soluções de ML porque o ML é uma tecnologia disruptiva.

Após determinar seus critérios de sucesso, avalie de maneira realista a capacidade de sua organização de executar esse objetivo. A meta deve ser tangível e fornecer um caminho claro até a produção.

Será desejável validar que o ML é a abordagem adequada para atingir sua meta empresarial. Avalie todas as opções disponíveis para concretizar a meta, qual a precisão prevista para os resultados, além do custo e da escalabilidade de cada abordagem ao decidir entre uma delas.

Para que uma abordagem baseada em ML tenha êxito, é essencial ter abundância de dados relevantes e de alta qualidade que sejam aplicáveis ao algoritmo que você está tentando treinar. Avalie cuidadosamente a disponibilidade dos dados para garantir que as fontes de dados corretas estejam disponíveis e sejam acessíveis. Por exemplo, você precisa de dados de treinamento para treinar e comparar seu modelo de ML, mas também precisa de dados dos negócios para avaliar o valor de uma solução de ML.

Aplique estas melhores práticas:

- Entender os requisitos comerciais
- Formular uma questão empresarial
- Determinar a viabilidade de ML e os requisitos de dados de um projeto
- Avaliar o custo de aquisição de dados, treinamento, inferência e previsões incorretas
- Analisar trabalhos comprovados ou publicados em domínios semelhantes, se disponíveis
- Determinar as principais métricas de performance, incluindo os erros aceitáveis
- Definir a tarefa de machine learning com base na questão empresarial
- Identificar recursos críticos e indispensáveis

Enquadramento do problema de ML

Nesta fase, o problema empresarial é enquadrado como um problema de machine learning: o que é observado e o que deve ser previsto (conhecido como rótulo ou variável de destino). Determinar o que deve ser previsto e como as métricas de performance e erro precisam ser otimizadas é uma etapa essencial em ML.

Por exemplo, imagine um cenário no qual uma fábrica quer identificar quais produtos maximizarão os lucros. A concretização dessa meta empresarial depende parcialmente da determinação do número correto de produtos que serão produzidos. Nesse cenário, o objetivo é prever as vendas futuras do produto com base em vendas passadas e atuais. A previsão de vendas futuras passa a ser o problema a ser resolvido e o uso de ML é uma abordagem que pode ser empregada para solucioná-lo.

Aplique estas melhores práticas:

- Definir os critérios para um resultado bem-sucedido do projeto
- Estabelecer uma métrica de performance observável e quantificável para o projeto, como precisão, latência da previsão ou minimização do valor em estoque

- Formular a questão de ML em termos de entradas, resultados desejados e a métrica de performance a ser otimizada
- Avaliar se o ML é uma abordagem viável e adequada
- Criar um objetivo de aquisição de dados e de anotação de dados, bem como uma estratégia para concretizá-lo
- Começar com um modelo simples e fácil de interpretar que torne a depuração mais gerenciável

Coleta de dados

Em cargas de trabalho de ML, os dados (entradas e a saída desejada correspondente) atendem a três funções importantes:

- Definição da meta do sistema: a representação da saída e o relacionamento entre cada saída e cada entrada, por meio de pares de entrada/saída
- Treinamento do algoritmo que associará as entradas às saídas
- Medição da performance do modelo treinado e avaliação da concretização ou não do alvo de performance

A primeira etapa é identificar quais são os dados necessários para seu modelo de ML e avaliar os vários meios disponíveis para a coleta desses dados para treinar seu modelo.

Conforme as organizações coletam e analisam quantidades cada vez maiores de dados, as soluções tradicionais on-premise para armazenamento, gerenciamento e análise de dados não conseguem mais acompanhar o ritmo. Um data lake baseado na nuvem é um repositório centralizado que permite armazenar todos os seus dados estruturados e não estruturados, independentemente da escala. É possível armazenar os dados no estado em que se encontram, sem precisar estruturá-los antes, e executar diversos tipos de análise (desde painéis e visualizações a processamento de big data, análise em tempo real e ML) para orientá-lo a tomar decisões melhores.

A AWS disponibiliza várias maneiras de ingerir dados em massa provenientes de recursos estáticos ou de fontes novas geradas dinamicamente, como sites, aplicativos para dispositivos móveis e dispositivos conectados à Internet. Por exemplo, é possível criar um data lake altamente escalável usando o Amazon Simple Storage Service (Amazon S3). Você pode usar o AWS Lake Formation para configurar seu data lake com facilidade.

Para ingerir dados, você pode usar o AWS Direct Connect para conectar seu datacenter diretamente a uma região da AWS. É possível transferir fisicamente petabytes de dados em lotes usando o AWS Snowball ou, se você tiver exabytes de dados, usando o AWS Snowmobile. É possível integrar o armazenamento on-premise existente usando o AWS Storage Gateway ou adicionar funcionalidades de nuvem usando o AWS Snowball Edge. Você também pode usar o Amazon Kinesis Data Firehose para coletar e ingerir várias fontes de dados de streaming.

Aplique estas melhores práticas:

- Detalhar as várias fontes e etapas necessárias para extrair dados
- Confirmar a disponibilidade dos dados, tanto em termos de quantidade quanto de qualidade
- Entender seus dados de maneira abrangente antes de prepará-los para consumo em etapas posteriores
- Definir a governança de dados: quem é o proprietário dos dados, quem tem acesso, o uso adequado dos dados e a capacidade de acessar e excluir partes específicas dos dados sob demanda
- Rastrear a linhagem dos dados, de modo que a localização e a fonte de dados sejam monitorados e conhecidos durante o processamento posterior
- Usar os serviços gerenciados da AWS para a coleta e a integração de dados
- Usar uma abordagem para armazenar seus dados, como um data lake

Preparação de dados

Os modelos de ML são tão bons quanto os dados usados para treiná-los. Depois que os dados são coletados, é crucial integrar, anotar, preparar e processar esses dados. Uma característica essencial dos dados de treinamento adequados é que sejam fornecidos de uma maneira otimizada para aprendizagem e generalização. A preparação dos dados deve começar com uma amostra pequena e estatisticamente válida, e ser aprimorada iterativamente com diferentes estratégias de preparação de dados, mantendo a integridade dos dados durante esse processo.

A AWS fornece vários serviços que você pode usar para anotar seus dados e aplicar o processo Extract, Transform, and Load (ETL – Extrair, transformar e carregar) neles em grande escala.

O Amazon SageMaker é um produto totalmente gerenciado que envolve todo o fluxo de trabalho de ML para rotular e preparar seus dados, escolher um algoritmo, treiná-lo, ajustá-lo e otimizá-lo para implantação e para fazer previsões.

O Amazon SageMaker Ground Truth oferece fácil acesso a rotuladores humanos públicos e privados, e fornece fluxos de trabalho e interfaces de usuário integrados para tarefas comuns de rotulagem. Ele usa um modelo de machine learning para rotular automaticamente os dados brutos a fim de produzir rapidamente conjuntos de dados de treinamento com alta qualidade a uma fração do custo da rotulagem manual. Os dados são encaminhados para humanos apenas se o modelo de aprendizagem ativo não conseguir rotulá-los de maneira confiável. O produto fornece fluxos de trabalho personalizados dinâmicos, encadeamento e rastreamento de trabalhos para economizar tempo em trabalhos subsequentes de rotulagem de ML usando a saída de trabalhos de rotulagem anteriores como entrada para os novos.

O AWS Glue é um serviço totalmente gerenciado de Extract, Transform, and Load (ETL – Extrair, transformar e carregar) que pode ser usado para automatizar o pipeline de ETL. O AWS Glue descobre seus dados e traça o perfil deles automaticamente com o Glue Data Catalog, recomenda e gera código de ETL para transformar seus dados de origem em esquemas de destino, e executa os trabalhos de ETL em um ambiente Apache Spark totalmente gerenciado e com dimensionamento horizontal para carregar seus dados nos respectivos destinos. Ele também permite configurar, orquestrar e monitorar fluxos de dados complexos.

O Amazon EMR fornece uma estrutura de trabalho gerenciada do Hadoop que facilita e acelera o processamento de quantidades enormes de dados em instâncias dinamicamente escaláveis do Amazon EC2. Também é possível executar outras estruturas de trabalho distribuídas conhecidas no EMR, como Apache Spark, HBase, Presto e Flink, e interagir com dados em outros datastores da AWS, como Amazon S3 e Amazon DynamoDB.

A preparação de dados aplica-se não somente aos dados de treinamento usados para criar um modelo de machine learning, mas também aos novos dados empresariais que são usados para fazer inferências em relação ao modelo depois que ele é implantado. Normalmente, a mesma sequência de etapas de processamento de dados que você aplica aos dados de treinamento também é aplicada às solicitações de inferência.

O Amazon SageMaker Inference Pipeline implanta pipelines para que você possa transmitir dados brutos de entrada e executar pré-processamento, previsões e pós-processamento em solicitações de inferência em tempo real e em lote. Os pipelines de inferência permitem reutilizar a funcionalidade existente de processamento de dados.

Aplique estas melhores práticas:

- Começar com um conjunto pequeno e estatisticamente válido de dados de amostra para a preparação de dados
- Experimentar iterativamente com diferentes estratégias de preparação de dados
- Implementar um ciclo de comentário durante o processo de limpeza de dados que forneça alertas relacionados a anomalias durante todas as etapas de preparação de dados

- Aplicar a integridade de dados continuamente
- Aproveitar os serviços gerenciados de ETL

Análise e visualização de dados

Um aspecto essencial para entender seus dados é a identificação de padrões. Muitas vezes esses padrões não ficam evidentes quando você está apenas olhando para os dados em tabelas. A ferramenta de visualização correta pode ajudá-lo a ter um entendimento mais profundo dos dados. Antes de criar um gráfico ou quadro, é necessário decidir o que você quer mostrar. Por exemplo, gráficos podem reunir informações como Key Performance Indicadores (KPIs – Indicadores-chave de performance), relacionamentos, comparações, distribuições ou composições.

A AWS fornece vários serviços que você pode usar para visualizar e analisar dados em grande escala.

O Amazon SageMaker fornece um ambiente de bloco de anotações Jupyter hospedado que você pode usar para visualizar e analisar dados. O Project Jupyter é uma aplicação Web de código aberto que permite criar visualizações e texto narrativo, bem como executar limpeza de dados, transformação de dados, simulação numérica, modelagem estatística e visualização de dados.

O Amazon Athena é um serviço de consulta totalmente gerenciado que você pode usar para consultar dados no Amazon S3 usando operadores e funções ANSI SQL. O Amazon Athena usa a tecnologia sem servidor e pode ser dimensionado de maneira contínua para atender às suas demandas de consultas.

O Amazon Kinesis Data Analytics fornece capacidades analíticas em tempo real ao analisar os dados de streaming para obter insights práticos. O produto é automaticamente dimensionado para corresponder ao volume e à taxa de transferência de seus dados recebidos.

O Amazon QuickSight é um produto de Business Intelligence (BI – Inteligência de negócios) viabilizado pela nuvem que fornece painéis e visualizações. O produto é automaticamente dimensionado para dar suporte a centenas de usuários e oferece compartilhamento e colaboração seguros para a criação do storyboard. Além disso, o produto tem recursos integrados de ML que fornecem detecção de anomalias, previsão e análise de cenários hipotéticos prontos para uso.

Aplique estas melhores práticas:

- Criar um perfil de seus dados (visualização categórica vs. ordinal vs. quantitativa)
- Escolher a ferramenta ou combinação de ferramentas correta para seu caso de uso (como tamanho dos dados, complexidade dos dados e em tempo real vs. em lote)
- Monitorar o pipeline de análise de dados
- Validar suposições a respeito dos dados

Engenharia de recursos

Após explorar e entender os dados por meio de visualizações e análise, é hora da engenharia de recursos. Cada atributo exclusivo dos dados é considerado um recurso. Por exemplo, ao projetar uma solução para um problema de previsão de variação de um cliente, você começa com os dados do cliente que foram coletados no decorrer do tempo. Os dados do cliente capturam recursos (também conhecidos como atributos) como localização, idade, nível de renda e compras recentes do cliente.

A engenharia de recursos é um processo para selecionar e transformar variáveis durante a criação de um modelo preditivo usando machine learning ou modelagem estatística. Normalmente, a engenharia de recursos inclui a criação, transformação, extração e seleção de recursos.

- A criação de recursos identifica os recursos no conjunto de dados que são relevantes para o problema em questão.

- A transformação de recursos gerencia a substituição de recursos ausentes ou que são inválidos. Algumas técnicas incluem a formação de produtos cartesianos de recursos, transformações não lineares (como a separação de variáveis numéricas em categorias) e a criação de recursos específicos de domínios.
- A extração de recursos é o processo de criar novos recursos a partir de outros existentes, normalmente com a meta de reduzir a dimensionalidade dos recursos.
- A seleção de recursos é a filtragem de recursos irrelevantes ou redundantes do seu conjunto de dados. Normalmente isso é feito mediante a observação dos limites de variância ou correlação para determinar quais recursos devem ser removidos.

O Amazon SageMaker fornece um ambiente de bloco de anotações Jupyter com pré-processadores Spark e scikit-learn que você pode usar para projetar recursos e transformar os dados. No Amazon SageMaker, você também pode executar trabalhos de extração e transformação de recursos usando serviços de ETL, como o AWS Glue ou o Amazon EMR. Além disso, é possível usar o Amazon SageMaker Inference Pipeline para reutilizar a funcionalidade existente de processamento de dados.

O Amazon SageMaker Processing permite executar trabalhos de análise para engenharia de recursos (e avaliação do modelo) em grande escala em um ambiente totalmente gerenciado com todos os recursos de segurança e conformidade fornecidos pelo Amazon SageMaker. Com o Amazon SageMaker Processing, você tem a flexibilidade de usar os contêineres integrados de processamento de dados ou trazer seus próprios contêineres e enviar trabalhos personalizados para executar na infraestrutura gerenciada. Após o envio, o Amazon SageMaker inicia as instâncias de computação, processa e analisa os dados de entrada, e libera os recursos mediante a conclusão do processo.

Aplique estas melhores práticas:

- Usar especialistas em domínios para ajudar a avaliar a viabilidade e a importância dos recursos
- Remover recursos redundantes e irrelevantes (a fim de reduzir o ruído nos dados e reduzir as correlações)
- Começar com recursos que generalizam entre os contextos
- Iterar conforme cria seu modelo (novos recursos, combinações de recursos e novos objetivos de ajuste)

Treinamento de modelos

Nesta fase, você seleciona um algoritmo de machine learning que seja adequado ao problema e treina o modelo de ML. Como parte desse treinamento, você fornece ao algoritmo os dados de treinamento usados para a aprendizagem e define os parâmetros do modelo para otimizar o processo de treinamento.

Normalmente, um algoritmo de treinamento calcula várias métricas, como erro no treinamento e precisão da previsão. Essas métricas ajudam a determinar se o modelo está aprendendo bem e fará uma boa generalização para fazer previsões com base em dados inéditos. As métricas relatadas pelo algoritmo dependem do problema empresarial e da técnica de ML empregada. Por exemplo, é possível medir um algoritmo de classificação usando uma matriz de confusão que captura casos de positivos verdadeiros ou falsos positivos e negativos verdadeiros ou falsos negativos, enquanto é possível medir um algoritmo de regressão empregando a metodologia Root Mean Square Error (RMSE – Raiz do erro quadrático médio).

As configurações que podem ser ajustadas para controlar o comportamento do algoritmo de ML e a arquitetura resultante do modelo são conhecidos como hiperparâmetros. O número e o tipo de hiperparâmetros nos algoritmos de ML são específicos de cada modelo. Alguns exemplos de hiperparâmetros comumente usados são: Taxa de aprendizagem, Número de épocas, Camadas ocultas, Unidades ocultas e Funções de ativação. O ajuste (ou otimização) de hiperparâmetros é o processo de escolha da arquitetura ideal para o modelo.

O Amazon SageMaker fornece vários algoritmos populares integrados que podem ser treinados com os dados de treinamento que você preparou e armazenou no Amazon S3. Também é possível trazer seus

próprios algoritmos personalizados para treinamento no Amazon SageMaker. O algoritmo personalizado deve ser conteinerizado usando o Amazon ECS e o Amazon ECR.

Após selecionar o algoritmo, é possível iniciar o treinamento no Amazon SageMaker com uma chamada de API. Você pode optar por treinar em uma única instância ou em um cluster distribuído de instâncias. O gerenciamento de infraestrutura necessário para o processo de treinamento é gerenciado pelo Amazon SageMaker, que elimina o fardo das tarefas pesadas genéricas.

O Amazon SageMaker também permite o ajuste automático do modelo por meio de trabalhos de ajuste de hiperparâmetros. Depois de configurado, um trabalho de ajuste de hiperparâmetros encontra a melhor versão do modelo executando vários trabalhos de treinamento em seu conjunto de dados. Para isso, ele usa o algoritmo e os intervalos de hiperparâmetros especificados por você. Em seguida, o trabalho seleciona os valores de hiperparâmetros que resultam no modelo com melhor performance de acordo com a métrica selecionada. É possível usar o ajuste automático de modelo do Amazon SageMaker com algoritmos integrados, algoritmos personalizados e contêineres predefinidos do Amazon SageMaker para estruturas de trabalho de ML.

O Amazon SageMaker Debugger fornece visibilidade no processo de treinamento de ML ao monitorar, gravar e analisar dados que capturam o estado de um trabalho de treinamento em intervalos periódicos. Ele também oferece a capacidade de executar exploração interativa dos dados capturados durante o treinamento e uma funcionalidade de emissão de alertas para erros detectados durante o treinamento. Por exemplo, ele pode detectar automaticamente e emitir alertas para erros que ocorrem habitualmente, como o aumento ou diminuição excessiva de valores de gradiente.

O Amazon SageMaker Autopilot simplifica o processo de treinamento de ML ao gerenciar automaticamente o pré-processamento de dados, a seleção de algoritmos e o ajuste de hiperparâmetros. Ele permite criar modelos de classificação e regressão simplesmente fornecendo os dados de treinamento em formato tabular. Esse recurso explora várias soluções de ML com diferentes combinações de pré-processadores de dados, algoritmos e configurações de parâmetros de algoritmos para encontrar o modelo mais preciso. O Amazon SageMaker Autopilot seleciona o melhor algoritmo da lista de algoritmos de alta performance que contam com suporte nativo. Ele também testa automaticamente diferentes configurações de parâmetros nesses algoritmos para obter a melhor qualidade do modelo. Em seguida, você pode implantar o melhor modelo na produção ou avaliar vários candidatos para ponderar métricas como precisão, latência e tamanho do modelo.

O AWS Deep Learning AMI e o AWS Deep Learning Containers permitem usar várias estruturas de trabalho de ML de código aberto para treinamento em sua infraestrutura. As AMIs do Amazon Deep Learning vêm com estruturas de trabalho e interfaces populares de aprendizado profundo pré-instaladas, como TensorFlow, PyTorch, Apache MXNet, Chainer, Gluon, Horovod e Keras. A AMI ou o contêiner podem ser executados em uma infraestrutura poderosa otimizada para a performance de ML.

O Amazon EMR tem funcionalidades de cluster distribuído e também é uma opção para executar trabalhos de treinamento em dados que estejam armazenados localmente no cluster ou no Amazon S3.

Aplique estas melhores práticas:

- Gerar um plano de teste de modelos antes de treinar seu modelo
- Ter um entendimento claro do tipo de algoritmo que você precisa treinar
- Garantir que os dados de treinamento representem bem seu problema empresarial
- Usar serviços gerenciados para suas implantações de treinamento
- Aplicar estratégias de treinamento incremental ou de aprendizagem por transferência
- Para evitar sobreajuste e reduzir o custo, interromper trabalhos de treinamento precocemente quando os resultados não apresentarem melhora significativa conforme medidos pela métrica objetiva
- Monitorar atentamente as métricas de treinamento, porque a performance do modelo pode piorar com o tempo
- Aproveitar serviços gerenciados para o ajuste automático de modelos

Avaliação do modelo e avaliação comercial

Após o treinamento do modelo, avalie-o para determinar se a performance e a precisão permitirão alcançar suas metas de negócios. Uma alternativa interessante é gerar vários modelos usando diferentes métodos e avaliar a eficácia de cada modelo. Por exemplo, você pode aplicar diferentes regras empresariais para cada modelo e, em seguida, aplicar várias medições para determinar a adequação de cada um deles. Também é possível avaliar se o modelo precisa ser mais sensível do que específico, ou mais específico do que sensível. Para modelos multiclasse, avalie as taxas de erros de cada classe separadamente.

Você pode avaliar seu modelo usando dados históricos (avaliação offline) ou dados dinâmicos (avaliação online). Na avaliação offline, o modelo treinado é avaliado com uma parte do conjunto de dados que foi separada como um conjunto de controle. Esses dados de controle nunca são usados para o treinamento ou a validação do modelo. São usados apenas para avaliar erros no modelo final. As anotações de dados em espera precisam ter alta precisão para que a avaliação faça sentido. Aloque recursos adicionais par verificar a precisão dos dados de controle.

Os serviços da AWS usados para o treinamento de modelos também têm uma função nessa fase. É possível realizar a validação de modelos usando o Amazon SageMaker, AMIs do Amazon Deep Learning ou o Amazon EMR.

Com base nos resultados da avaliação, é possível fazer o ajuste fino dos dados, do algoritmo ou de ambos. Ao fazer um ajuste fino nos dados, você aplica os conceitos de limpeza de dados, preparação e engenharia de recursos.

Aplique estas melhores práticas:

- Ter um entendimento claro de como medir seu sucesso
- Avaliar as métricas de modelo em relação às expectativas empresariais referentes ao projeto
- Planejar e executar a implantação na produção (implantação do modelo e inferência do modelo)

Depois que um modelo for treinado, ajustado e testado, é possível implantá-lo na produção e fazer inferências (previsões) em relação ao próprio modelo.

O Amazon SageMaker oferece uma ampla variedade de opções para implantação e inferência, e é o produto da AWS recomendado para hospedar seus modelos de ML de produção.

Assim como ocorre com o treinamento de modelos, é possível hospedar modelos no Amazon SageMaker usando uma chamada de API. Você pode optar por hospedar seu modelo em uma única instância ou em várias instâncias. A mesma API permite configurar o dimensionamento automático, de modo que você possa atender às variadas demandas de inferência em seu modelo de ML. O gerenciamento de infraestrutura necessário para hospedar seus modelos é completamente gerenciado pelo Amazon SageMaker, eliminando a sobrecarga das tarefas pesadas genéricas.

O Amazon SageMaker Inference Pipelines permite que você implante pipelines de inferência para que possa transmitir dados brutos de entrada e executar pré-processamento, previsões e concluir o pós-processamento em solicitações de inferência em tempo real e em lote. Os pipelines de inferência podem ser compostos por qualquer estrutura de trabalho de ML, algoritmo integrado ou contêineres personalizados que você possa usar no Amazon SageMaker. É possível criar pipelines de processamento de dados de recursos e de engenharia de recursos com um pacote de transformadores de recursos disponíveis nos contêineres das estruturas de trabalho SparkML e Scikit-learn. Além disso, é possível implantá-los como parte dos pipelines de inferência para reutilizar o código de processamento de dados e simplificar o gerenciamento de seus processos de ML. Esses pipelines de inferência são totalmente gerenciados e podem combinar pré-processamento, previsões e pós-processamento como parte de um processo de ciência de dados.

O Amazon SageMaker Model Monitor monitora continuamente os modelos de ML em produção. Depois que um modelo de ML for implantado na produção, os dados do mundo real podem começar a se

diferenciar dos dados que foram usados para treinar o modelo, resultando em desvios na qualidade do modelo e, consequentemente, em modelos menos precisos. O Model Monitor detecta desvios, como desvios de dados, capazes de degradar a performance do modelo no decorrer do tempo, e emite alertas para que você tome as medidas corretivas.

O Amazon SageMaker Neo permite treinar os modelos de ML uma vez e então executá-los em qualquer lugar na nuvem e na borda. O Amazon SageMaker Neo consiste em um compilador e um tempo de execução. A API de compilação lê os modelos exportados de várias estruturas de trabalho, os converte em representações independentes da estrutura de trabalho e gera o código binário otimizado. Em seguida, o tempo de execução de cada plataforma de destino é carregado e executa o modelo compilado.

O Amazon Elastic Inference permite vincular aceleração de baixo custo habilitada por GPU a instâncias do Amazon EC2 e do Amazon SageMaker para reduzir o custo de execução de inferências de aprendizado profundo. As instâncias de GPU independentes são projetadas para treinamento de modelo e geralmente são superdimensionadas para inferência. Mesmo que os trabalhos de treinamento processem em lote centenas de amostras de dados em paralelo, a maioria das inferências ocorre em uma única entrada em tempo real e consome apenas uma pequena quantidade de computação da GPU. O Amazon Elastic Inference soluciona esse problema ao permitir que você vincule a quantidade adequada de aceleração de inferência habilitada por GPU a qualquer tipo de instância do Amazon EC2 ou do Amazon SageMaker, sem alterações de código.

Embora o Elastic Inference tenha suporte nativo para algumas estruturas de trabalho de aprendizado profundo, como TensorFlow e Apache MXNet, também é possível usá-lo com outras estruturas de trabalho de aprendizado profundo usando o Open Neural Network Exchange (ONNX) para exportar seu modelo e importá-lo no MXNet.

Aplique estas melhores práticas:

- Monitorar a performance do modelo na produção e comparar às expectativas de negócios
- Monitorar as diferenças entre a performance do modelo durante o treinamento e na produção
- Retreinar o modelo quando detectar alterações na performance dele. Por exemplo, as expectativas de vendas e as previsões posteriores podem mudar devido a novos concorrentes
- Usar transformação em lote como alternativa para hospedar serviços com a finalidade de obter inferências em conjuntos de dados inteiros
- Aproveitar as variantes da produção para testar variações de um novo modelo com testes A/B

Princípios gerais de design

O Well-Architected Framework identifica um conjunto de princípios gerais de design a fim de facilitar um bom design na nuvem para cargas de trabalho de machine learning:

- Possibilite agilidade por meio da disponibilidade de conjuntos de dados de alta qualidade

  As cargas de trabalho de ciência de dados exigem acesso a dados ativos ou dados em lote para todas as fases em um pipeline de entrega. Implemente mecanismos para viabilizar o acesso aos dados com validação de dados e controles de qualidade.
- Comece de forma simples e avance por meio de experimentos

  Ao começar com um pequeno conjunto de recursos, você pode evitar o erro de começar com um modelo complexo e perder o controle do impacto dos recursos. Escolha um modelo simples e execute uma série de experimentos durante todo o processo.
- Desacople o treinamento e a avaliação de modelos da hospedagem de modelos

  Selecione os recursos que têm o melhor alinhamento com fases específicas no ciclo de vida da ciência de dados, separando os recursos de treinamento de modelo, avaliação de modelo e hospedagem de modelos.
- Detecte desvio de dados

  Para gerenciar o desvio de dados no decorrer do tempo, meça continuamente a precisão da inferência depois que o modelo estiver em produção. Geralmente os dados usados no ML vêm de várias fontes, e o formato e o significado desses dados podem mudar conforme os sistemas e processos em etapas anteriores do processo mudam. Tenha mecanismos implementados que detectem essas alterações, para que você possa tomar as medidas adequadas.
- Automatize o pipeline de treinamento e avaliação

  A automação permite que você acione o treinamento automático de modelo e a criação de artefatos de modelo, que podem ser implantados de maneira consistente em vários ambientes de endpoint. A aplicação de automação para acionar atividades de retreinamento de modelo diminui o esforço manual, reduz o erro humano e oferece suporte ao aprimoramento contínuo da performance do modelo.
- Prefira abstrações mais altas para acelerar os resultados

  Ao selecionar o serviço adequado de IA/ML, primeiramente você deve avaliar a adequabilidade dos serviços de nível superior e, em seguida, como um mecanismo para atender rapidamente aos seus objetivos empresariais, remover tarefas pesadas genéricas e reduzir os custos de desenvolvimento.

Cenários

A seguir apresentamos alguns cenários comuns que influenciam o design e a arquitetura de suas cargas de trabalho de machine learning na AWS. Cada cenário inclui os impulsionadores comuns para o design e uma arquitetura de referência para mostrar a você como implementar cada cenário.

Tópicos

Criar aplicações inteligentes usando os serviços de IA da AWS

O nível de serviços de IA da AWS na pilha de machine learning é uma boa opção para organizações que desejam adicionar funcionalidades de IA a aplicações novas ou existentes com o mínimo de esforço de desenvolvimento e tempo rápido de execução. Nesse nível, os serviços oferecem capacidades de visão computacional, fala, linguagem natural e chatbot totalmente gerenciados e prontos para uso.

Ao usarem esses serviços, os desenvolvedores não precisam gerenciar as fases de preparação de dados, análise de dados, treinamento de modelos e avaliação vinculadas ao processo de ML. Em vez disso, esses recursos podem ser integrados a aplicações usando uma simples chamada de API.

O Amazon Comprehend é um produto de Natural Language Processing (NLP – Processamento de linguagem natural) que usa ML para ajudar você a descobrir insights e relações em dados de texto não estruturados. Primeiramente, o produto identifica o idioma do texto. Em seguida, ele extrai frases, lugares, pessoas, marcas e eventos principais. Ele analisa texto usando tokenização e partes da fala. Como o produto entende qual o nível de positividade ou negatividade do texto, ele pode organizar automaticamente uma coleção de arquivos de texto por tópico. Você também pode usar as capacidades do AutoML no Amazon Comprehend para criar um conjunto personalizado de entidades ou modelos de classificação de texto exclusivamente personalizados de acordo com as necessidades da sua organização.

O Amazon Lex é um produto para criar interfaces de conversação em qualquer aplicação usando voz e texto. O Amazon Lex fornece as funcionalidades avançadas de aprendizado profundo de Automatic Speech Recognition (ASR – Reconhecimento automático de fala) para converter fala em texto, além de Natural Language Understanding (NLU – Compreensão de linguagem natural) para reconhecer a intenção do texto. Esses recursos permitem que você crie aplicações com experiências de usuário altamente envolventes e interações conversacionais realistas.

O Amazon Polly é um produto que transforma texto em fala realista, permitindo que você crie aplicações que falam e crie categorias completamente novas de produtos habilitados por voz. O Amazon Polly é um produto de conversão de texto em fala que usa tecnologias avançadas de aprendizado profundo para sintetizar fala semelhante à voz humana.

O Amazon Rekognition facilita a inclusão de análises de imagens e vídeos às suas aplicações. Você fornece uma imagem ou vídeo à API do Amazon Rekognition e o serviço pode identificar os objetos, pessoas, texto, cenas e atividades, bem como detectar qualquer conteúdo inadequado. O Amazon Rekognition também fornece análise facial altamente precisa e reconhecimento facial em imagens e vídeos fornecidos por você. É possível detectar, analisar e comparar rostos para uma grande variedade de casos de uso de verificação de usuários, contagem de pessoas e segurança pública.

O Amazon Transcribe é um serviço de Automatic Speech Recognition (ASR – Reconhecimento automático de fala) que facilita adicionar a funcionalidade de conversão de fala em texto às suas aplicações. Usando a API do Amazon Transcribe, você pode analisar arquivos de áudio armazenados no Amazon S3 e fazer com que o serviço retorne um arquivo de texto transcrito da fala. Você também pode enviar um stream de áudio ao vivo para o Amazon Transcribe e receber um stream de transcrições em tempo real.

O Amazon Translate é um serviço de tradução automática por redes neurais que oferece tradução rápida, acessível e de alta qualidade. A tradução automática por redes neurais é uma forma de automação da tradução idiomas que utiliza modelos de aprendizado profundo para fornecer traduções mais precisas e naturais do que os tradicionais algoritmos de tradução baseada em regras e estatística. O Amazon Translate permite que você faça a localização de conteúdo (p. ex., sites e aplicações) para usuários internacionais e traduza facilmente grandes volumes de texto com eficiência.

As respostas dos produtos de IA da AWS também incluem uma pontuação de confiança que representa o nível de confiança do serviço de IA sobre um resultado específico. Como todos os sistemas de ML são probabilísticos por natureza, você pode encarar a pontuação de confiança como uma medida da confiança que os sistemas colocam em seus resultados. Ao usar os serviços de IA, certifique-se de definir um limite apropriado que seja aceitável para seu caso de uso específico. Para modelos de várias classes, use um limite por classe, definido com base nas taxas de erro de classe. Por exemplo, usar o Amazon Rekognition para avaliar o interesse da multidão em um evento pode exigir um limite menor de pontuação de confiança, mas usar o mesmo serviço para analisar imagens médicas pode exigir um limite mais alto. Casos de uso de domínios específicos com resultados impactantes, como análise de imagens médicas, também podem precisar de uma validação de segundo nível por um especialista médico.

Como os serviços de IA não têm servidor e seguem o modelo de pagamento conforme o uso, você pode ampliar o serviço de acordo com os negócios e manter seus custos baixos durante as fases de entrada e os períodos fora de pico. A natureza sem servidor dos serviços de IA os torna candidatos ideais para arquiteturas orientadas por eventos usando o AWS Lambda. Com o AWS Lambda, você pode executar código para praticamente qualquer tipo de aplicação ou serviço de back-end sem precisar administrar nada. Você paga somente pelo tempo de computação utilizado. Não há cobrança quando seu código não está em execução.

Um exemplo de caso de uso é capturar e analisar dados demográficos de clientes em uma loja de varejo, com o objetivo empresarial de melhorar a experiência e o envolvimento do cliente. Conforme captura e processa imagens de rostos, você deve implementar mecanismos de segurança para proteger esses dados e aplicar os níveis adequados de confiança antes de usar esses dados. A arquitetura de referência na Figura 2 mostra uma implementação usando o Amazon Rekognition para análise facial, o Amazon Athena para a análise dos dados de atributos faciais e o Amazon QuickSight para a visualização da análise.

O uso da tecnologia de análise facial deve estar em conformidade com todas as leis, inclusive leis que protegem os direitos civis. Os clientes da AWS são responsáveis por seguir todas as leis aplicáveis na forma como usam a tecnologia. A AWS Acceptable Use Policy (AUP – Política de uso aceitável) proíbe que os clientes usem qualquer produto da AWS, entre eles o Amazon Rekognition, para violar a lei, e os clientes que violarem nossa AUP não poderão usar nossos serviços.

Arquitetura de referência

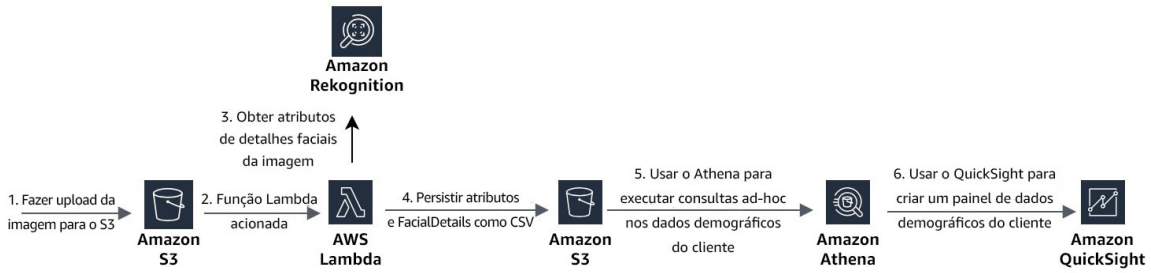

Figura 2 – Solução de análise demográfica do cliente

Essa arquitetura de referência inclui estes processos de alto nível:

- Crie um bucket do Amazon S3 para armazenar temporariamente imagens e habilite a criptografia no bucket para proteger as imagens. Restrinja o acesso ao bucket do S3 usando o AWS IAM: fornecendo permissões somente de gravação para o processo de upload (sem leitura pública) e permissões somente de leitura para a função do AWS Lambda. Habilite o registro em log de eventos de dados para o bucket do S3 em um bucket distinto do S3 usando o CloudTrail, para que você possa coletar logs de todas as atividades relacionadas ao bucket.
- As imagens de clientes capturadas na loja de varejo são carregadas no bucket do Amazon S3. Portanto, você precisa estabelecer políticas de ciclo de vida para garantir que uma imagem seja excluída automaticamente após o processamento.
- Cada imagem carregada no Amazon S3 aciona uma função do AWS Lambda, e você pode usar a análise facial para compreender dados demográficos, como idade, sexo e sentimento. A função do Lambda invoca o serviço Amazon Rekognition para extrair atributos faciais das imagens, que refletem a idade, o sexo e o sentimento do cliente, como feliz, calmo, irritado. As informações sobre a inferência também são incluídas nos atributos, juntamente com os níveis de confiança.
- Os dados demográficos são armazenados no formato.csv em um segundo bucket do Amazon S3. Criptografe o arquivo .csv usando uma chave exclusiva e armazenada com segurança. É possível usar um serviço como o AWS Key Management Service (AWS KMS) para gerenciar e armazenar essas chaves.
- O Amazon Athena lê e carrega os dados demográficos dos arquivos .csv para consultas. O Amazon Athena é compatível com dados criptografados para os dados de origem e para os resultados de consulta, p. ex., usando Amazon S3 com AWS KMS. Para garantir que os níveis de confiança sejam usados adequadamente, use Visualizações no Amazon Athena para restringir as pesquisas exclusivamente àquelas com um grau suficiente de confiança para o seu caso de uso.
- Crie painéis de insights sobre o cliente no Amazon QuickSight. Use o AWS IAM para restringir o acesso ao painel do Amazon QuickSight e consultas do Amazon Athena ao pessoal adequado, com o registro seguro em log de qualquer acesso.

Neste exemplo, o objeto de interesse é uma imagem e o Amazon Rekognition é usado para analisá-la. Implementa-se salvaguardas para proteger imagens faciais, excluir automaticamente as imagens e usar e gravar os níveis de confiança empregados na inferência. O uso correto dos níveis de confiança é imposto pela filtragem dos resultados de baixa confiança. É possível usar arquitetura para analisar uma variedade de tipos de objeto, como texto ou áudio, usando o serviço de IA adequado. Por exemplo, é possível transcrever um arquivo de áudio usando o Amazon Transcribe e analisar o texto não estruturado usando o Amazon Comprehend.

Adição de sofisticação

Embora os serviços individuais de IA sejam pré-treinados para processar uma tarefa específica (p. ex., análise de imagens ou transcrição), a natureza sem servidor desses serviços permite que você crie soluções sofisticadas orquestrando vários serviços usando o AWS Step Functions. Um exemplo é a solução de análise de mídia da AWS, que ajuda os clientes a analisar, compreender e criar facilmente um catálogo pesquisável de arquivos de mídia existentes. A arquitetura de referência a seguir usa vários serviços de IA (Amazon Rekognition, Amazon Transcribe, Amazon Comprehend) para analisar e extrair metadados da mídia. Os metadados extraídos são indexados e persistidos no Amazon OpenSearch Service para tornar pesquisável todo o conteúdo de mídia.

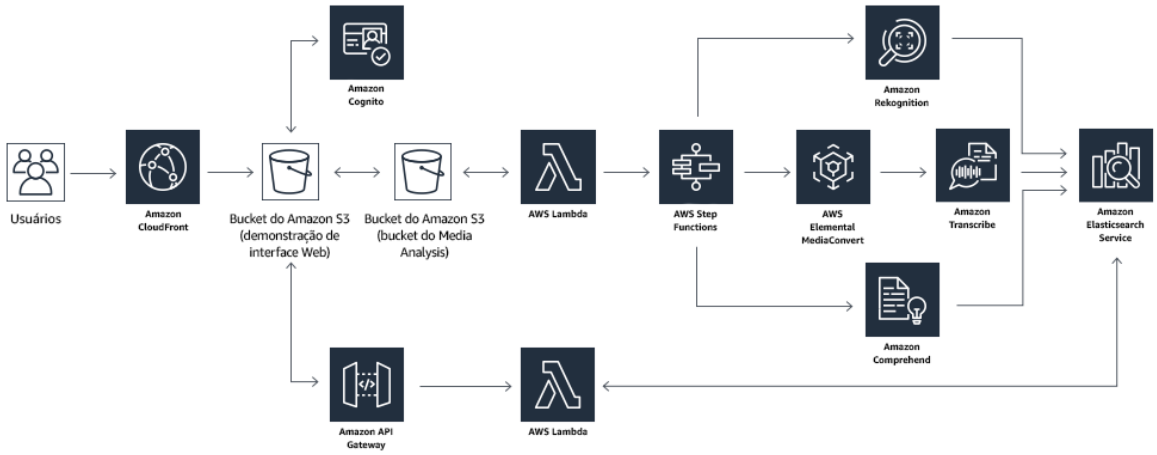

Figura 3 – Arquitetura de referência da solução de análise de mídia

Essa arquitetura de referência inclui estes processos de alto nível:

- Implante uma interface Web em um bucket do Amazon S3, o que permite iniciar imediatamente a análise de pequenos arquivos de mídia com uma interface Web simples. Use o Amazon CloudFront para restringir o acesso ao conteúdo do bucket do Amazon S3.
- Os arquivos de mídia carregados fluem por meio de uma API RESTful do Amazon API Gateway, uma função do AWS Lambda que processa solicitações de API, enquanto um grupo de usuários do Amazon Cognito permite uma interação segura com os arquivos de mídia.
- A máquina de estado do AWS Step Functions orquestra os processos de análise de mídia. Uma segunda função do Lambda executa a análise e a extração de metadados usando serviços gerenciados de IA, como Amazon Rekognition, Amazon Transcribe e Amazon Comprehend.
- Quando um arquivo de vídeo MP4 é carregado, o AWS Elemental MediaConvert extrai o áudio para análise pelo Amazon Transcribe e Amazon Comprehend.
- Os metadados resultantes são armazenados em um bucket do S3 e indexados em um cluster do Amazon OpenSearch Service (OpenSearch Service).

Os serviços de IA, que solucionam casos de uso específicos, como análise de imagens, tradução de idiomas, transcrição, permitem que você crie funcionalidades avançadas e inteligentes sem precisar de um amplo conhecimento em machine learning e aprendizado profundo. O resultado é um processo rápido de experimentação e avaliação em relação aos seus objetivos de negócios, o que resulta na redução do tempo de entrada no mercado. Neste exemplo, o impacto de um erro é baixo, portanto, você pode usar níveis de confiança mais baixos para qualquer uma de suas abordagens de ML.

Uso de serviços de IA com seus dados

Embora os serviços de IA abordados anteriormente sejam baseados em modelos pré-treinados, a AWS também oferece serviços de IA que retornam modelos de ML treinados com seus dados.

O Amazon Personalize é um serviço totalmente gerenciado que permite criar recomendações privadas e individualizadas de personalização para suas aplicações, com base nos dados de interação entre usuários e itens que você fornece. Seja uma recomendação de vídeo oportuna dentro de uma aplicação ou um e-mail personalizado de notificação entregue no momento certo, as experiências personalizadas, com base em seus dados, oferecem experiências mais relevantes para os clientes, muitas vezes com retornos comerciais muito superiores.

O Amazon Forecast é um serviço totalmente gerenciado que gera previsões altamente precisas com base nos dados históricos fornecidos por você. O serviço usa aprendizado profundo para aprender com vários

conjuntos de dados e é capaz de testar automaticamente diferentes algoritmos para a melhor adequação aos seus dados. É possível usá-lo para vários casos de uso, p. ex., estimar a demanda de produtos, uso de computação em nuvem, planejamento financeiro ou planejamento de recursos em um sistema de gerenciamento de cadeia de suprimentos.

Usar serviços gerenciados de ML para criar modelos personalizados de ML

Você pode adotar uma abordagem de serviços gerenciados para criar e implantar modelos de ML, usando seus próprios dados, para criar modelos preditivos e prescritivos visando agregar valor de negócios. Quando você usa serviços gerenciados de ML, suas equipes de desenvolvimento e ciência de dados são responsáveis por gerenciar as fases de preparação de dados, análise de dados, treinamento de modelos, avaliação do modelo e hospedagem de modelos do processo completo de ML.

O Amazon SageMaker é um serviço totalmente gerenciado que abrange todo o fluxo de trabalho de ML para rotular e preparar seus dados, escolher um algoritmo, treinar, ajustar e otimizar o modelo para implantação, fazer previsões e executar ações. Para permitir que desenvolvedores e cientistas de dados criem um modelo de ML sem a sobrecarga do gerenciamento genérico de infraestrutura, o Amazon SageMaker oferece as seguintes capacidades:

- Coletar e preparar dados de treinamento

  Rotule seus dados usando o Amazon SageMaker Ground Truth e aproveite vários blocos de anotações predefinidos para muitos problemas comuns de ML.
- Suporte para algoritmo de machine learning

  Escolha entre vários algoritmos integrados e de alta performance, traga seu próprio algoritmo ou explore o AWS Marketplace para obter algoritmos que se adequem ao seu caso de uso.
- Treinamento de modelos

  Treine o modelo de ML com seus próprios dados usando uma chamada de API que configura, gerencia e encerra um cluster de treinamento de alta performance. Configure o cluster de treinamento para usar uma única instância ou escolha várias instâncias para oferecer suporte a treinamento distribuído. O Amazon SageMaker Debugger fornece insight em tempo real sobre o processo de treinamento, automatizando a captura e a análise de dados com base em execuções de treinamento.
- Otimização de modelo

  Treine seu modelo uma vez no Amazon SageMaker e então otimize-o para outras estruturas de trabalho de ML usando o Amazon SageMaker Neo.
- Implantar seu modelo em produção

  Implante seus modelos treinados na infraestrutura com capacidade de dimensionamento automático de sua preferência usando uma chamada de API.
- Monitorar modelos implantados

  Monitore continuamente modelos de ML na produção para detectar desvios, como desvios de dados, que podem degradar a performance do modelo e automatizar ações de remediação.

O AWS Lambda é uma ferramenta compatível com arquitetura orientada por eventos e conecta várias fases do processo de ML em conjunto, desde a ingestão de dados até a realização de previsões.

Arquitetura de referência

A automação de um processo completo de ML usando Amazon SageMaker, Amazon Kinesis Data Streams, Amazon S3 e AWS Lambda é descrita na seguinte arquitetura de referência (Figura 4) para a solução Ciência de dados preditiva com Amazon SageMaker e um data lake na AWS.

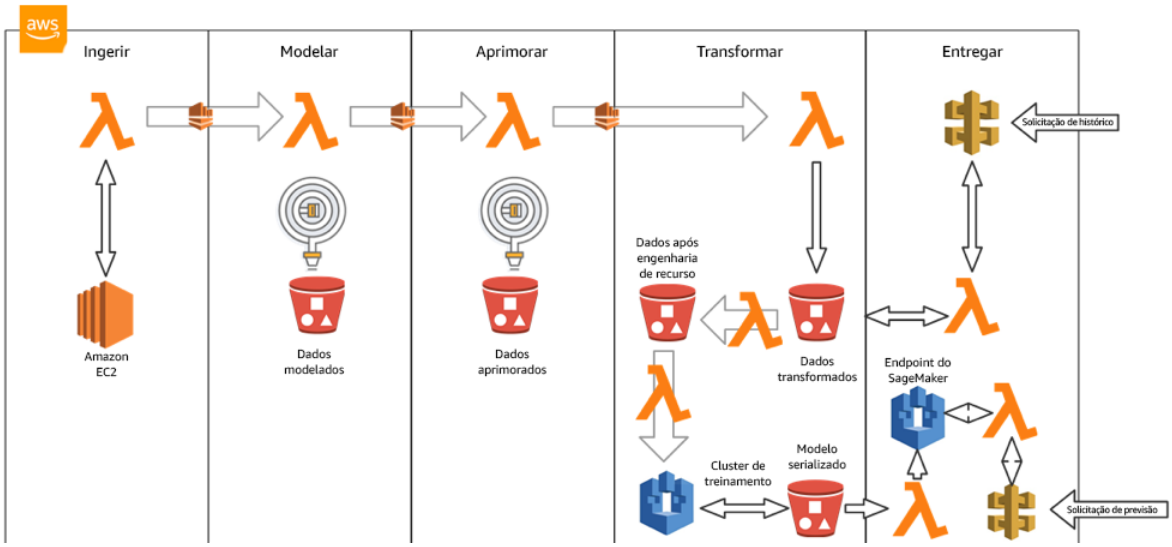

Figura 4 – Ciência de dados preditiva com Amazon SageMaker e um data lake na AWS

Essa arquitetura de referência inclui estes elementos de alto nível:

- O Amazon S3 é usado como um data lake que retém os dados brutos, modelados, aprimorados e transformados.
- O Amazon Kinesis Data Streams permite o processamento em tempo real de novos dados nos estágios Ingerir, Modelar, Aprimorar e Transformar.
- O código de transformação de dados é hospedado no AWS Lambda para preparar os dados brutos para consumo e treinamento de modelo de ML, e para transformar a entrada e a saída de dados.
- O AWS Lambda automatiza as chamadas de API do Amazon SageMaker para compilar, gerenciar e criar endpoints REST para novos modelos, com base em uma programação ou acionadas por alterações de dados no data lake.

Essa arquitetura oferece treinamento e aprimoramento contínuos e automatizados dos modelos de ML que usam dados do cliente, sem o trabalho pesado genérico de gerenciamento da infraestrutura.

O código de transformação de dados é hospedado no AWS Lambda. Também é possível executar o código de transformação de dados em uma instância de bloco de anotações do Amazon SageMaker. No entanto, essas opções podem não ser a escolha correta em todas as situações, especialmente para transformações de dados em grande escala.

Serviços gerenciados de ETL para processamento de dados

As atividades de processamento de dados, como limpeza, descoberta e engenharia de recursos em grande escala, são ideais para ferramentas como o Apache Spark, que é compatível com SQL para descoberta de dados, entre outros utilitários úteis. Na AWS, o Amazon EMR facilita o gerenciamento de

clusters do Spark e habilita funcionalidades como escalabilidade elástica enquanto minimiza custos por meio da definição de preço de instâncias spot.

Os blocos de anotações do Amazon SageMaker viabilizam conectividade com um cluster externo do Amazon EMR, permitindo o processamento de dados no cluster escalável de maneira elástica usando o Apache Spark. Em seguida, você pode treinar modelos e implantá-los usando as APIs de treinamento e implantação do Amazon SageMaker.

Por exemplo, imagine um caso de uso comercial de marketing direcionado para consumidores com base em compreensão profunda do comportamento do consumidor. o Amazon Pinpoint é um serviço gerenciado capaz de enviar mensagens direcionadas a consumidores por meio de vários canais de relacionamento, como e-mails, texto e SMS. Alguns exemplos de campanhas direcionadas incluem alertas promocionais e campanhas de retenção de clientes, além de mensagens transacionais, como confirmações de pedidos e mensagens de redefinição de senha. No entanto, identificar os clientes corretos, ou segmentos de clientes, para enviar a mensagem é um componente essencial. Você pode usar o ML para prever o comportamento futuro de compras com base em padrões históricos de compra de consumidores. Em seguida, é possível usar o comportamento de compra previsto para entregar campanhas direcionadas por meio do Amazon Pinpoint.

Arquitetura de referência

Essa arquitetura de referência mostra como você pode usar o Amazon EMR, o Apache Spark e o Amazon SageMaker para as diferentes fases do ML, e o Amazon Pinpoint para enviar mensagens de marketing direcionadas.

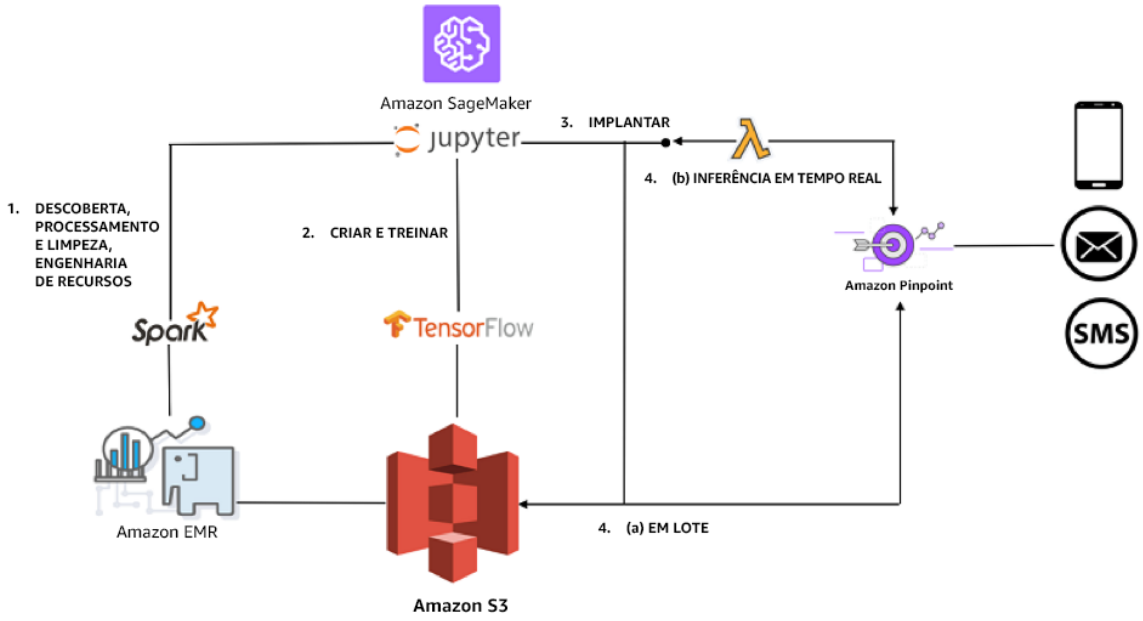

Figura 5 – Impulsão de campanha do Amazon Pinpoint por ML no Amazon SageMaker

Essa arquitetura de referência inclui estes elementos de alto nível:

- Use o Amazon S3 como um data lake que retém grandes volumes de dados.
- Configure um bloco de anotações do Amazon SageMaker para execução em relação a um cluster Amazon EMR externo. A limpeza, o processamento, a descoberta e a engenharia de recursos de dados são feitos usando o Apache cluster do EMR. Os dados transformados são armazenados no Amazon S3.
- Use o Amazon SageMaker para treinar um modelo personalizado usando os dados transformados e aproveitando a funcionalidade de treinamento distribuído.

- Use o Amazon SageMaker para criar um endpoint de API de Auto Scaling para o modelo treinado.
- Use o endpoint da API para fazer inferências em lote e em tempo real.
- Processe previsões em lotes e catalogue-as no data lake. Em seguida, a equipe de marketing pode importar os dados para o Amazon Pinpoint para iniciar uma campanha.

Machine learning na borda e em várias plataformas

O treinamento de seus modelos de ML exige a infraestrutura poderosa de computação disponível na nuvem. No entanto, geralmente a realização de inferências nesses modelos requer muito menos poder computacional. Em alguns casos, como com dispositivos de borda, a inferência precisa ocorrer mesmo quando não há conectividade ou a conectividade com a nuvem é limitada. Campos de mineração são um exemplo desse tipo de caso de uso. Para garantir que um dispositivo de borda possa responder rapidamente a eventos locais, é essencial que você obtenha resultados de inferência com baixa latência.

O AWS IoT Greengrass viabiliza o machine learning em dispositivos de borda. O AWS IoT Greengrass facilita a execução de inferência de ML localmente nos dispositivos, com modelos que são criados, treinados e otimizados na nuvem. Modelos de ML criados usando Amazon SageMaker, AWS Deep Learning AMI ou AWS Deep Learning Containers e persistidos no Amazon S3 são implantados nos dispositivos de borda.

A Figura 6 mostra a interação entre o AWS IoT Greengrass e o treinamento do modelo de ML na Nuvem AWS.

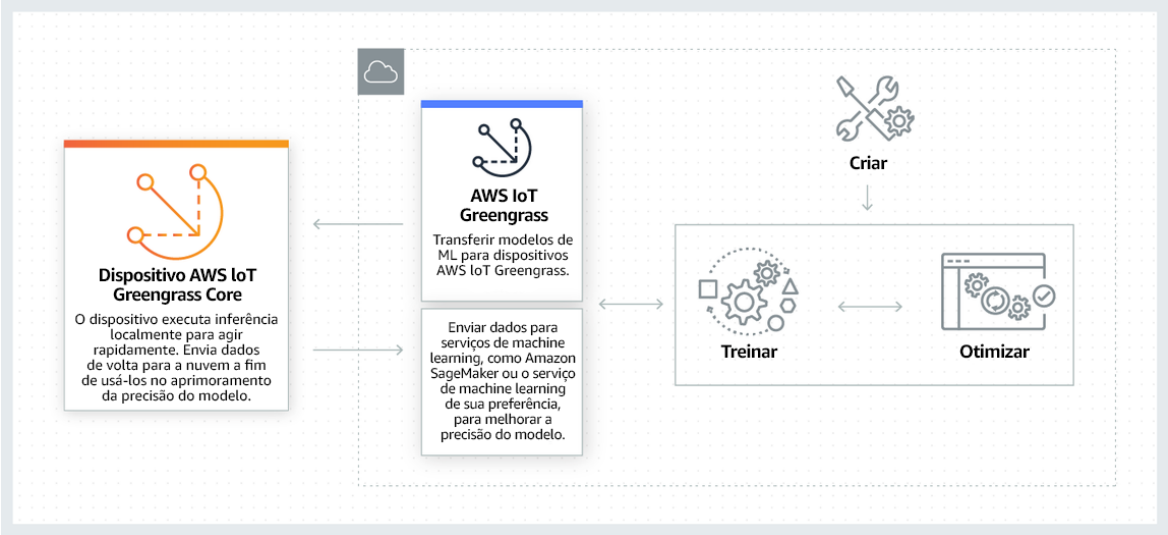

Figura 6 – AWS IoT Greengrass e o modelo de ML na nuvem

A execução de inferência localmente em dispositivos conectados executando o AWS IoT Greengrass reduz a latência e o custo. Em vez de enviar todos os dados do dispositivo para a nuvem a fim de executar inferência de ML e fazer uma previsão, você pode executar a inferência diretamente no dispositivo. Conforme as previsões são feitas nesses dispositivos de borda, você pode capturar os resultados e analisá-los para detectar exceções. Os dados analisados podem, então, ser enviados de volta para o Amazon SageMaker na nuvem, onde podem ser reclassificados e marcados para aprimorar o modelo de ML.

Você pode usar modelos de ML que são criados, treinados e otimizados na nuvem e executar sua inferência localmente em dispositivos. Por exemplo, você pode criar um modelo preditivo no Amazon SageMaker para análise de detecção de cenas, otimizá-lo para execução em qualquer câmera e implantá-lo para prever atividades suspeitas e enviar um alerta. Os dados coletados da inferência em execução no

AWS IoT Greengrass podem ser enviados de volta para o Amazon SageMaker, onde podem ser marcados e usados para aprimorar continuamente a qualidade dos modelos de ML.

Arquitetura de referência

Uma arquitetura de referência para um caso de uso de Identificação de espécies de pássaros na borda é apresentada na Figura 7. Nessa arquitetura, um modelo de detecção de objetos é treinado no Amazon SageMaker e implantado em um dispositivo de borda. A detecção de objetos personalizados tornou-se um importante viabilizador para uma grande variedade de setores e casos de uso, como encontrar tumores em ressonâncias magnéticas, identificar plantações com doenças e monitorar plataformas ferroviárias. O dispositivo de borda empregado nesse caso de uso é o AWS DeepLens, que é uma câmera de vídeo habilitada para aprendizado profundo.

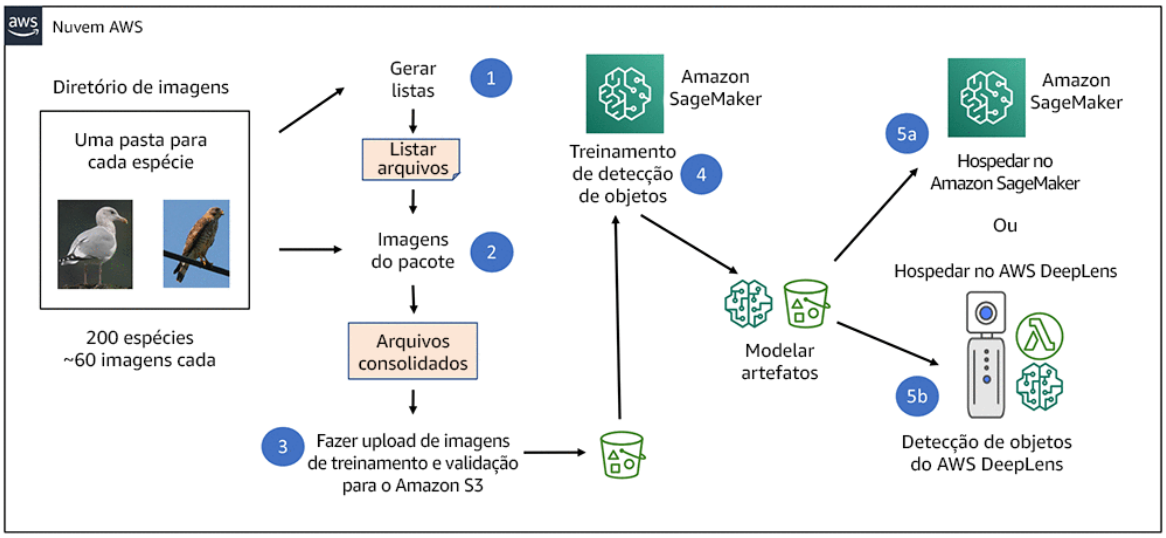

Figura 7 – Identificação de espécies de pássaros na arquitetura de borda

Essa arquitetura de referência inclui estes elementos:

- Coletar, compreender e preparar o conjunto de dados de imagens de pássaros
- Treinar o modelo de detecção de objetos usando o algoritmo integrado do Amazon SageMaker
- Hospedar o modelo usando um endpoint Amazon SageMaker
- Implantar o modelo na borda no AWS DeepLens:
  - Converter os artefatos de modelo antes de implantar no AWS DeepLens
  - Otimizar o modelo de sua função do AWS Lambda no AWS DeepLens
- Executar inferência de modelo e identificação de espécies de pássaros no AWS DeepLens

O AWS DeepLens é um dos dispositivos de borda usados na arquitetura anterior. Embora você possa implantar o modelo de ML na borda e na nuvem em várias plataformas de hardware, como Intel ou NVIDIA, nem sempre isso é prático, pois o modelo de ML está profundamente acoplado à estrutura de trabalho usada para treiná-lo, como MXNet, Tensor ou PyTorch. Se quiser implantar um modelo de ML em uma plataforma diferente da plataforma específica para a qual você o treinou, você deverá primeiro otimizar o modelo. Conforme o número de estruturas de trabalho e plataformas de ML aumenta, o esforço necessário para otimizar os modelos para plataformas adicionais aumenta e pode se tornar proibitivamente demorado.

O Amazon SageMaker Neo inclui dois componentes para resolver esse problema: um compilador e um tempo de execução. O compilador converte modelos em um formato comum e eficiente, que é executado no dispositivo por um tempo de execução compacto que usa menos de um centésimo dos recursos que

uma estrutura de trabalho genérica normalmente consome. O tempo de execução do Amazon SageMaker Neo é otimizado para o hardware subjacente e usa conjuntos de instruções específicos que ajudam a acelerar a inferência de ML. Os modelos são otimizados ocupando menos de um décimo do espaço de memória para que possam ser executados em dispositivos com recursos restritos, como acionadores e câmeras de segurança residencial.

Abordagens de implantação de modelo

Um modelo de ML treinado deve ser hospedado de modo que os consumidores possam invocá-lo e obter previsões dele com facilidade. Os consumidores dos modelos de ML podem ser externos ou internos à sua organização. Normalmente, os consumidores do modelo de ML não entendem o processo de ML e querem apenas uma API simples capaz de fornecer previsões em tempo real ou em modo de lote.

O Amazon SageMaker fornece serviços de hospedagem de modelos para a implantação de modelos, e fornece um endpoint HTTPS no qual o modelo de ML fica disponível para fornecer inferências.

A implantação de um modelo usando os serviços de hospedagem do Amazon SageMaker é um processo com três etapas:

1. Crie um modelo no Amazon SageMaker.

   Aplique as melhores práticas para garantir que o modelo atenda aos requisitos comerciais antes de prosseguir.
2. Crie uma configuração de endpoint para um endpoint HTTPS.

   Especifique o nome de um ou mais modelos em variantes em produção, além das instâncias de computação de ML que deseja que o Amazon SageMaker execute para hospedar cada variante de produção. A configuração do endpoint permite vincular vários modelos ao mesmo endpoint, com diferentes pesos e configurações de instância (variantes de produção). Você pode atualizar a configuração a qualquer momento durante a vida útil do endpoint.
3. Crie um endpoint HTTPS.

   O Amazon SageMaker inicia as instâncias de computação de ML e implanta o modelo (ou modelos) conforme especificado nos detalhes de configuração do endpoint, e fornece um endpoint HTTPS. Em seguida, os consumidores do modelo podem usar o endpoint para fazer inferências.

A funcionalidade de variantes de produção de configuração de endpoints de modelos do Amazon SageMaker permite a hospedagem de vários modelos de ML em infraestruturas diferentes, com cada um processando um subconjunto ou todas as solicitações de inferência. Você pode aproveitar variantes de produção para minimizar os riscos de implantação.

Para todas as variantes, inclua uma versão de modelo na resposta do endpoint de modelo. Quando houver problemas com inferências de modelo, ou em casos que exijam a explicação do modelo, saber a versão específica do modelo pode ajudar a rastrear as alterações até a origem.

Implantação padrão

Em uma implantação de modelo padrão, o endpoint do Amazon SageMaker é configurado com uma única variante de produção. A configuração da variante de produção especifica o tipo e a contagem da instância para hospedar o modelo. Todo o tráfego de inferência é processado pelo único modelo hospedado no endpoint.

Veja a seguir um exemplo de configuração de variante de produção para uma implantação padrão.

```
ProductionVariants=[{
  'InstanceType':'ml.m4.xlarge',
```

```
  'InitialInstanceCount':1,
  'ModelName':model_name,
  'VariantName':'AllTraffic'
}])
```

Implantações azul/verde

A técnica de implantação azul/verde fornece dois ambientes de produção idênticos. Você pode usar essa técnica quando precisar implantar uma nova versão do modelo na produção.

Conforme apresentado na Figura 8, essa técnica requer dois ambientes idênticos:

- Um ambiente de produção ativo (azul) que executa a versão n,
- Uma cópia exata desse ambiente (verde) que executa a versão n+1.

Enquanto o ambiente azul (versão n) está processando o tráfego ativo, você testa a próxima versão (versão n+1) no ambiente verde com tráfego sintético. Os testes devem incluir a verificação de que o novo modelo está atendendo às métricas técnicas e comerciais. Se todos os testes da versão n+1 no ambiente verde forem bem-sucedidos, o tráfego ao vivo será alternado para o ambiente verde. Em seguida, valide as métricas novamente no ambiente verde, dessa vez com tráfego ativo. Se você encontrar problemas neste teste, alterne o tráfego de volta para o ambiente azul. Se nenhum problema for encontrado durante um período, você pode remover o ambiente azul.

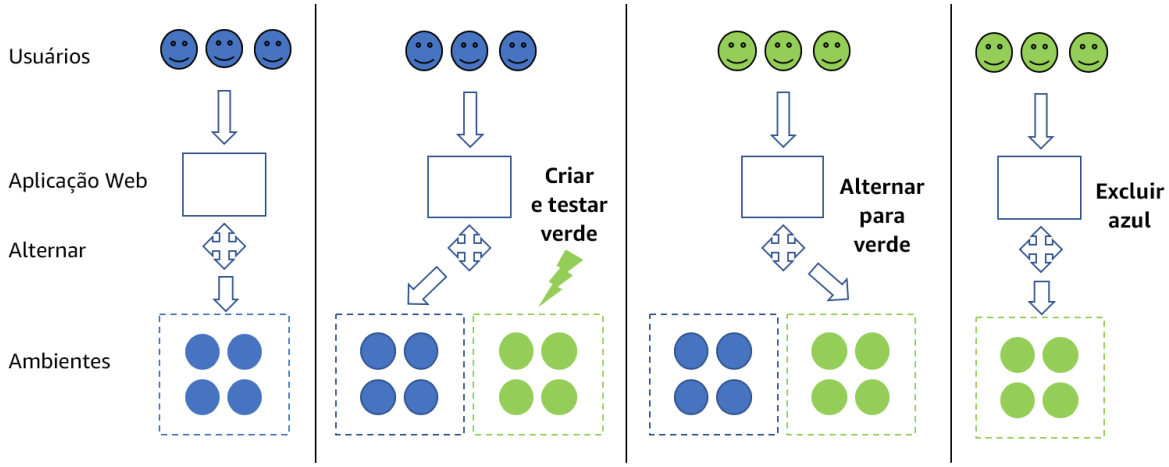

Figura 8 – Técnica de implantação azul/verde

A implementação de uma implantação azul/verde no Amazon SageMaker inclui estas etapas:

1. Criar uma nova configuração de endpoint usando as mesmas variantes de produção para o modelo ativo existente e para o novo modelo.
2. Atualizar o endpoint ativo existente com a nova configuração de endpoint. O Amazon SageMaker cria a infraestrutura necessária para a nova variante de produção e atualiza os pesos sem nenhum tempo de inatividade.
3. Alternar o tráfego para o novo modelo por meio de uma chamada de API.
4. Criar uma nova configuração de endpoint exclusivamente com a nova variante de produção e aplicá-la ao endpoint.

O Amazon SageMaker encerra a infraestrutura da variante de produção anterior.

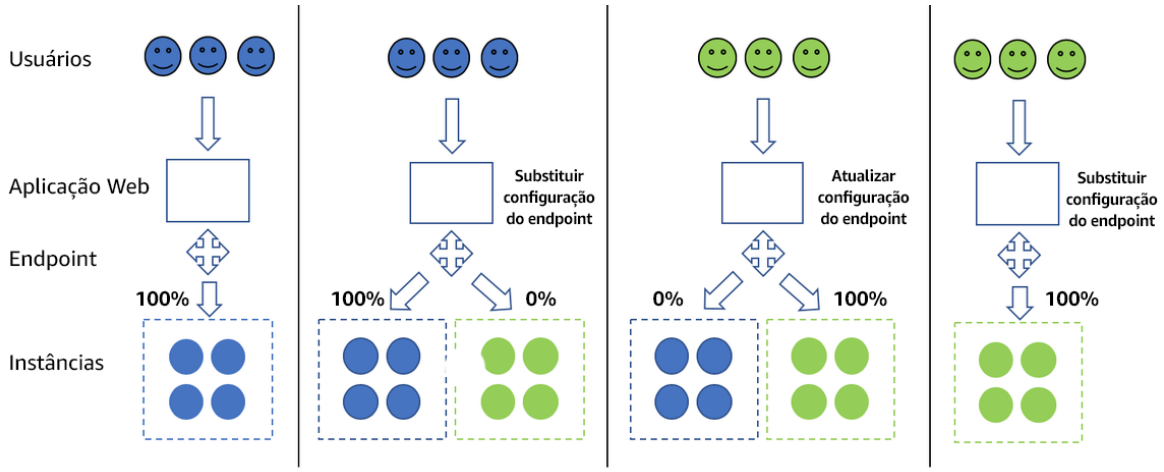

Figura 9 – Implantação de modelo azul/verde com variantes de produção do Amazon SageMaker

Implantação canário

Com uma implantação canário, você pode validar uma nova versão com risco mínimo ao implantá-la primeiramente para um pequeno grupo de seus usuários. Outros usuários continuam a usar a versão anterior até que você esteja satisfeito com a nova versão. Em seguida, você pode liberar gradualmente a nova versão para todos os usuários.

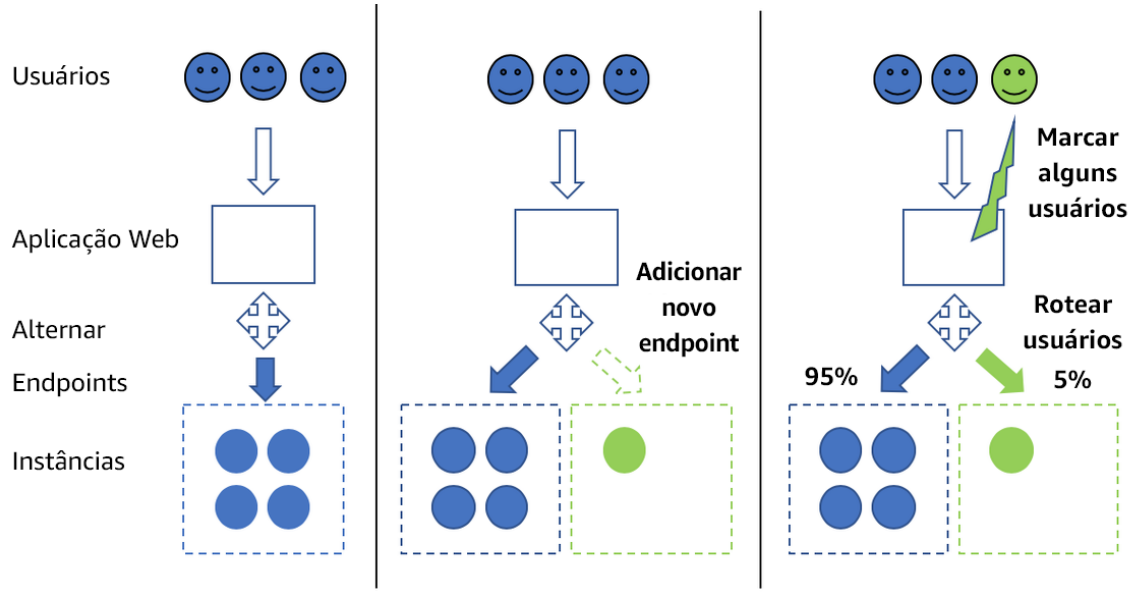

Figura 10 – Implantação canário com variantes de produção do Amazon SageMaker: distribuição inicial

Após ter confirmado que o novo modelo funciona conforme o esperado, você pode distribuí-lo gradualmente para todos os usuários, aumentando e diminuindo a escala dos endpoints de modo adequado.

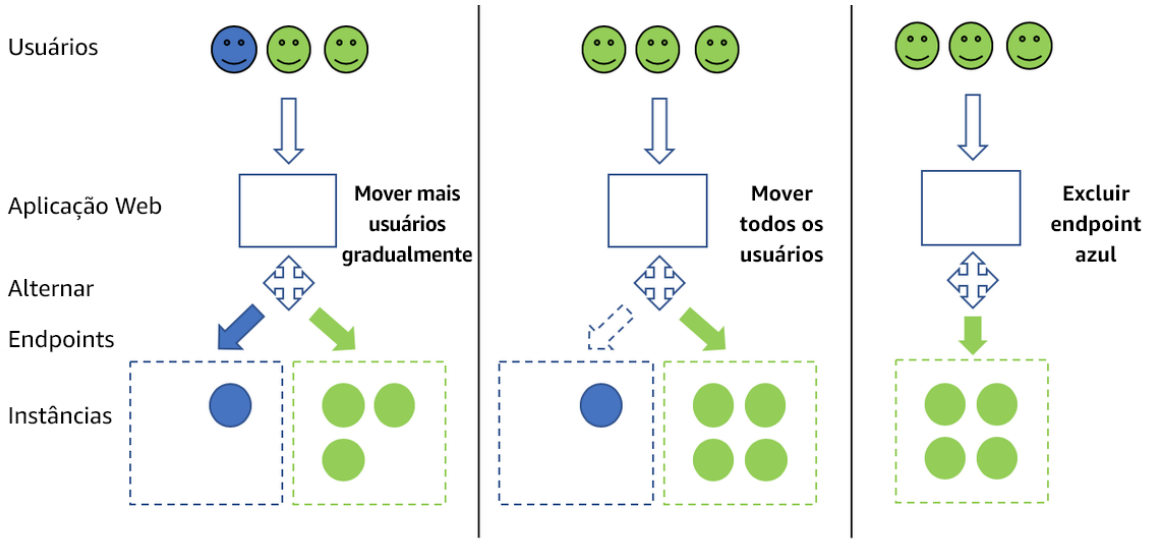

Figura 11 – Implantação canário com variantes de produção do Amazon SageMaker: distribuição completa

Testes A/B

Você pode usar a técnica de testes A/B para comparar a performance de diferentes versões do mesmo recurso enquanto monitora uma métrica de alto nível, como a taxa de cliques ou a taxa de conversão. Nesse contexto, isso significa fazer inferências usando modelos diferentes para usuários diferentes, analisando os resultados em seguida. Os modelos diferentes são criados usando o mesmo algoritmo (o algoritmo integrado do Amazon SageMaker ou seu algoritmo personalizado), mas usando duas configurações diferentes de hiperparâmetros.

Os testes A/B são semelhantes aos testes canário, mas têm grupos de usuários maiores e uma escala de tempo mais longa, normalmente dias ou até mesmo semanas. Para esse tipo de teste, a configuração de endpoint do Amazon SageMaker usa duas variantes de produção: uma para o modelo A e outra para o modelo B. Para começar, defina as configurações de ambos os modelos para equilibrar igualmente o tráfego entre os modelos (50/50) e certifique-se de que ambos os modelos tenham configurações idênticas de instância. Após ter monitorado a performance de ambos os modelos com a configuração inicial de pesos iguais, você pode alterar gradualmente os pesos do tráfego para retirar o equilíbrio dos modelos (60/40, 80/20 etc.) ou pode alterar os pesos em uma única etapa, continuando até que um único modelo esteja processando todo o tráfego ativo.

Veja a seguir uma amostra de configuração de variante de produção para testes A/B.

```
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
  }
```

```
])
```

Os pilares do Well-Architected Framework

Cada um dos seguintes pilares é importante para ajudar você a alcançar uma solução de carga de trabalho de machine learning bem arquitetada. Para cada pilar, abordamos apenas detalhes específicos à lente de Machine Learning, incluindo definições, melhores práticas, perguntas, implicações e os principais produtos da AWS que são específicos para cargas de trabalho de ML.

Ao projetar suas cargas de trabalho de ML, não esqueça de empregar também as melhores práticas e perguntas aplicáveis do whitepaper sobre o AWS Well-Architected Framework.

Tópicos

Pilar Excelência operacional

O pilar Excelência operacional inclui a capacidade de executar, monitorar e obter insights sobre sistemas para fornecer valor de negócios e aprimorar continuamente os processos e procedimentos auxiliares.

Tópicos

Princípios do design

Na nuvem, há vários princípios que podem ajudar você a fortalecer sua capacidade de otimizar aspectos operacionais das suas cargas de trabalho de ML. Ter a capacidade de operacionalizar essas cargas de trabalho é essencial para levar rapidamente cargas de trabalho de ML para o mercado.

As melhores práticas de excelência operacional da AWS foram projetadas para garantir que cargas de trabalho de ML operem com eficiência na nuvem. Para as práticas padrão de excelência operacional aplicáveis a todas as cargas de trabalho da AWS, consulte o whitepaper Pilar Excelência operacional: AWS Well-Architected Framework. Os princípios de design para otimizar a excelência operacional para cargas de trabalho de ML incluem:

- Estabelecer equipes interfuncionais: para garantir que as cargas de trabalho de ML tenham um caminho até a produção, inclua especialização entre funções e domínios nas equipes de projeto. Inclua todas as partes interessadas necessárias para desenvolver, implantar e oferecer suporte à sua carga de trabalho de ML.

- Identificar a arquitetura e o modelo operacional completos antecipadamente: no início do ciclo de vida de desenvolvimento de ML, identifique a arquitetura e o modelo operacional completos para treinamento e hospedagem de modelos. Isso permite a identificação precoce de fatores arquitetônicos e operacionais que serão necessários para o desenvolvimento, a implantação, o gerenciamento e a integração de cargas de trabalho de ML.
- Monitorar e mensurar continuamente as cargas de trabalho de ML: identifique e colete regularmente as métricas-chave relacionadas ao treinamento, hospedagem e previsões feitas em relação a um modelo. Isso garante que você seja capaz de monitorar continuamente a integridade de um modelo implantado entre os principais critérios de avaliação, como métricas do sistema, latência do modelo ou detecção de desvio de dados.
- Estabelecer uma estratégia de retreinamento de modelo: a performance e a eficácia de um modelo implantado podem mudar com o passar do tempo. Identifique métricas que indicam quando a performance e a eficácia de uma versão do modelo estão atendendo aos objetivos de negócios e crie alertas em limites que indicam que um modelo precisa ser treinado novamente para acionar essas atividades. Os alertas podem acionar atividades como invalidar o modelo atual, reverter para uma versão anterior do modelo, treinar novamente um novo modelo com base em novos dados da realidade prática ou equipes de ciência de dados refinando sua estratégia de retreinamento de modelos.
- Documentar as atividades e achados da descoberta de machine learning: as tarefas de descoberta e exploração de ciência de dados fornecem antecedentes e insights sobre a criação e a evolução de modelos de machine learning. Documente essas atividades em um pacote de código gerenciado e versionado no controle de origem.
- Versionar entradas e artefatos de machine learning: entradas e artefatos versionados permitem que você recrie artefatos para versões anteriores de sua carga de trabalho de ML. Versione entradas usadas para criar modelos, incluindo dados de treinamento e código-fonte de treinamento, além de artefatos de modelo. Versione também os algoritmos usados, o código-fonte de engenharia de recursos, as configurações de hospedagem, o código de inferência e os dados e o código-fonte de pós-processamento.
- Automatizar pipelines de implantação de machine learning: minimize pontos de contato humano em pipelines de implantação de ML para garantir que os modelos de ML sejam implantados de maneira consistente e repetida usando um pipeline que define como os modelos mudam do desenvolvimento para a produção. Identifique e implemente uma estratégia de implantação que atenda aos requisitos de seu caso de uso e problema de negócios. Se necessário, inclua pontos de controle humano de qualidade no pipeline para que humanos avaliem se um modelo está pronto para implantação em um ambiente de destino.

Melhores práticas

Há três áreas de melhores práticas para a excelência operacional na nuvem.

Tópicos

Preparar

Para se preparar para a excelência operacional, você precisa entender suas cargas de trabalho e os comportamentos esperados. Para se preparar para a prontidão operacional em cargas de trabalho de ML, você precisa avaliar:

- Prioridades operacionais
- Design das operações

- Prontidão operacional

MLOPS 01: como você preparou sua equipe para operar e oferecer suporte a uma carga de trabalho de machine learning?

Frequentemente as cargas de trabalho de ML são diferentes sob uma perspectiva de suporte, pois as equipes necessárias para integrar e implantar modelos de ML podem não estar familiarizadas com aspectos operacionais específicos de cargas de trabalho de ML. As melhores práticas para garantir que os modelos de ML sejam efetivamente integrados aos ambientes de produção e atendam aos objetivos de negócios incluem garantir a intercolaboração entre equipes e o treinamento de todos os recursos responsáveis pelo suporte e manutenção de cargas de trabalho de machine learning em níveis básicos de proficiência.

Muitas vezes, é necessário levar em consideração alguns requisitos operacionais para uma carga de trabalho de ML que talvez um cientista de dados não possa atender, como a capacidade de escalar ou modelar a latência. Por outro lado, também é necessário capturar comportamentos específicos de modelo cujas medidas a equipe operacional talvez não seja capaz de avaliar, como a eficácia do modelo ao longo do tempo.

Ao considerar sua abordagem para preparar equipes para integrar e operar cargas de trabalho de ML, as principais práticas incluem:

- Forneça treinamento cruzado de alto nível entre as equipes que desenvolverão modelos e APIs, e as equipes que oferecerão suporte ou terão responsabilidade de auditoria para sua carga de trabalho de ML.
- Estabeleça equipes interfuncionais para garantir que modelos e APIs possam ser efetivamente integrados a uma solução em produção. Isso elimina os obstáculos que normalmente podem impedir que cargas de trabalho de ML sejam implantadas e integradas a uma solução em produção.

MLOPS 02: como você está documentando as atividades de criação de modelos?

Um dos aspectos que torna o ciclo de vida de desenvolvimento do modelo de ML significativamente diferente do ciclo de vida de desenvolvimento de uma aplicação é a quantidade de experimentos necessária antes de finalizar uma versão de um modelo. Visando acrescentar clareza para oferecer suporte e trabalhar com a versão do modelo, documente o processo de criação do modelo, especialmente no que diz respeito às suposições feitas, ao pré-/pós-processamento de dados necessários para o modelo, bem como para integrar sistemas ou aplicações com a versão do modelo.

A documentação desse processo proporciona transparência ao modelo para outras partes interessadas responsáveis pela integração e pelo suporte dele. Armazenar essa documentação em um local seguro e versionado, como um repositório de controle de origem, também captura a propriedade intelectual relacionada à criação e evolução do modelo.

Na AWS, os blocos de anotações do Amazon SageMaker e o Amazon SageMaker Studio oferecem ambientes gerenciados de blocos de anotações onde cientistas de dados podem documentar seu processo de desenvolvimento e seus experimentos. Esses blocos de anotações podem ser integrados a sistemas de controle de origem e se tornar uma parte padrão da documentação criada para cada modelo implantado.

MLOPS 03: como você está rastreando a linhagem de modelos?

Ao desenvolver iterativamente seus modelos de ML usando diferentes algoritmos e hiperparâmetros para cada algoritmo, você terá muitos experimentos de treinamento de modelo e versões de modelo como resultado. Acompanhar esses modelos e rastrear a linhagem de qualquer modelo é importante não apenas para auditoria e conformidade, mas também para executar a análise de causa raiz da degradação da performance do modelo.

Além disso, sincronizar a linhagem de modelo com a linhagem de dados é importante porque, conforme as versões do código de processamento de dados e as versões de modelo são geradas, torna-se necessário documentar o pipeline completo de dados para treinamento de cada versão de modelo a fim de viabilizar a depuração de erros de modelo e também para auditorias de conformidade.

Na AWS, o Amazon SageMaker Experiments permite organizar e rastrear iterações de modelos de ML. O Amazon SageMaker Experiments captura automaticamente parâmetros de entrada, configurações e artefatos de saída para cada modelo, e os armazena como experimentos. Isso evita o uso de rastreamento manual ou a criação de soluções personalizadas de rastreamento para gerenciar as várias versões de artefatos de entrada e de saída criados e utilizados para cada iteração do desenvolvimento do modelo. Essa abordagem permite que as equipes escolham e implantem facilmente o modelo com os melhores resultados em vários experimentos.

Example

MLOPS 04: como você automatizou o pipeline de desenvolvimento e implantação para sua carga de trabalho de ML?

Crie uma arquitetura operacional que defina como sua carga de trabalho de ML será implantada, atualizada e operada como parte de seu design. A incorporação de práticas comuns de Infrastructure-as-Code (IaC - Infraestrutura como código) e Configuration-as-Code (CaC - Configuração como código) possibilita manter a consistência em implantações, bem como a capacidade de recriar recursos de maneira confiável entre os ambientes. Além disso, garantir que haja um mecanismo automatizado para orquestrar a movimentação de uma carga de trabalho de ML entre fases e ambientes de destino de maneira controlada reduz o risco ao atualizar sua carga de trabalho.

Incorpore práticas de Continuous Integration/Continuous Delivery (CI/CD – Integração e entrega contínuas) às cargas de trabalho de ML (MLOps) para garantir que a automação inclua rastreabilidade juntamente com pontos de controle de qualidade. Por exemplo, os pipelines de CI/CD começam com o controle de versão de origem e artefato que oferece suporte a atividades padrão de gerenciamento de alterações, além de proporcionar níveis mais altos de confiança para atividades de depuração. A aplicação da prática de controle de origem, dados e versão de artefatos a cargas de trabalho de ML melhora as atividades de depuração operacional, permitindo a rastreabilidade de volta para as versões implantadas. Além disso, o versionamento permite a capacidade de reverter para uma versão de trabalho conhecida específica após uma alteração com falha ou quando um novo modelo não fornece a funcionalidade necessária.

A implementação de monitoramento e registro em log em todo o pipeline de CI/CD também estabelece a base para a inserção de pontos de controle de qualidade, permitindo ou negando a implantação em ambientes de nível superior. As melhores práticas incluem pontos de controle padrão de qualidade, como verificar contêineres em busca de vulnerabilidades de pacotes e garantir que portas de qualidade específicas de ML sejam incluídas no pipeline. Esses portões de qualidade devem avaliar modelos usando métricas identificadas e específicas para seu caso de uso empresarial. Isso pode incluir métricas como avaliação de precisão-recall, F1 ou precisão. A injeção de pontos de controle de qualidade ajuda a garantir que uma versão mais recente do modelo não substitua um modelo atualmente implantado quando houver a identificação de uma condição que indique uma preocupação operacional, como uma exposição de segurança ou uma diminuição na performance do modelo ou nas métricas de precisão.

Na AWS, serviços de IA como o Amazon Polly são fornecidos por meio de um endpoint de API. Como resultado, não há melhores práticas exclusivas nessa área, pois o modelo já está treinado e implantado. A

automação de desenvolvimento e implantação relacionada ao código e aos sistemas que interagem com esse endpoint de API deve seguir as melhores práticas padrão da AWS. Alguns serviços de IA da AWS, como o Amazon Personalize, treinam um modelo com base nos dados de treinamento que você fornece. Siga as melhores práticas descritas neste whitepaper para proteger e salvaguardar seus dados ao criar ou atualizar modelos nesses serviços.

Ao criar e treinar seus próprios modelos de ML na AWS, a automação do pipeline de desenvolvimento e implantação é obtida por meio de uma combinação de produtos da AWS e integrações de terceiros. A identificação do serviço ou das ferramentas corretos a serem usados para criar um pipeline automatizado de implantação de modelo depende da identificação da estratégia de implantação, das características do modelo e da estratégia de treinamento de modelo.

Cada carga de trabalho de ML varia de acordo com os produtos de ML da AWS que estão sendo usados. No entanto, uma diretriz geral para a criação de pipelines inclui o uso de uma camada de orquestração, como o AWS CodePipeline, combinada com uma lógica que seja responsável pela execução das etapas dentro do pipeline. Use o AWS Lambda para criar e executar a lógica baseada em função, pois o serviço tem uma baixa sobrecarga operacional e não tem servidores para gerenciar. A figura a seguir representa um pipeline de referência para implantação na AWS. No entanto, essa implantação variará de acordo com os fatores discutidos anteriormente.

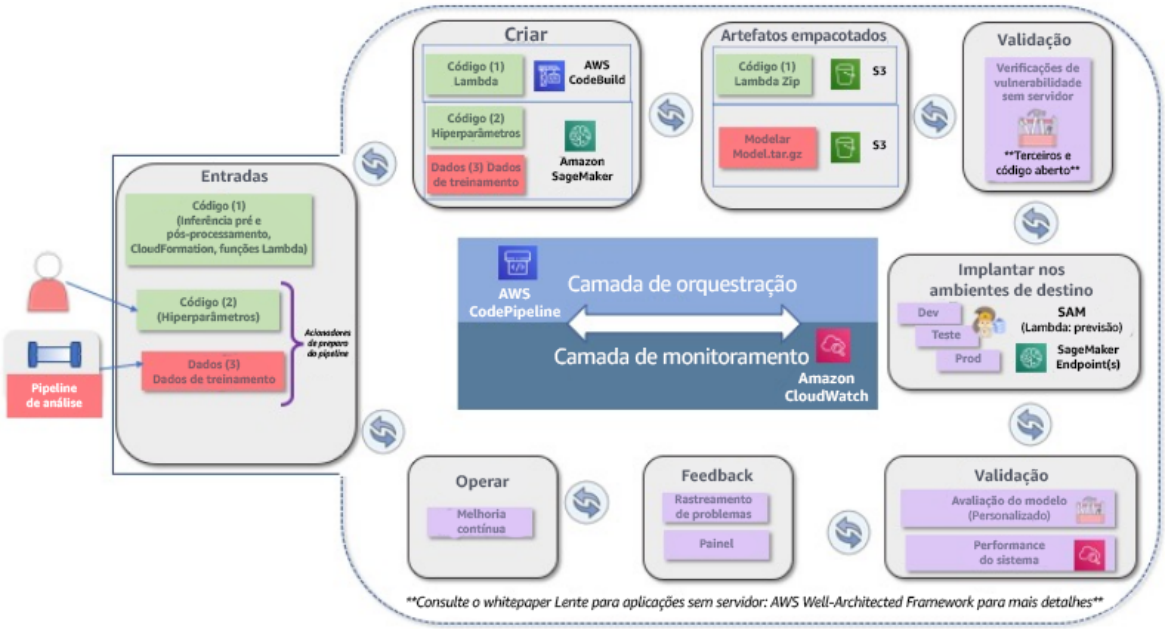

Figura 12 – Pipeline de referência de CI/CD de MLOps para machine learning na AWS

Operar

MLOPS 05: como você está monitorando e registrando em log as atividades de hospedagem de modelos?

Ao hospedar um endpoint de modelo para previsões, o endpoint deve ter monitoramento e alertas estabelecidos para identificar e reagir a possíveis problemas ou oportunidades de aprimoramento. Os endpoints de modelo devem incluir monitoramento e alertas sobre métricas que medem a integridade operacional dos recursos computacionais subjacentes que hospedam o endpoint, bem como o monitoramento da integridade das respostas de endpoint.

Na AWS, as práticas padrão para gerenciar a integridade operacional de recursos de computação de endpoint devem incluir aquelas já definidas no whitepaper AWS Well Architected: Excelência operacional. O Amazon SageMaker monitora automaticamente métricas básicas do sistema e também inclui recursos para configurar capacidades de escalabilidade automática para que seu modelo hospedado ajuste de maneira dinâmica a computação adjacente que oferece suporte ao um endpoint com base na demanda. Essa funcionalidade garante que o endpoint possa atender dinamicamente a demanda enquanto reduz a sobrecarga operacional.

Além de monitorar recursos de computação para viabilizar a escalabilidade automática, o Amazon SageMaker também gera métricas de endpoint para monitorar o uso e a integridade do endpoint. O Amazon SageMaker Model Monitor oferece a capacidade de monitorar seus modelos de ML em produção e fornece alertas mediante o surgimento de problemas de qualidade de dados. As melhores práticas incluem a criação de um mecanismo para agregar e analisar métricas de endpoint de previsão de modelo usando serviços, como o Amazon OpenSearch Service, com suporte integrado ao Kibana para painéis e visualização. Além disso, ter a capacidade de garantir a rastreabilidade da hospedagem de métricas em relação às entradas versionadas permite a análise de alterações que poderiam afetar a performance operacional atual.

Evoluir

MLOPS 06: como saber quando retreinar modelos de ML com dados novos ou atualizados?

Inicialmente as cargas de trabalho de ML podem fornecer previsões de alto valor, mas a precisão das previsões do mesmo modelo pode se degradar ao longo do tempo. Muitas vezes, isso ocorre devido a um conceito conhecido como desvio, que pode ser resultado de muitos fatores que incluem alterações nos dados da realidade prática no decorrer do tempo. À medida que as previsões de modelo são integradas às decisões empresariais, isso pode afetar indiretamente a performance dos modelos existentes. Por exemplo, considere um cenário de varejo prevendo o risco associado a uma remessa específica, em que os dados de treinamento incluem remessas danificadas anteriores. Conforme a empresa começa a usar o modelo para tomar decisões empresariais, isso afeta indiretamente os dados, pois haverá menos instâncias de produtos danificados.

Frequentemente há a necessidade de retreinar um modelo usando dados novos ou atualizados para garantir que o modelo seja capaz de aprender e prever de maneira eficaz com base nos dados mais recentes disponíveis. Para ser capaz de incorporar efetivamente dados adicionais em um modelo de ML, deve haver um mecanismo implementado para analisar a performance do modelo existente em relação a métricas definidas e acionar um alarme ou um evento de retreinamento quando a variância do modelo atingir um limite específico, ou retreinar proativamente o modelo ao longo do tempo com base em novos dados conhecidos.

As melhores práticas para levar em consideração dados adicionais incluem:

- Definir métricas que sejam indicativas da performance e da precisão do modelo
- Garantir que exista um mecanismo implementado para capturar regularmente essas métricas para análise e alerta com base nos limites de métrica. Por exemplo, pode ser necessário haver um sistema capaz de identificar, capturar ou rastrear resultados em etapas posteriores do processo em relação a previsões específicas de modelo, de modo que as métricas, como taxas de erro, possam ser calculadas ao longo do tempo.
- Avaliar se é adequado treinar novamente o modelo. Identificar se há disponibilidade ou é possível adquirir dados reais adicionais, ou se é necessário marcar dados adicionais. Decida uma estratégia inicial para retreinamento com base em características de cargas de trabalho conhecidas, como treinamento programado regularmente com novos dados, novos dados como gatilho para o retreinamento ou avalie o retreinamento com base em limites de métrica. A estratégia deve avaliar compensações entre a quantidade de alteração, o custo de retreinamento e o valor potencial de ter um

modelo mais recente em produção. Configure o retreinamento automatizado com base na estratégia definida.

Na AWS, serviços de IA, como o Amazon Translate, são treinados automaticamente com novos dados para que você possa aproveitar um modelo atualizado pela AWS para melhorar a performance do modelo ao longo do tempo.

Ao usar os serviços de ML na AWS para criar e treinar seus próprios modelos, a AWS fornece várias funcionalidades para atender ao retreinamento contínuo de modelos com novos dados. Armazene dados preparados usados para treinamento no Amazon S3. Os seguintes cenários de retreinamento estão incluídos e devem ser considerados com base nas características da carga de trabalho:

- Desvio de modelo (retreinamento orientado por métricas): para cargas de trabalho de ML que são sensíveis a variações, como quando a distribuição de dados se desvia significativamente dos dados de treinamento originais ou há um aumento nos dados fora da amostra, configure um mecanismo automatizado para acionar o retreinamento de um modelo com base em uma métrica definida ou na presença de novos dados preparados. Na AWS, um mecanismo para identificar um desvio nos dados inclui a utilização do Amazon SageMaker Model Monitor para detectar a mudança na distribuição dos dados. A detecção de desvio é disponibilizada por meio de métricas do AWS CloudWatch, que podem ser usadas para acionar automaticamente trabalhos de retreinamento.
- Dados adicionais de treinamento: a AWS oferece suporte a mecanismos para acionar automaticamente o novo treinamento com base em PUT de novos dados para um bucket do Amazon S3. O método preferencial para iniciar uma execução controlada de retreinamento de modelo é configurar um pipeline de ML que inclua um trigger de evento com base em alterações em um bucket do Amazon S3 de origem. Para detectar a presença de novos dados de treinamento em um bucket do S3, o CloudTrail combinado com o CloudWatch Events permite que você acione uma função do AWS Lambda ou fluxo de trabalho do AWS Step Functions para iniciar tarefas de retreinamento em seu pipeline de treinamento. A figura a seguir ilustra a prática que mostra o AWS CodePipeline com serviços de ML:

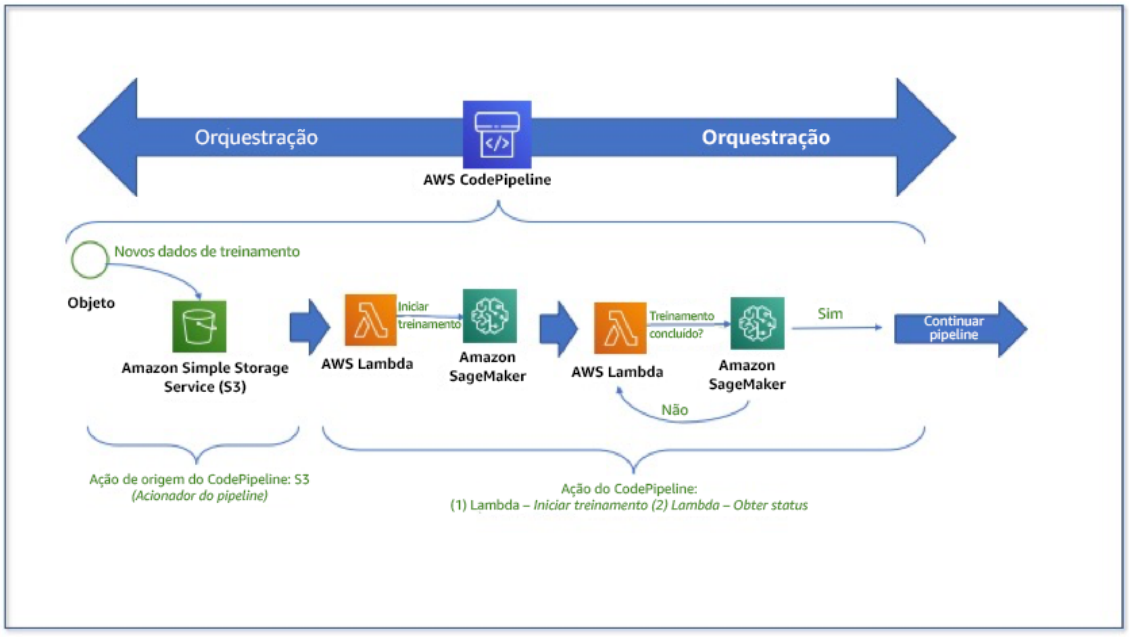

Figura 13 – Exemplo de trigger de evento para novos dados de treinamento de serviços de ML

Como alternativa, você também pode usar ferramentas de orquestração de implantação de terceiros, como Jenkins, que se integram às APIs do produto da AWS para automatizar o retreinamento de modelos mediante a disponibilidade de novos dados.

Ao definir sua estratégia para incorporar dados adicionais aos modelos, certifique-se de que a estratégia seja compatível com um versionamento de modelos que mantenha todos os dados de treinamento anteriores em sua forma original, ou que versões anteriores dos dados de treinamento sejam facilmente reproduzidas. Isso garante que o mesmo artefato de modelo possa ser recriado usando as versões combinadas de todos os componentes usados para criar o artefato versionado em caso de exclusão acidental de um artefato de modelo.

MLOPS 07: como incorporar aprendizados entre iterações de desenvolvimento, treinamento e hospedagem de modelos?

Para incorporar aprendizados, é fundamental ter um mecanismo de feedback contínuo implementado que ofereça a capacidade de compartilhar e comunicar experiências de desenvolvimento bem-sucedidas, análise de falhas e atividades operacionais. Isso facilita a capacidade de melhorar continuamente em futuras iterações da carga de trabalho de ML.

As principais considerações para aprendizados devem incluir a avaliação do modelo nas seguintes dimensões:

- Avaliação empresarial: para validar o sucesso de um modelo em relação à meta comercial, você deve garantir a existência de métricas empresariais de referência, bem como de um mecanismo para coletar e monitorar continuamente essas informações ao longo do tempo. Por exemplo, se o objetivo da sua empresa for aumentar as vendas de um produto direcionando clientes específicos para campanhas publicitárias, é necessário criar uma linha de base e incluir um mecanismo operacional para medir continuamente os Key Performance Indicators (KPIs – Indicadores-chave de performance) do sucesso, como vendas de um produto, clientes direcionados e clientes que compram o produto.
- Avaliação do modelo: para validar o sucesso do modelo em relação ao problema de ML que você enquadrou, é necessário capturar as principais métricas relacionadas à performance do modelo no pipeline completo. Isso inclui métricas de treinamento, como erros de treinamento ou validação, bem como métricas contínuas para um modelo hospedado, como precisão de previsão. Deve-se escolher as métricas específicas com base no caso de uso e nos KPIs empresariais.
- Avaliação do sistema: para validar os recursos do sistema usados para viabilizar as fases das cargas de trabalho de ML, é fundamental coletar e monitorar continuamente recursos do sistema, como computação, memória e rede. Os requisitos para cargas de trabalho de ML mudam em fases diferentes. Por exemplo, os trabalhos de treinamento usam mais memória, enquanto os trabalhos de inferência usam mais computação.

Na AWS, além das práticas padrão nessa área, você também pode utilizar instâncias de bloco de anotações do SageMaker para capturar atividades de exploração de ciência de dados, fornecendo documentação e explicação detalhadas do ciclo de vida de desenvolvimento de modelos. Isso é essencial não apenas para permitir que você tenha êxito para apoiar um modelo em produção, mas também para fornecer visibilidade e rastreabilidade das atividades de vários cientistas de dados e desenvolvedores conforme os modelos evoluem. Além disso, oferecer visibilidade centralizada às principais métricas operacionais coletadas permite que as equipes analisem e executem análises retrospectivas de suas operações ao longo do tempo.

Recursos

Consulte os recursos a seguir para saber mais sobre nossas melhores práticas de excelência operacional.

Documentação e blogs

- Crie fluxos de trabalho completos de machine learning com Amazon SageMaker e Apache Airflow
- Implantação contínua e automatizada de modelos Amazon SageMaker com AWS Step Functions
- Gerencie Amazon SageMaker com o Step Functions
- Como criar um pipeline usando AWS CodePipeline e AWS Lambda

Whitepapers

- Pilar Excelência operacional – AWS Well-Architected Framework
- Lente de aplicações sem servidor – AWS Well-Architected Framework

Pilar Segurança

O pilar Segurança inclui a capacidade de proteger informações, sistemas e ativos enquanto agrega valor empresarial por meio de avaliações de risco e estratégias de mitigação.

Tópicos

Princípios do design

Além dos princípios gerais de design de segurança do Well-Architected Framework, há princípios de design específicos para a segurança de ML:

- Restringir o acesso aos sistemas de ML: o nível de acesso aos sistemas de ML deve ser levado em conta ao projetar o sistema. O acesso aos modelos de ML e aos conjuntos de dados usados para treinar o modelo deve ser restrito para evitar a contaminação dos dados e do modelo. Os endpoints de inferência devem ser protegidos para que apenas partes autorizadas possam fazer inferências em relação ao modelo de ML.
- Garantir a governança de dados: os dados usados para ML podem ser coletados de várias fontes e precisam estar disponíveis para várias equipes em toda a organização. Como os dados de produção são necessários não apenas para atividades de desenvolvimento de ciência de dados, mas também para modelos de treinamento, garantir que as equipes tenham acesso completo a conjuntos de dados de alta qualidade exige uma estratégia de governança de dados que garanta a integridade, a segurança e a disponibilidade dos conjuntos de dados. A implementação de soluções de data lake com controles de governança e acesso garante que desenvolvedores e cientistas de dados controlem o acesso a dados de qualidade para uso em atividades de exploração e modelos de treinamento. Os dados também devem ser protegidos contra exfiltração ou mutação. Controle quais ações diferentes equipes em sua organização podem executar sobre os dados e para onde podem enviá-los.
- Aplicar linhagem de dados: como dados de várias fontes são usados em diferentes fases do processo de ML, monitore e rastreie as origens e transformações dos dados ao longo do tempo. A linhagem de dados viabiliza a visibilidade e simplifica o processo de rastreamento de erros de processamento de dados e machine learning até a causa raiz. Controle rigorosamente quem pode acessar os dados e o que é possível fazer com eles. Controles preventivos, auditoria e monitoramento são necessários para demonstrar como os dados foram controlados durante sua vida útil.
- Aplicar conformidade normativa: as questões normativas relacionadas aos sistemas de ML incluem implicações de privacidade, como as descritas na HIPAA ou no GDPR, em que o sistema de ML deve

aderir às diretrizes normativas estabelecidas nessas estruturas de trabalho. Elas também podem incluir questões de gerenciamento de riscos financeiros, como as diretrizes SR 11-7 do Federal Reserve. Ao contrário dos modelos tradicionais, nos quais os algoritmos permanecem estáticos, os modelos que aproveitam algoritmos de ML/IA podem evoluir ao longo do tempo. Portanto, é necessário realizar vigilância contínua para garantir a conformidade com órgãos normativos.

Melhores práticas

Há cinco áreas de melhores práticas de segurança na nuvem.

Tópicos

Gerenciamento de identidade e acesso

MLSEC 01: como controlar o acesso à sua carga de trabalho de ML?

Normalmente, várias equipes estão envolvidas na criação de cargas de trabalho de ML, com cada equipe responsável por uma ou mais fases de ML. O acesso a todos os recursos usados nas várias fases do processo de ML, incluindo dados, algoritmos, hiperparâmetros, artefatos de modelo treinados e infraestrutura, deve ser rigorosamente controlado com acesso baseado no mínimo de privilégio. Por exemplo, uma equipe responsável pela engenharia de recursos pode não ser responsável pelo treinamento ou implantação do modelo e, portanto, não deve ter permissões para fazer isso. Da mesma forma, uma equipe de operações do responsável por implantar o modelo na produção não deve ter permissões para acessar ou modificar dados de treinamento. Algumas cargas de trabalho podem ter membros da equipe com responsabilidades sobrepostas em várias fases das cargas de trabalho de ML e exigir permissões adequadas para executar responsabilidades de função.

Na AWS, o acesso aos vários recursos e serviços é controlado por meio do AWS IAM. Embora utilize-se identidades para autenticação, o controle detalhado sobre quem (humanos) e o quê (processos) pode acessar os dados, modificar os dados e os algoritmos, iniciar trabalhos de treinamento e implantar modelos, é implementado por meio de usuários, grupos, funções e políticas do IAM.

Restrinja o acesso a um modelo implantado exclusivamente aos consumidores legítimos previstos. Para consumidores de modelos que estão localizados em seu ambiente da AWS ou têm os meios para recuperar credenciais temporárias do IAM para acessar seu ambiente, use uma função do IAM com permissões de privilégio mínimo para invocar o endpoint do modelo implantado. Para consumidores externos ao seu ambiente, forneça acesso por meio de uma API segura usando uma combinação de API Gateway e endpoints de modelo hospedados.

Controles de detecção

Consulte o whitepaper do AWS Well-Architected Framework para conhecer as melhores práticas na área de controles de detecção para segurança aplicáveis a ML.

Proteção de infraestrutura

Consulte o whitepaper do AWS Well-Architected Framework para conhecer as melhores práticas na área de proteção de infraestrutura para segurança aplicáveis a ML.

Proteção de dados

MLSEC 02: como você está protegendo e monitorando o acesso a dados confidenciais usados em suas cargas de trabalho de ML?

Os dados são usados em todas as fases no processo de ML. No início de um projeto, após identificar os objetivos empresariais, você avalia a acessibilidade e a disponibilidade de várias fontes de dados, e interage com os dados disponíveis. Antes que a parte de ML de um projeto possa começar, normalmente já existe um data lake centralizado ou ele é criado. Proteja dados em seu data lake em repouso e enquanto ele se movimenta pelas diferentes fases do processo de ML. As equipes em sua organização não precisam ter acesso a todos os dados. Classifique os dados, implemente controles de acesso granulares com base no mínimo de privilégio para várias partes dos dados e monitore continuamente o acesso a eles.

Na AWS, um data lake centralizado é implementado usando o AWS Lake Formation no Amazon S3. É possível proteger e monitorar um data lake no Amazon S3 usando uma combinação de vários serviços e capacidades para criptografar dados em trânsito e em repouso, além de monitorar o acesso incluindo políticas granulares do AWS IAM, políticas de bucket do S3, logs de acesso do S3, Amazon CloudWatch e AWS CloudTrail. O whitepaper Como criar soluções para armazenamento de big data (data lakes) visando o máximo de flexibilidade (em inglês) aborda o uso dessas várias capacidades para criar um data lake seguro.

Além de implementar controle de acesso por meio do AWS IAM, use o Amazon Macie para proteger e classificar dados no Amazon S3. O Amazon Macie é um serviço de segurança totalmente gerenciado que usa machine learning para descobrir, classificar e proteger automaticamente dados confidenciais na AWS. O serviço reconhece dados confidenciais, como Personally Identifiable Information (PII – Informações de identificação pessoal) ou propriedade intelectual, e fornece visibilidade a sobre como esses dados estão sendo acessados ou movimentados. O Amazon Macie monitora continuamente a atividade de acesso aos dados em busca de anomalias, e gera alertas detalhados ao detectar um risco de acesso não autorizado ou vazamentos acidentais de dados.

Conforme os dados são movidos do data lake para instâncias de computação, seja para exploração ou treinamento, certifique-se de que o acesso às instâncias de computação de destino também seja rigorosamente controlado. Mais uma vez, criptografe dados em trânsito e em repouso na infraestrutura de computação.

Durante as fases de preparação de dados e engenharia de recursos, há várias opções para a exploração segura de dados na AWS. É possível explorar os dados em um ambiente gerenciado de bloco de anotações hospedado pela Amazon SageMaker ou em um bloco de anotações Amazon EMR. Você também pode usar serviços gerenciados, como o Amazon Athena e o AWS Glue, para explorar os dados sem movê-los para fora do data lake no Amazon S3. Também é possível usar uma combinação das duas abordagens. Use um bloco de anotações Jupyter hospedado em uma instância de bloco de anotações do Amazon SageMaker para explorar, visualizar e fazer a engenharia de recursos em um pequeno subconjunto de dados. Em seguida, escale a engenharia de recursos usando um serviço de ETL gerenciado, como o Amazon EMR ou o AWS Glue.

Ao usar um bloco de anotações Jupyter hospedado em uma instância de bloco de anotações do Amazon SageMaker, implante a instância de bloco de anotações em uma Amazon VPC, o que permite usar controles de nível de rede para limitar a comunicação com o bloco de anotações hospedado. Além disso, é possível capturar as chamadas de rede para dentro e para fora da instância de bloco de anotações em

logs de fluxo da VPC a fim de permitir visibilidade e controle adicionais da rede. Ao implantar o bloco de anotações em sua VPC, você também poderá consultar fontes de dados e sistemas acessíveis de dentro da sua VPC, como bancos de dados relacionais nos data warehouses do Amazon RDS ou do Amazon Redshift. Usando o IAM, você pode restringir ainda mais o acesso à interface do usuário baseada na Web da instância de bloco de anotações, de modo que ela só possa ser acessada de dentro da VPC.

Para se comunicar com dados armazenados no data lake no Amazon S3 a partir da instância de bloco de anotações dentro da sua VPC, use a conectividade do endpoint da interface VPC. Isso garante que a comunicação entre sua instância de bloco de anotações e o Amazon S3 seja realizada de maneira completa e segura dentro da rede da AWS. Criptografe os dados em repouso nas instâncias de bloco de anotações criptografando os volumes do EBS anexados à instância de bloco de anotações do Amazon SageMaker usando uma chave gerenciada pelo AWS KMS.

O servidor de bloco de anotações Jupyter fornece acesso baseado na Web ao sistema operacional subjacente, o que proporciona aos desenvolvedores e cientistas de dados a capacidade de instalar pacotes adicionais de software ou kernels Jupyter para personalizar o ambiente. Por padrão, um usuário tem permissões para assumir permissões raiz locais, dando a ele controle total sobre a instância do EC2 subjacente. É possível restringir esse acesso para eliminar a capacidade do usuário de assumir permissões de root, mas ainda permitir que ele tenha controle sobre o ambiente do usuário local.

Além de restringir o acesso a permissões raiz, use as configurações de ciclo de vida para gerenciar instâncias de bloco de anotações Jupyter. As configurações de ciclo de vida são scripts de shell executados como root quando a instância de bloco de anotações é criada pela primeira vez ou quando a instância de bloco de anotações está sendo iniciada. Elas permitem que você instale ferramentas personalizadas, pacotes ou monitoramento. As configurações de ciclo de vida podem ser alteradas e reutilizadas em várias instâncias de bloco de anotações para que você possa fazer uma alteração uma vez e aplicar a nova configuração a instâncias de bloco de anotações gerenciadas reiniciando-as. Isso permite que as equipes de TI, operações e segurança tenham o controle necessário enquanto viabilizam as necessidades dos seus desenvolvedores e cientistas de dados.

Ao treinar um modelo, muitas vezes é necessário ter mais poder computacional do que o que uma só instância de bloco de anotações pode fornecer. Na AWS, você pode usar o Amazon SageMaker para treinar modelos em um cluster de instâncias de treinamento. O Amazon SageMaker fornece a infraestrutura subjacente usada para executar seus trabalhos de treinamento, rodando seus algoritmos em relação aos seus dados para produzir um modelo treinado.

Inicie seu cluster de instâncias de treinamento em sua VPC, o que permite aplicar controles de rede a instâncias de treinamento e conceder acesso a serviços da AWS, inclusive Amazon S3 e Amazon ECR, por meio de VPC endpoints. Usando grupos de segurança, restrinja o acesso aos trabalhos de treinamento a fontes de dados não hospedadas em produtos da AWS dentro da VPC. Além disso, controle o acesso à rede além da VPC, usando servidores de proxy e grupos de segurança. Criptografe dados nos volumes do EBS do seu nó de treinamento usando chaves de criptografia gerenciadas pelo KMS para fornecer proteção adicional para dados confidenciais durante o treinamento. Use VPC endpoints de plano de controle do Amazon SageMaker para permitir a comunicação privada entre sua VPC e o plano de controle do Amazon SageMaker a fim de gerenciar e monitorar trabalhos de treinamento.

Ao treinar seu modelo em um cluster de instâncias, não esqueça de também acomodar informações trocadas por algoritmos durante esse processo. É comum que estruturas de trabalho como o TensorFlow compartilhem informações na forma de coeficientes como parte de um trabalho de treinamento distribuído. Esses não são seus dados de treinamento; em vez disso, são as informações que os algoritmos exigem para permanecerem sincronizados entre si. Nem sempre esses dados são criptografados por padrão. Como parte de um trabalho de treinamento distribuído, configure o Amazon SageMaker para criptografar a comunicação entre os nós para o seu trabalho de treinamento. Em seguida, os dados transferidos entre esses nós são criptografados em trânsito.

Juntamente com a proteção do ambiente de bloco de anotações Jupyter hospedado e o treinamento de clusters, também é essencial proteger as implementações de algoritmo de ML. O Amazon SageMaker usa tecnologia de contêiner para treinar e hospedar algoritmos e modelos. Isso permite que o Amazon

SageMaker e outros parceiros de ML empacotem algoritmos e modelos como contêineres, que você pode usar como parte de seu projeto de ML. Além disso, você pode empacotar qualquer tecnologia, linguagem ou estrutura de trabalho para uso com o Amazon SageMaker. Ao criar seus próprios contêineres, publique-os em um registro de contêiner privado hospedado no AWS Elastic Container Repository (ECR) e criptografe contêineres hospedados no Amazon ECR em repouso usando uma chave gerenciada do KMS.

Durante o treinamento, o Amazon SageMaker recupera o contêiner que você especifica do Amazon ECR e o prepara para execução em uma instância de treinamento. Para conjuntos de dados menores, o Amazon SageMaker é compatível com o modo “File” para treinamento, que faz download dos dados de treinamento do bucket do S3 para o volume do EBS vinculado à instância de treinamento. Isso permite que o algoritmo leia seus dados de treinamento diretamente do sistema de arquivos local, sem uma integração direta ao Amazon S3. Ao usar contêineres e copiar objetos do Amazon S3, o Amazon SageMaker permite o isolamento de rede dos seus algoritmos e modelos durante o treinamento e a hospedagem.

No entanto, se você tiver grandes conjuntos de dados de treinamento, copiá-los para um sistema de arquivos local antes de iniciar um trabalho de treinamento será ineficiente. Para essa situação, use o modo “Pipe” do Amazon SageMaker, que transmite dados diretamente do Amazon S3 para a instância de treinamento. Isso significa que os trabalhos de treinamento começam mais cedo, terminam mais rapidamente e precisam de menos espaço em disco, reduzindo o custo geral para treinar modelos de ML no Amazon SageMaker.

Os logs gerados durante o treinamento pelo Amazon SageMaker são gravados no AWS CloudWatch Logs. Use uma chave de criptografia gerenciada pelo AWS KMS para criptografar os dados de log ingeridos pelo AWS CloudWatch Logs.

A proteção de dados é importante não somente para os dados de treinamento, mas também para os dados de produção/ativos usados para inferência. Um exemplo é uma chamada de API de inferência feita para um serviço de IA da AWS ou para um endpoint de modelo hospedado na Amazon. As solicitações HTTPS para essas chamadas de API devem ser assinadas, de modo que seja possível verificar a identidade do solicitante e proteger os dados da carga da solicitação em trânsito e contra possíveis ataques de reprodução. Quando você usa a Interface de linhas de comando da AWS (CLI da AWS) ou um dos SDKs da AWS para fazer a chamada de API, essas ferramentas assinam automaticamente as solicitações para você com a chave de acesso especificada quando você configurou as ferramentas. No entanto, se você escrever um código personalizado para enviar solicitações HTTPS para a AWS, é necessário implementar a funcionalidade para assinar as solicitações.

Além disso, os produtos de IA da AWS, como o Amazon Translate e o Amazon Comprehend, têm provisões para usar seus dados para desenvolvimento e aprimoramento contínuos da AWS e das tecnologias de ML e IA afiliadas. Você pode optar por não ter seus dados usados para essas finalidades entrando em contato com o AWS Support. Após receber a confirmação de que sua conta teve a adesão removida e seguir qualquer instrução adicional fornecida, seu conteúdo não será mais armazenado ou usado para desenvolver ou aprimorar o produto de IA da AWS ou qualquer tecnologia de Amazon ML/IA.

MLSEC 03: como você está protegendo os modelos de ML treinados?

Além de proteger os dados usados para treinar um modelo de ML, proteja o acesso ao artefato de modelo gerado pelo processo de treinamento. Hospede seu modelo para que um consumidor do modelo possa executar inferência nele com segurança. Os consumidores de um modelo de ML, que podem ser aplicações ou usuários internos ou externos, normalmente se integram a ele por meio de um endpoint simples ou API capaz de fornecer previsões.

Na AWS, normalmente um modelo de ML gerado no final da fase de treinamento é persistido no Amazon S3. Faça upload de modelos treinados em sua VPC para o Amazon S3 usando um VPC endpoint privado. Isso garante que o modelo seja transferido para o Amazon S3 com segurança dentro da rede da AWS. Quando um modelo é treinado usando o Amazon SageMaker, o serviço criptografa os artefatos de modelo e outros artefatos de sistema em trânsito e em repouso.

O Amazon SageMaker implanta e hospeda um modelo treinado em um cluster de nós de computação de inferência e fornece um endpoint (URL HTTPS) que serve de base para a realização de inferências. Endpoints hospedados no Amazon SageMaker são compatíveis com inferências em tempo real e previsões de transformação em lote. Em ambos os casos, os endpoints hospedados possibilitam as mesmas proteções de rede baseadas em VPC, isolamento de rede do contêiner que hospeda o modelo e criptografia dos volumes do EBS do nó de inferência.

Os endpoints hospedados do Amazon SageMaker oferecem segurança adicional para proteger seus modelos e invocações usando o IAM. Isso permite que você controle quais usuários do IAM, funções do IAM, VPCs de origem ou IPs podem executar inferência em relação ao modelo. Além disso, você pode usar o AWS PrivateLink para compartilhar com segurança seu modelo como um serviço para outros consumidores.

Assim como acontece para os logs capturados como parte do treinamento, o Amazon SageMaker registra em log as atividades de inferência de modelo no AWS CloudWatch Logs. Novamente, certifique-se de que os logs ingeridos pelo AWS CloudWatch Logs sejam criptografados usando uma chave de criptografia gerenciada pelo KMS. Isso fornece um log da atividade de seus modelos durante a inferência e permite que você disponibilize todos os detalhes necessários para atender aos requisitos de segurança e auditabilidade.

Frequentemente os consumidores de um modelo de ML fazem previsões com base no modelo a partir de aplicações externas ao ambiente que hospeda o modelo, por exemplo, uma aplicação Web pode fazer inferências em um endpoint com acesso à Internet. O diagrama a seguir mostra uma arquitetura sem servidor para acessar um modelo hospedado no Amazon SageMaker. Nessa arquitetura, um API Gateway é acessado diretamente por usuários finais, enquanto o AWS Lambda e o endpoint modelo do Amazon SageMaker são operados em uma rede privada protegida.

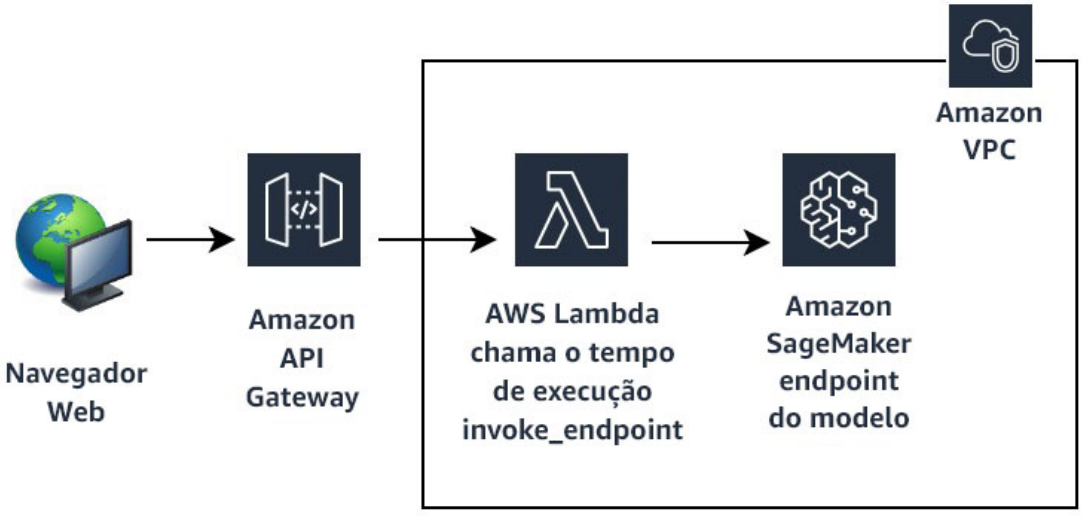

Figura 14 – Arquitetura sem servidor para inferência.

As etapas de alto nível nessa arquitetura são:

1. Uma aplicação consumidora invoca a API do API Gateway com valores de parâmetro de solicitação.
2. O API Gateway repassa os valores de parâmetro para a função Lambda. A função Lambda analisa o valor e o envia para o endpoint modelo do Amazon SageMaker.
3. O modelo executa a previsão e retorna o valor previsto para o AWS Lambda. A função do Lambda analisa o valor retornado e o envia de volta para o API Gateway.
4. O API Gateway responde ao cliente com o valor de inferência.

Para o caso de uso completo atendido por essa arquitetura, consulte Chamar um endpoint modelo do Amazon SageMaker usando o Amazon API Gateway e o AWS Lambda.

Resposta a incidentes

Consulte o whitepaper do AWS Well-Architected Framework para conhecer as melhores práticas na área de resposta a incidentes para segurança aplicáveis a ML.

Principais produtos da AWS

Os principais serviços da AWS para segurança e monitoramento de dados na AWS são:

- AWS IAM
- Amazon Virtual Private Cloud (Amazon VPC) e VPC endpoints
- Amazon SageMaker

Recursos

Consulte os recursos a seguir para saber mais sobre nossas melhores práticas de segurança na AWS.

Whitepapers

- Melhores práticas de segurança da AWS
- Como criar soluções de armazenamento de big data (data lakes) para o máximo de flexibilidade

Documentação e blogs

- Melhores práticas de segurança de criação de código da OWASP
- Autenticação e controle de acesso para Amazon SageMaker
- Chamar um endpoint modelo do Amazon SageMaker usando o Amazon API Gateway e o AWS Lambda
- Criar um endpoint sem servidor para um endpoint Amazon SageMaker

Pilar Confiabilidade

O pilar Confiabilidade inclui a capacidade de um sistema se recuperar de interrupções de infraestrutura ou de serviço, adquirir dinamicamente recursos de computação para atender à demanda e mitigar transtornos, como configurações incorretas ou problemas temporários de rede.

Tópicos

Princípios do design

Na nuvem, há vários princípios que podem ajudar você a fortalecer a confiabilidade do seu sistema Para práticas padrão, consulte o whitepaper Pilar Confiabilidade: AWS Well-Architected Framework. Além

disso, também existem princípios projetados para ajudar a aumentar a confiabilidade especificamente para cargas de trabalho de ML:

- Gerenciar alterações nas entradas do modelo por meio de automação: visando ter a capacidade de recriar a versão exata de um modelo em caso de falha ou erro humano, as cargas de trabalho de ML têm requisitos adicionais para gerenciar alterações nos dados que são usados para treinar um modelo. O gerenciamento de versões e alterações por meio da automação fornece um método de recuperação confiável e consistente.
- Treinar uma vez e implantar em ambientes: ao implantar a mesma versão de um modelo de ML em várias contas ou ambientes, a mesma prática de criação única aplicada ao código do aplicativo deve ser aplicada para o treinamento de modelo. Uma versão específica de um modelo só deve ser treinada uma vez e os artefatos do modelo de saída devem ser usados para implantar em vários ambientes, a fim de evitar a introdução de alterações inesperadas no modelo entre ambientes.

Melhores práticas

Há três áreas de melhores práticas para confiabilidade na nuvem.

Tópicos

Fundamentos

Não há práticas básicas exclusivas para cargas de trabalho de ML que pertencem a esta subseção. As práticas identificadas no whitepaper Pilar Confiabilidade do AWS Well-Architected Framework devem ser usadas para garantir capacidades básicas.

Gerenciamento de alterações

MLREL 01: como gerenciar alterações em seus modelos de machine learning e endpoints de previsão?

Para cargas de trabalho de ML, é importante criar um mecanismo para permitir a rastreabilidade de alterações feitas no modelo, bem como alterações em endpoints de previsão. Isso permite agilizar a solução de problemas e a reversão para uma versão de modelo anterior se um modelo mais recente não apresentar o desempenho esperado. Uma versão implantada de um modelo deve ser rastreável até um artefato de modelo versionado específico, protegido em um repositório de artefatos com acesso somente leitura a recursos limitados. Retenha artefatos de modelo usando um período de retenção definido pela empresa.

Para reduzir a sobrecarga e a intervenção manual, deve-se automatizar as alterações em um modelo ou endpoint por meio de um pipeline que inclua a integração com qualquer sistema de rastreamento de gerenciamento de alterações conforme exigido pelos negócios. A inclusão da capacidade de rastreamento por meio de entradas e artefatos de pipeline versionados permite rastrear alterações e reverter automaticamente após uma alteração com falha.

Para implantar alterações em um modelo, recomenda-se usar estratégias padrão de teste A/B, no qual uma parte definida do tráfego é direcionada para o novo modelo enquanto direciona o tráfego restante para o modelo antigo. Nesse caso, a reversão inclui uma alteração de DNS de volta para a versão mais antiga. Para identificar efetivamente quando uma reversão ou postergação é necessária, as métricas que

avaliam a performance do modelo devem ser implementadas para alertar quando ações de reversão ou postergação são necessárias. Ao projetar a arquitetura para reversão ou postergação, é fundamental avaliar o seguinte para cada modelo:

- Onde o artefato de modelo é armazenado?
- Os artefatos de modelo são versionados?
- Quais alterações estão incluídas em cada versão?
- Para um endpoint implantado, qual versão do modelo é implantada?

A criação de mecanismos de rastreabilidade para rastrear recursos e implantar automaticamente seus modelos proporciona funcionalidades confiáveis de reversão e recuperação. Além disso, para garantir que um modelo possa ser implantado de maneira confiável em vários ambientes, uma estratégia de treinamento único deve ser usada para reduzir qualquer variabilidade acidental no processo de implantação.

Ao criar modelos na AWS, recomenda-se usar as funcionalidades existentes do serviço, bem como implementar padrões que garantam que os modelos de ML possam ser restaurados para uma versão anterior.

Para serviços de IA na AWS, como o Amazon Transcribe, a AWS executa controle de versão nos endpoints implantados que são usados para fazer previsões de endpoints. A AWS é responsável pelo gerenciamento de alterações relacionado à hospedagem do endpoint como um serviço.

Existem padrões comuns para serviços de ML na AWS e estruturas de trabalho e interfaces de ML na AWS para fornecer rastreabilidade no gerenciamento de alterações, bem como garantir recursos para operações de roll forward e reversão. Armazene artefatos de modelo como objetos versionados no Amazon S3 para garantir a durabilidade e a disponibilidade do modelo. Certifique-se de que as imagens de contêiner usadas para treinamento de modelo e hospedagem de modelo sejam armazenadas em um repositório durável e seguro de imagens, como o AWS Elastic Container Registry (ECR). Além disso, proteja a integridade de artefatos de modelo e imagens de contêiner ao restringir o acesso a artefatos de modelo usando o acesso baseado em função do IAM e implementando privilégios mínimos em políticas aplicadas a recursos. Armazene todas as configurações usadas para criar um artefato como código em um sistema gerenciado de controle de origem, como o AWS CodeCommit.

Além disso, ter rastreabilidade dos artefatos permite que você faça operações de roll forward ou reversão para uma versão específica. Crie e mantenha um manifesto do histórico de versões de artefatos de modelo destacando as alterações implantadas entre as versões de artefatos de modelo. É possível fazer isso armazenando dados de manifesto de alterações em um armazenamento persistente ou, de maneira ideal, por meio de um pipeline de implantação automatizado que controle o desenvolvimento e a implantação completos de modelos, conforme descrito posteriormente em Gerenciamento de falhas. Quando possível, use as funcionalidades de teste A/B nativos do SageMaker por meio de variantes de produção para avaliar e reagir rapidamente a alterações em vários modelos.

Para estruturas de trabalho e interfaces de ML na AWS, há várias capacidades disponíveis para criar padrões reutilizáveis de design para aumentar a automação e a capacidade de recuperação de cargas de trabalho na forma de modelos do AWS CloudFormation e ferramentas de desenvolvedor da AWS. Crie uma estratégia de implantação compatível com funcionalidades de operações de roll forward e reversão conforme descrito no whitepaper Pilar Confiabilidade do AWS Well-Architected Framework para avaliar e reagir rapidamente a alterações em vários modelos.

A implementação de capacidades de operações automáticas de roll forward e reversão permite que você se recupere de uma alteração defeituosa, falha do sistema ou degradação de performance do modelo. Essa capacidade requer uma estratégia bem definida de versionamento combinada a um mecanismo para rastrear e reverter alterações mediante a detecção de problemas. Certifique-se de que todas as métricas críticas para a avaliação do modelo sejam definidas e coletadas. Colete métricas em um sistema de monitoramento, como o Amazon CloudWatch, com alarmes definidos para acionar eventos de reversão caso uma versão de modelo não esteja funcionando conforme o esperado.

MLREL 02: como as alterações nos modelos de ML são coordenadas em sua carga de trabalho?

Para garantir que as alterações em um modelo de ML sejam introduzidas sem interrupção ou com o mínimo de interrupção dos recursos de carga de trabalho existentes, é importante acomodar como sistemas e aplicações dependentes se integrarão ao projetar aplicativos de interface. O design flexível de aplicações e APIs ajuda a abstrair mudanças de aplicações de interface. Além disso, ter uma estratégia para comunicar e coordenar alterações em sistemas e/ou aplicativos dependentes é essencial. Siga uma estratégia definida de gerenciamento de alterações para introduzir alterações e comunicá-las às equipes afetadas e permitir a rastreabilidade dessas alterações.

Gerencie a implantação de novas versões de modelo da mesma forma que as alterações no nível do aplicativo são regidas e controladas. Uma estratégia de gerenciamento de alterações para modelos de ML deve considerar como as alterações são comunicadas e implantadas para ajudar a evitar interrupções no serviço e degradar a performance e a precisão do modelo. Sua estratégia para implantar novas versões de modelo também deve incluir atividades de validação a serem executadas antes e depois da implantação em um ambiente de destino.

Para garantir que o gerenciamento de alterações seja executado de forma consistente e correta, é uma prática recomendada executar todas as alterações por meio de um pipeline de CI/CD com controles de acesso que seguem o princípio do privilégio mínimo para impor seu processo de implantação. O controle de implantações por meio da automação combinada com pontos manuais ou automatizados de controle de qualidade garante que as alterações possam ser efetivamente validadas nos sistemas dependentes antes da implantação.

MLREL 03: como você está gerenciando a escalabilidade de endpoints que hospedam modelos para previsões?

É essencial implementar capacidades que permitam escalar automaticamente seus endpoints de modelo. Isso garante que você possa processar previsões de maneira confiável para atender às demandas dinâmicas da carga de trabalho. Para escalar seus endpoints, você deve incluir monitoramento de endpoints para identificar um limite que aciona a adição ou a remoção de recursos para oferecer suporte à demanda atual. Assim que um trigger de escalabilidade é recebido, uma solução deve ser implementada para escalar recursos de back-end que oferecem suporte a esse endpoint. Execute testes de carga em endpoints para poder validar a capacidade deles de escalar com eficiência e fornecer previsões de maneira confiável.

Na AWS, a escalabilidade e a responsabilidade de endpoint nessa área dependem do serviço de IA/ML utilizado. Para serviços de IA, incluindo Amazon Comprehend, Amazon Polly e Amazon Translate, os endpoints são gerenciados e dimensionados automaticamente pela AWS. Para serviços de ML da AWS, como o Amazon SageMaker, os recursos de escalabilidade automática em zonas de disponibilidade são configuráveis dentro do serviço. Para alta disponibilidade, configure a escalabilidade horizontal automática em várias zonas de disponibilidade para todas as variantes de produção. Uma vez configurado, é crucial realizar testes de falha para garantir que o endpoint possa se recuperar de falhas e atender aos requisitos de disponibilidade.

Para estruturas de trabalho e interfaces de ML da AWS, configure capacidades com escalabilidade e balanceamento de carga automáticos, independentemente de o modelo estar sendo hospedado em instâncias do EC2 ou usando contêineres no ECS ou no EKS que estejam hospedados em instâncias do EC2 ou no AWS Fargate. Também é possível usar as capacidades de escalabilidade automática para recuperar automaticamente a integridade de instâncias do EC2 substituindo uma instância não íntegra. Consulte o pilar Confiabilidade no AWS Well Architected Framework para conhecer as melhores práticas padrão nessa área.

Gerenciamento de falhas

MLREL 04: como recuperar-se de falhas ou perda acidental de um modelo de ML treinado?

Um modelo de ML treinado é um artefato empacotado que deve ser recuperável em caso de falha ou perda. Uma falha ou perda de recursos pode ser causada por vários eventos que variam de falha do sistema a erro humano. Para modelos de ML, o seguinte cenário de falha deve ser considerado e comparado com seus objetivos de recuperação para garantir que a estratégia apropriada seja implantada. Se um artefato de modelo for excluído acidentalmente devido a um erro humano ou se o armazenamento subjacente ficar indisponível, será possível recuperar ou recriar esse artefato facilmente?

É possível obter a capacidade de proteger um artefato de modelo contra exclusão acidental ao garantir que o artefato de modelo seja protegido, permitindo apenas privilégios mínimos necessários para usar o artefato, implementando mecanismos adicionais como MFA para exclusão por usuários privilegiados e armazenando uma cópia secundária do artefato conforme exigido por sua estratégia definida de recuperação de desastres. Em paralelo, a implementação de uma estratégia de versionamento de artefatos permite a recuperação do artefato versionado específico. Em segundo lugar, a capacidade de recriar uma versão específica de um artefato de modelo fornece proteção adicional contra falhas ou perdas. Aplique os mesmos mecanismos de proteção e estratégia de versionamento a entradas de modelo, incluindo dados e código de treinamento.

Na AWS, as melhores práticas relacionadas à falha e à capacidade de recuperação do modelo variam de acordo com o produto da AWS usado. Os serviços de IA da AWS, como o Amazon Polly, usam modelos predefinidos que são protegidos e gerenciados pela AWS. Os serviços de ML da AWS, como o Amazon SageMaker, criam e armazenam artefatos de modelo no Amazon S3. Nesse caso, você pode aproveitar os controles de acesso fornecidos pelo AWS IAM para proteger entradas e artefatos de modelo visando a proteção de recursos. Além disso, você pode usar um mecanismo, como o versionamento do Amazon S3 combinado com a marcação de objetos para versionamento e rastreabilidade de artefatos de modelo, para recuperação em caso de falha.

MLREL 05: como recuperar-se de falhas ou perda acidental de recursos de hospedagem de modelos?

Ter a capacidade de recuperar qualquer componente em uma carga de trabalho de ML garante que a solução seja capaz de resistir a falhas ou perdas de um recurso. Uma falha ou perda de recursos pode ser causada por vários eventos que variam de falha do sistema a erro humano. Para cargas de trabalho de ML, os seguintes cenários de falha devem ser considerados e comparados com seus objetivos de recuperação para garantir que a estratégia apropriada seja implantada. Se um endpoint de modelo for excluído acidentalmente, será possível recriar esse endpoint para recuperá-lo em uma versão específica?

Na AWS, as melhores práticas relacionadas ao gerenciamento de falhas variam de acordo com o produto da AWS usado. Os serviços de IA da AWS, como o Amazon Polly, são hospedados, escalados e gerenciados pela AWS para que você não seja responsável pela recuperação de endpoints. Uma melhor prática para os serviços de ML da AWS, bem como para a infraestrutura e as estruturas de trabalho de ML da AWS, é garantir que um endpoint responsável por hospedar previsões de modelo seja totalmente recuperável para uma versão ou point-in-time específico, conforme definido por seus negócios. A capacidade de recuperar um endpoint de modelo exige que todos os componentes e configurações usados para criar esse endpoint sejam incluídos em uma estratégia de controle de versão gerenciada para habilitar a recuperação completa mediante a indisponibilidade de qualquer componente.

Como exemplo, recriar um endpoint no SageMaker requer versões exatas de vários componentes no SageMaker, entre eles: artefatos de modelo, imagens de contêiner e configurações de endpoint. Para recriar uma versão específica de artefato de modelo, você também precisa saber a estratégia de

versionamento para os dados de treinamento e o algoritmo usado para criar esse artefato de modelo. Para garantir que você tenha a capacidade de recriar qualquer componente no pipeline em caso de falha, faça a versão de todos os recursos dependentes também. Além do versionamento, é fundamental garantir que todos os artefatos versionados sejam incluídos em um manifesto documentando a implantação e que todos os artefatos versionados sejam protegidos usando o princípio do privilégio mínimo, conforme descrito no pilar Segurança.

Recursos

Recursos

Consulte os recursos a seguir para saber mais sobre as práticas recomendadas para confiabilidade.

Whitepapers

- Pilar Confiabilidade – AWS Well-Architected Framework

Pilar Eficiência de performance

O pilar Eficiência de performance concentra-se no uso eficiente de recursos de computação para cumprir os requisitos e em como manter essa eficiência conforme a demanda muda e as tecnologias evoluem.

Tópicos

Princípios do design

Na nuvem, há vários princípios que podem ajudar você a fortalecer a eficiência de performance do seu sistema. Para práticas padrão, consulte o whitepaper Pilar Eficiência de performance: AWS Well-Architected Framework. Além disso, há vários princípios desenvolvidos para ajudar a aumentar a eficiência de performance especificamente para cargas de trabalho de ML:

- Otimizar a computação para sua carga de trabalho de ML: a maioria das cargas de trabalho de ML exige muita computação, pois é necessário realizar grandes quantidades de multiplicações e adições vetoriais em uma infinidade de dados e parâmetros. Especialmente no aprendizado profundo, há a necessidade de escalar para chipsets que fornecem maior profundidade de fila, maiores unidades lógicas aritméticas e contagens de registros, para permitir processamento paralelo massivo. Por isso, as GPUs são o tipo de processador preferencial para treinar um modelo de aprendizagem profunda. Discutiremos os detalhes relativos à seleção do recurso computacional apropriado na seção MLPER 01 abaixo.
- Definir os requisitos de performance de latência e largura de banda de rede para seus modelos: algumas de suas aplicações de ML podem exigir resultados de inferência quase instantâneos para atender aos seus requisitos empresariais. A oferta da menor latência possível pode exigir a remoção de viagens de ida e volta dispendiosas para os endpoints de API mais próximos. Essa redução na latência pode ser obtida executando a inferência diretamente no próprio dispositivo. Isso é conhecido como machine learning na borda. Um caso de uso comum para esse requisito é a manutenção preditiva em fábricas. Essa forma de baixa latência e inferência quase em tempo real na borda permite indicações antecipadas de falha, o que pode mitigar reparos dispendiosos de máquinas antes que a falha realmente ocorra.
- Monitorar e mensurar continuamente a performance do sistema: a prática de identificar e coletar regularmente métricas-chave relacionadas à criação, treinamento, hospedagem e execução de

previsões em relação a um modelo garante que você seja capaz de monitorar continuamente o sucesso holístico entre os principais critérios de avaliação. Para validar os recursos do sistema usados para oferecer suporte às fases de cargas de trabalho de ML, é fundamental coletar e monitorar continuamente recursos do sistema, como computação, memória e rede. Os requisitos para cargas de trabalho de ML mudam em fases diferentes, pois os trabalhos de treinamento usam mais memória, enquanto os trabalhos de inferência usam mais computação, conforme discutido nos dois princípios de design anteriores.

Melhores práticas

Há quatro áreas de melhores práticas para eficiência de performance na nuvem:

- Seleção (computação, armazenamento, banco de dados, rede)
- Análise
- Monitoramento
- Concessões

Adote uma abordagem orientada por dados para selecionar uma arquitetura de alta performance. Reúna dados sobre todos os aspectos da arquitetura, desde o design de alto nível até a seleção e a configuração dos tipos de recursos. Ao avaliar suas escolhas de forma cíclica, você pode garantir que está aproveitando a evolução contínua dos serviços da AWS. O monitoramento garantirá que você esteja ciente de qualquer desvio da performance esperado e poderá tomar medidas. Por fim, sua arquitetura pode fazer concessões para melhorar a performance, como usar compactação ou armazenamento em cache ou relaxar os requisitos de consistência.

Seleção

MLPER 01: como você está escolhendo o tipo de instância mais adequado para treinar e hospedar seus modelos?

Um pipeline habitual de machine learning é composto por uma série de etapas. O processo começa com a coleta de dados de treinamento e teste, seguido pela engenharia de recursos e transformação dos dados coletados. Após a fase inicial de preparação de dados, treinar os modelos de ML, bem como avaliá-los e ajustá-los, leva ao estágio final de implantação, fornecimento e monitoramento da performance dos modelos durante todo o ciclo de vida. A performance da carga de trabalho de ML em cada uma dessas fases deve ser cuidadosamente considerada, pois as necessidades de computação de cada fase são diferentes. Por exemplo, embora você possa precisar de um cluster potente de instâncias de GPU para treinamento de modelo, quando se trata de inferência, um cluster de instâncias de CPU com escalabilidade automática pode ser suficiente para atender aos seus requisitos de performance.

O tamanho dos dados, o tipo de dados e a seleção do algoritmo podem ter um efeito perceptível sobre qual configuração é mais eficaz. Ao treinar o mesmo modelo repetidamente, é altamente recomendável executar testes iniciais em um espectro de tipos de instância para descobrir configurações com boa performance e economia. Como diretriz geral, as instâncias de GPU são recomendadas para a maioria das finalidades de aprendizagem profunda, pois o treinamento de novos modelos é mais rápido em uma instância de GPU do que em uma instância de CPU. Você pode escalar de forma sublinear quando tiver instâncias de várias GPUs ou se usar treinamento distribuído em várias instâncias com GPUs. No entanto, é importante observar que os algoritmos que treinam com mais eficiência em GPUs podem não exigir necessariamente GPUs para inferência eficiente.

A AWS fornece uma seleção de tipos de instância otimizados para atender a diferentes casos de uso de machine learning (ML). Os tipos de instância têm combinações variáveis de CPU, GPU, FPGA, memória,

armazenamento e capacidade de rede. Além disso, você pode vincular um acelerador de inferência habilitado por GPU às suas instâncias do Amazon EC2 ou do Amazon SageMaker por meio do Amazon Elastic Inference ou usar instâncias do Amazon EC2 com tecnologia do AWS Inferentia, um chip de inferência de ML de alta performance, projetado de modo personalizado pela AWS. Isso proporciona a flexibilidade de escolher a combinação adequada de recursos otimizados para atender aos seus casos de uso de ML, esteja treinando modelos ou executando inferência em modelos treinados. Cada tipo de instância inclui um ou mais tamanhos de instância, permitindo escalar seus recursos de acordo com os requisitos da carga de trabalho de destino.

Por fim, alguns algoritmos, como XGBoost, implementam um algoritmo de código aberto que foi otimizado para computações de CPU, enquanto na AWS Deep Learning AMI (DLAMI), algumas estruturas de trabalho, como Caffe, funcionam apenas com suporte a GPU e não podem ser executadas no modo CPU.

Independentemente da seleção de instância para treinamento e hospedagem, teste a carga da sua instância ou dos endpoints do Amazon SageMaker para determinar o pico de carga que sua instância ou endpoint pode suportar, e a latência das solicitações conforme a simultaneidade aumenta.

MLPER 02: como escalar a carga de trabalho de ML mantendo a performance ideal?

Ao avaliar como escalar sua arquitetura de ML para um aumento na demanda e para a performance ideal, é importante diferenciar entre a implantação de seus modelos por meio da experiência de serviço gerenciado em relação à implantação e o gerenciamento de modelos de ML por conta própria.

Na AWS, embora a experiência de ML gerenciada seja fornecida pelo Amazon SageMaker, normalmente você usa AMIs do Deep Learning (DLAMI), que fornecem estruturas de trabalho MXNet, TensorFlow, Caffe, Chainer, Theano, PyTorch e CNTK, em instâncias do EC2 para gerenciar modelos por conta própria. Esta seção aborda ambos os casos de uso enquanto discute brevemente um terceiro caso de uso de serviços de IA da AWS.

O Amazon SageMaker gerencia sua infraestrutura computacional de produção em seu nome para executar verificações de integridade, aplicar patches de segurança e realizar outras manutenções de rotina, tudo isso com o monitoramento e o registro em log incorporados do Amazon CloudWatch. O Amazon SageMaker Model Monitor monitora continuamente modelos de ML em produção, detecta desvios, como desvios de dados capazes de degradar a performance do modelo ao longo do tempo, e alerta você sobre ações corretivas

Além disso, a hospedagem do Amazon SageMaker escala automaticamente a performance necessária para a aplicação usando o Application Auto Scaling. Ao usar o Application Auto Scaling, você pode ajustar automaticamente sua capacidade de inferência para manter uma performance previsível a um baixo custo. Além disso, ao modificar a configuração do endpoint, você pode alterar manualmente o número e o tipo de instâncias do EC2 sem incorrer em tempo de inatividade.

Para suas cargas de trabalho de aprendizado profundo, você tem a opção de escalar com o Amazon Elastic Inference (EI) para aumentar a taxa de transferência e diminuir a latência para inferências em tempo real em seus modelos de aprendizado profundo. O Amazon Elastic Inference permite que você vincule aceleração de inferência habilitada por GPU a qualquer instância do Amazon EC2. Esse recurso também está disponível para instâncias de bloco de anotações e endpoints do Amazon SageMaker, acelerando algoritmos integrados e ambientes de aprendizado profundo a custos reduzidos.

As redes neurais de aprendizado profundo são ideais para aproveitar vários processadores, distribuindo cargas de trabalho de maneira ininterrupta e eficiente entre diferentes tipos e quantidades de processadores. Com a ampla variedade de recursos sob demanda disponíveis na nuvem, você pode implantar recursos praticamente ilimitados para lidar com modelos de aprendizado profundo de qualquer porte. Ao utilizar redes distribuídas, o aprendizado profundo na nuvem permite que você projete, desenvolva e treine aplicações de aprendizado profundo com mais rapidez.

As AWS Deep Learning AMI para Ubuntu e Amazon Linux oferecem suporte ao treinamento distribuído de modelos de aprendizado profundo do TensorFlow com eficiência de escalabilidade quase linear. As AWS Deep Learning AMI vêm pré-compiladas com uma versão aprimorada do TensorFlow integrada a uma versão otimizada da estrutura de treinamento distribuída Horovod. Essa otimização resulta em implementações de alta performance que permitem que os nós se comuniquem diretamente entre si em vez de passar por um nó centralizado e gradientes intermediários usando o algoritmo ring-allreduce. Há muitas outras estruturas de trabalho que oferecem suporte ao treinamento distribuído, como o uso do Chainer ou do Keras, além da distribuição Horovod mencionada.

O uso dos serviços de IA da AWS permite adicionar inteligência aos aplicativos por meio de uma chamada de API para um serviço pré-treinado, em vez de desenvolver e treinar seus próprios modelos. A escalabilidade de uma arquitetura que usa os serviços de IA da AWS envolve o monitoramento de limites de recursos e taxas, como taxas de solicitação de APIs. Disponibilizamos informações detalhadas sobre como gerenciar limites de serviço na seção de confiabilidade do Well-Architected Framework geral.

Recursos

Consulte os seguintes recursos para saber mais sobre nossas melhores práticas para eficiência de performance.

Documentação e blogs

- Treinamento escalável de vários nós com TensorFlow
- Amazon Elastic Inference
- Teste de carga e otimização de um endpoint do Amazon SageMaker usando escalabilidade automática
- Seleção do tipo de instância para DLAMI

Whitepapers

- Pilar Performance – AWS Well Architected Framework

Pilar Otimização de custos

O pilar Otimização de custos inclui o processo contínuo de refinamento e aprimoramento de um sistema durante todo o seu ciclo de vida. Do design inicial de sua primeira prova de conceito até a operação contínua de cargas de trabalho de produção, a adoção das práticas descritas neste artigo permitirão que você crie e opere sistemas atentos ao custo que obtenham resultados empresariais e minimizem custos, possibilitando que a sua empresa maximize o retorno sobre o investimento.

Tópicos

Princípios do design

Na nuvem, há uma série de princípios que podem ajudar você a melhorar a otimização de custos do seu sistema. Para práticas padrão, consulte o whitepaper AWS Well-Architected Framework. Além disso, há

vários princípios desenvolvidos para ajudar a aumentar a otimização de custo especificamente para cargas de trabalho de ML:

- Usar serviços gerenciados para reduzir o custo de propriedade: adote serviços gerenciados adequados para cada fase da carga de trabalho de ML a fim de aproveitar o modelo “pague apenas pelo que usar”. Por exemplo, geralmente o ajuste de modelos é um processo com uso intenso de computação e tempo. Para evitar cobranças desnecessárias, use um serviço gerenciado que crie um cluster de treinamento distribuído, execute trabalhos de treinamento para ajustar o modelo, persista os modelos resultantes e desative automaticamente o cluster quando o treinamento for concluído.
- Experimentar usando conjuntos de dados pequenos: embora as cargas de trabalho de ML se beneficiem de grandes conjuntos de dados de treinamento de alta qualidade, comece com conjuntos de dados menores em uma instância de computação pequena (ou seu sistema local) para iterar rapidamente a um baixo custo. Após o período de experimentação, aumente a escala para treinar com o conjunto de dados completo disponível em um cluster de computação distribuído. Ao treinar com o conjunto de dados completo, opte por fazer streaming de dados no cluster ao invés de armazenar dados nos nós do cluster.
- Dimensionar corretamente as instâncias de treinamento e hospedagem de modelos: para treinamento e hospedagem de modelos, experimente determinar a capacidade computacional ideal necessária. Recomenda-se que você comece com instâncias menores e dimensione horizontalmente antes de começar a dimensionar verticalmente. Além disso, meça a diferença entre as necessidades de CPU e GPU durante o treinamento e a hospedagem. Embora alguns modelos de ML exijam instância de GPU de alta potência para treinamento, as inferências em relação ao modelo implantado podem não exigir toda a potência de uma GPU.
- Fatorar a arquitetura de inferência baseada em padrões de consumo: alguns modelos, como detecção de fraudes de comércio eletrônico, precisam estar constantemente disponíveis para previsões em tempo real, enquanto outros, como modelos de previsão de comércio eletrônico, podem precisar estar disponíveis apenas periodicamente. No primeiro caso, o custo de um modelo de hospedagem 24 horas por dia, 7 dias por semana é justificado, mas é possível obter economias de custos significativas no segundo caso ao implantar o modelo sob demanda, executando previsões e, em seguida, desativando o modelo.
- Definir o custo geral de ROI e de oportunidade: pondere o custo da adoção de ML em relação ao custo de oportunidade de não recorrer à transformação de ML. Recursos especializados, como o tempo de cientistas de dados ou o tempo de entrada no mercado do modelo, podem ser seus recursos mais caros e restritos. Talvez a opção de hardware mais econômica não proporcione otimização de custo se ela limitar a velocidade de experimentação e de desenvolvimento.

Melhores práticas

Há quatro áreas de melhores práticas para otimização de custo na nuvem:

Tópicos

Assim como nos outros pilares, há vantagens e desvantagens a serem avaliadas. Por exemplo, você deseja otimizar a velocidade para entrada no mercado ou o custo? Em alguns casos, é melhor otimizar a velocidade (entrar no mercado rapidamente, enviar novos recursos ou simplesmente cumprir um prazo) em vez de investir na otimização de custos inicial. Às vezes, as decisões de projeto são tomadas com base na pressa e não em dados empíricos, já que sempre existe a tentação de compensar “para garantir” em vez de gastar tempo para identificar a implantação mais econômica. Isso geralmente leva a implantações

excessivamente provisionadas e pouco otimizadas. As melhores práticas a seguir fornecem técnicas e orientações estratégicas para a otimização inicial e contínua dos custos de implantação.

Recursos econômicos

MLCOST 01: como otimizar os custos de rotulagem de dados?

A criação de um modelo de ML exige que desenvolvedores e cientistas de dados preparem seus conjuntos de dados para treinar seus modelos de ML. Antes que os desenvolvedores possam selecionar seus algoritmos, criar seus modelos e implantá-los para fazer previsões, os classificadores humanos analisam manualmente milhares de exemplos e adicionam os rótulos necessários para treinar os modelos de ML. Esse processo é demorado e caro.

Na AWS, o Amazon SageMaker Ground Truth simplifica as tarefas de marcação de dados usando classificadores humanos por meio do Amazon Mechanical Turk, fornecedores terceirizados ou funcionários próprios. O Amazon SageMaker Ground Truth aprende com essas anotações humanas em tempo real e aplica aprendizagem ativa para marcar automaticamente boa parte do conjunto de dados restante, reduzindo a necessidade de revisão humana. A combinação de capacidades humanas e de ML permite que o Amazon SageMaker Ground Truth crie conjuntos de dados de treinamento altamente precisos, economiza tempo e complexidade, e reduz custos em comparação à anotação exclusivamente humana.

MLCOST 02: como otimizar custos durante a experimentação de ML?

Os blocos de anotações são uma maneira popular de explorar e experimentar usando dados em pequenas quantidades. A iteração com uma pequena amostra do conjunto de dados localmente, com a posterior escalabilidade para treinar no conjunto de dados completo de maneira distribuída é algo comum em machine learning.

Na AWS, as instâncias de bloco de anotações do Amazon SageMaker oferecem um ambiente Jupyter hospedado que pode ser usado para explorar pequenas amostras de dados. Interrompa as instâncias de bloco de anotações quando você não estiver usando-as ativamente. Quando possível, confirme seu trabalho, interrompa essas instâncias e reinicie-as quando precisar delas novamente. O armazenamento é persistido e você pode usar a configuração de ciclo de vida para automatizar a instalação de pacotes ou a sincronização do repositório.

Ao fazer experimentos de treinamento de um modelo, use o modo “local” do bloco de anotações do Amazon SageMaker para treinar seu modelo na própria instância de bloco de anotações, ao invés de usar um cluster de treinamento gerenciado separado. Você pode iterar e testar seu trabalho sem precisar aguardar a criação de um novo cluster de treinamento ou hospedagem a cada processo. Isso economiza tempo e custo associados à criação de um cluster de treinamento gerenciado. A experimentação também pode ocorrer fora de um bloco de anotações, por exemplo, em uma máquina local. Na sua máquina local, você pode usar o SDK do SageMaker para treinar e implantar modelos na AWS.

Ao experimentar, analise também o AWS Marketplace for Machine Learning, que oferece um catálogo cada vez maior de algoritmos e modelos de machine learning. Os modelos do AWS Marketplace são implantados diretamente no Amazon SageMaker e permitem que você crie aplicações de ML rapidamente. Isso poupa o custo e o tempo associados ao desenvolvimento do modelo.

Além disso, o AWS Marketplace for Machine Learning permite que você venda os modelos desenvolvidos, fornecendo assim um fluxo adicional de receita para monetizar modelos internos. Você pode tornar seu modelo personalizado disponível para outros clientes e, ao mesmo tempo, proteger sua propriedade intelectual. O Amazon SageMaker proporciona acesso aos seus modelos por meio de endpoints protegidos sem expor os modelos subjacentes.

MLCOST 03: como selecionar os recursos com a melhor relação entre custo e benefício para o treinamento de ML?

Quando estiver pronto para treinar um modelo de ML com os dados completos de treinamento, evite executar os trabalhos de treinamento no modo local em uma instância de bloco de anotações, a menos que o conjunto de dados seja pequeno. Em vez disso, inicie um cluster de treinamento com uma ou mais instâncias de computação para treinamento distribuído. Faça o dimensionamento correto das instâncias de computação no cluster de treinamento com base na carga de trabalho.

Na AWS, use a API de treinamento do Amazon SageMaker para criar um cluster de instâncias gerenciadas. O uso de várias instâncias no cluster de treinamento permite o treinamento distribuído, resultando em um menor tempo de treinamento. Todas as instâncias no cluster de treinamento são encerradas automaticamente quando o treinamento é concluído.

Embora disponibilizemos uma variedade de tipos de instância com diferentes configurações de capacidade para uso no treinamento, é importante dimensionar corretamente as instâncias de treinamento de acordo com o algoritmo de ML usado. Esteja ciente que talvez modelos simples não sejam treinados mais rapidamente em instâncias maiores, pois talvez eles não sejam capazes de se beneficiar do maior paralelismo de hardware. Eles podem até treinar mais lentamente devido à alta sobrecarga de comunicação da GPU. Recomenda-se que você comece com instâncias menores e dimensione horizontalmente antes de começar a dimensionar verticalmente. Além disso, se o algoritmo de ML de sua preferência for compatível com pontos de verificação, avalie o uso de treinamento de spot gerenciado com o Amazon SageMaker para economizar custos.

Além de escolher tipos de instância otimizados para treinamento, é importante selecionar versões otimizadas de estruturas de trabalho de ML para um treinamento mais rápido. A AWS fornece versões otimizadas de estruturas de trabalho, como TensorFlow, Chainer, Keras e Theano, que incluem otimizações para treinamento de alta performance em todas as famílias de instância do Amazon EC2.

Ao lidar com grandes volumes de dados de treinamento, o modo “Pipe” do Amazon SageMaker oferece uma taxa de transferência de leitura significativamente melhor do que o modo “File” do Amazon SageMaker. Enquanto o modo “File” faz download de dados para o volume local do Amazon EBS antes de iniciar o treinamento do modelo, o modo “Pipe” faz o stream de dados do Amazon S3 para o Amazon SageMaker ao treinar modelos de ML. Isso significa que os trabalhos de treinamento começam mais cedo, terminam mais rapidamente e exigem menos espaço em disco, reduzindo o custo geral para treinar modelos de ML no Amazon SageMaker.

A determinação do conjunto correto de hiperparâmetros para um modelo de ML pode ser caro. Normalmente o processo exige técnicas como pesquisa em grade ou pesquisa aleatória que envolve o treinamento de centenas de modelos diferentes. O ajuste automático de modelo do Amazon SageMaker, também conhecido como ajuste de hiperparâmetro, encontra a melhor versão de um modelo ao executar vários trabalhos de treinamento em seu conjunto de dados usando o algoritmo e os intervalos de hiperparâmetros que você especifica. Em seguida, ele escolhe os valores de hiperparâmetros que resultam em um modelo com melhor performance, conforme medido por uma métrica que você escolhe. O ajuste de hiperparâmetros usa técnicas de ML que podem determinar com rapidez e eficiência o conjunto ideal de parâmetros com um número limitado de trabalhos de treinamento.

Além disso, uma inicialização a quente de trabalhos de ajuste de hiperparâmetros pode acelerar o processo de ajuste e reduzir o custo de ajuste de modelos. A inicialização a quente de trabalhos de ajuste de hiperparâmetros elimina a necessidade de iniciar um trabalho de ajuste do zero. Em vez disso, você pode criar um novo trabalho de ajuste de hiperparâmetros com base em trabalhos pai selecionados, para que os trabalhos de treinamento realizados nesses trabalhos pai possam ser reutilizados como conhecimento anterior, reduzindo assim os custos associados ao ajuste do modelo.

Por fim, avalie o Amazon SageMaker AutoPilot, que analisa automaticamente seus dados e cria um modelo, economizando tempo e custo. O AutoPilot seleciona o melhor algoritmo na lista de algoritmos de

alta performance e testa automaticamente diferentes configurações de parâmetros nesses algoritmos para obter a melhor qualidade de modelo.

MLCOST 04: como otimizar o custo da inferência de ML?

Em linha com o treinamento do modelo, é importante compreender qual tipo de instância é ideal para sua carga de trabalho. Comece levando em consideração latência, taxa de transferência e custo. Mais uma vez, recomenda-se que você comece aos poucos e dimensione horizontalmente antes de começar a dimensionar verticalmente.

Além de usar a escalabilidade automática da instância de computação de ML para poupar custos, meça a diferença entre CPU e GPU. Embora modelos de ML de aprendizado profundo exijam instância de GPU de alta potência para treinamento, normalmente inferências em modelos de aprendizado profundo não precisam de toda a potência de uma GPU. Dessa forma, hospedar essa aprendizagem profunda em uma GPU completa pode resultar em subutilização e custos desnecessários. Além disso, considere a arquitetura de inferência necessária para o modelo de ML. Ou seja, decida se é possível implantar o modelo sob demanda, conforme os lotes de solicitações de inferência tornam-se necessários, ou se ele precisa estar disponível 24 horas por dia, 7 dias por semana para previsões em tempo real.

Na AWS, os endpoints do Amazon SageMaker oferecem suporte à escalabilidade automática, permitindo que você faça a correspondência entre oferta e demanda de recursos. Com a escalabilidade automática no Amazon SageMaker, você pode garantir a elasticidade e a disponibilidade do seu modelo, bem como otimizar o custo selecionando as métricas corretas para escalar o endpoint de inferência.

Embora a escalabilidade automática aumente ou diminua o número de instâncias por trás de um endpoint de acordo com o volume de solicitação de inferência, você também pode aumentar o poder computacional de instâncias hospedadas ao vincular uma capacidade fracionada de computação de GPU a uma instância. O Amazon Elastic Inference permite que você vincule aceleração de baixo custo habilitada por GPU a instâncias do Amazon EC2 e do Amazon SageMaker para reduzir o custo de execução de inferência de aprendizado profundo.

Embora as instâncias autônomas de GPU sejam ideais para atividades de treinamento de modelo que exigem o processamento de centenas de amostras de dados em paralelo, geralmente elas são superdimensionadas para inferência, que consome uma pequena quantidade de recursos de GPU. Modelos diferentes precisam de quantidades diferentes de GPU, CPU e recursos de memória. A seleção de um tipo de instância de GPU para atender aos requisitos do recurso mais exigente geralmente resulta na subutilização de outros recursos e em custos desnecessários.

Com o Amazon Elastic Inference, você pode escolher o tipo de instância do Amazon EC2 mais adequado às necessidades gerais de CPU e memória do seu modelo, e então configurar separadamente a quantidade de aceleração de inferência necessária para usar recursos com eficiência e reduzir o custo de execução de inferências.

Algumas aplicações não precisam de previsões online/em tempo real e são mais adequadas para previsões periódicas em lote. Nesse caso, não é necessário hospedar um endpoint 24 horas por dia, 7 dias por semana. Essas aplicações são ideais para a transformação em lote do Amazon SageMaker. Em vez de uma única inferência por solicitação, a transformação em lote gera inferências para um conjunto inteiro de dados. A transformação em lote gerencia todos os recursos computacionais necessários para executar inferências. Isso inclui executar instâncias e encerrá-las após a conclusão do trabalho de transformação em lote.

Às vezes, a execução de inferência para determinados dados exige que os dados sejam pré-processados, pós-processados ou ambos. Isso pode envolver o encadeamento de inferências de vários modelos intermediários antes que um modelo final possa gerar a inferência desejada. Nessa situação, em vez de implantar os modelos intermediários em vários endpoints, use os pipelines de inferência do Amazon SageMaker.

Um pipeline de inferência é um modelo do Amazon SageMaker composto por uma sequência linear de contêineres que processam solicitações de inferências em dados. Esses pipelines de inferência são totalmente gerenciados e podem combinar pré-processamento, previsões e pós-processamento. As invocações do modelo de pipeline são tratadas como uma sequência de solicitações HTTPS. Ao implantar todas as etapas relevantes no mesmo endpoint, você pode economizar custos e reduzir a latência de inferência.

Quando você tem vários modelos em produção, com cada modelo implantado em um endpoint diferente, seus custos de inferência aumentam proporcionalmente ao número de modelos. No entanto, se tiver um grande número de modelos semelhantes que podem ser fornecidos por meio de um contêiner compartilhado de fornecimento e não precisar acessar todos os modelos ao mesmo tempo, use a funcionalidade de endpoints multimodelo do Amazon SageMaker. Isso permite implantar vários modelos treinados em um endpoint e fornecê-los usando um único contêiner. Você pode invocar facilmente um modelo específico especificando o nome do modelo de destino como um parâmetro na solicitação de previsão. Quando há um grande rastro de modelos de ML acessados com pouca frequência, o uso de um endpoint de vários modelos pode atender eficientemente ao tráfego de inferência e permitir economias de custos significativas.

Equilíbrio entre oferta e demanda

Consulte o whitepaper do AWS Well-Architected Framework para conhecer as melhores práticas na área de equilíbrio entre oferta e demanda para otimização de custos aplicáveis a cargas de trabalho de ML.

Percepção das despesas

Consulte o whitepaper do AWS Well-Architected Framework para conhecer as melhores práticas na área de percepção das despesas para otimização de custos aplicáveis a cargas de trabalho de ML.

Otimização ao longo do tempo

Consulte o whitepaper do AWS Well-Architected Framework para conhecer as melhores práticas na área de otimização ao longo do tempo para otimização de custos aplicáveis a cargas de trabalho de ML.

Recursos

Consulte os recursos a seguir para saber mais sobre nossas melhores práticas de otimização de custos.

Documentação e blogs

- Definição de preço do Amazon SageMaker
- Use o modo local do Amazon SageMaker para treinar sua instância de bloco de anotações
- Como aproveitar ao máximo seu orçamento de Machine Learning no Amazon SageMaker
- Como diminuir o custo total de propriedade para machine learning e aumentar a produtividade com o Amazon SageMaker

Conclusão

O Machine Learning disponibiliza oportunidades inéditas para as organizações, possibilitando automação, eficiência e inovação. Neste artigo, revisitamos os cinco pilares do Well-Architected Framework sob a perspectiva do ML a fim de fornecer melhores práticas de arquitetura para criar e operar cargas de trabalho de ML confiáveis, seguras, eficientes e econômicas na Nuvem AWS. As melhores práticas são discutidas no contexto do uso de serviços de IA, serviços gerenciados de ML e estruturas de trabalho de ML na AWS, o que permite escolher entre usar a opção mais adequada para seus objetivos empresariais. Use estas melhores práticas, apresentadas como um conjunto de perguntas, para analisar suas cargas de trabalho de ML existentes ou propostas.

Ao operar cargas de trabalho de ML, não esqueça de envolver equipes interfuncionais e garantir a implementação de automação de todo o pipeline. Para obter confiabilidade, garanta respostas robustas a falhas do sistema aproveitando os recursos de autorreparação da escalabilidade automática e acompanhando todos os artefatos por meio do versionamento, de modo que um sistema funcional possa ser recriado automaticamente.

As cargas de trabalho de ML na AWS devem ser protegidas usando controles de autenticação e autorização que controlam rigorosamente quem e o que pode acessar os vários artefatos de ML. Um aplicativo seguro protegerá os ativos de informações confidenciais da sua organização e atenderá os requisitos de conformidade em cada camada. Para habilitar cargas de trabalho de ML com boa performance, você tem uma opção de tipos de instâncias de serviço otimizados para atender a diferentes requisitos de ML. Adote uma abordagem orientada a dados para fazer uma seleção de instâncias de CPU versus GPU, sem deixar de ter em mente as várias combinações de CPU, GPU, FPGA, memória, armazenamento e capacidade de rede disponíveis. Para otimização de custos, aproveite o modelo “pague apenas pelo que usar” e reduza desperdícios desnecessários ao dimensionar recursos de acordo com as demandas de dados, treinamento e inferência.

O cenário do machine learning continua a evoluir com o crescimento e amadurecimento do ecossistema de ferramentas e processos. Conforme isso ocorre, continuaremos a atualizar este artigo para ajudar você a garantir que suas aplicações de ML estejam bem arquitetadas.

Colaboradores

Os colaboradores desse documento incluem:

- Sireesha Muppala, analista sênior especialista em IA/ML, Amazon Web Services
- Shelbee Eigenbrode, arquiteta de soluções de IA/ML, Amazon Web Services
- Christian Williams, analista sênior especialista em Machine Learning, Amazon Web Services
- Bardia Nikpourian, gerente técnico de conta especialista – IA/ML, Amazon Web Services
- Ryan King, gerente técnico sênior de programa, AWS Managed Cloud, Amazon Web Services

Leitura adicional

Para mais informações, veja estas fontes:

- Gerenciamento de projetos de machine learning (whitepaper da AWS)

Revisões do documento

Para ser notificado sobre atualizações deste whitepaper, inscreva-se no feed RSS.

update-history-change	update-history-description	update-history-date
Publicação inicial (p. 59)	Publicação inicial da Lente para Machine Learning.	April 16, 2020

Avisos