# AWS 기반의 딥 러닝

안내서

2019년 8월

# 고지 사항

고객은 본 문서에 수록된 정보의 정확성을 자체적으로 평가할 책임이 있습니다. 본 문서는 (a) 정보 제공만을 위한 것이며, (b) 사전 고지 없이 변경될 수 있는 현재의 AWS 제품 제공 서비스 및 사례를 보여 주며, (c) AWS 및 자회사, 공급업체 또는 라이선스 제공자로부터 어떠한 약정 또는 보증도 하지 않습니다. AWS 제품 또는 서비스는 명시적이든 묵시적이든 어떠한 종류의 보증, 진술 또는 조건 없이 "있는 그대로" 제공됩니다. 고객에 대한 AWS의 책임과 법적 책임은 AWS 계약서에 준하며 본 문서는 AWS와 고객 간의 계약에 포함되지 않고 계약을 변경하지도 않습니다.

# 목차

# 안내서 소개

오늘날 딥 러닝은 다양한 비즈니스 분야에서 대부분의 기계 학습 구현 사례의 최일선에 활용되고 있습니다. 신경망의 매우 유연한 특성 덕분에, 이미지의 객체를 분류하거나 비디오 게임을 몇 시간 만에 마스터하는 등 다양한 작업에서 신경망이 인간의 능력을 능가하면서 가능성의 한계가 허물어지고 있습니다. 이 안내서에서는 Amazon Web Services(AWS)를 기반으로 구현되는 포괄적인 딥 러닝 프로세스를 간략하게 설명합니다. 딥 러닝 프로젝트를 실행할 때 발생하는 문제에 대해 설명하고, AWS에서 제공하는 최고 수준의 최신 기술과 인프라를 자세히 살펴보며, 그 과정에서 아키텍처 지침과 모범 사례를 제공합니다.

이 백서는 딥 러닝 연구 과학자, 딥 러닝 엔지니어, 데이터 사이언티스트, 데이터 엔지니어, 기술 제품 관리자 및 엔지니어링 리더를 대상으로 합니다.

# 개요

딥 러닝의 기본 개념은 이미 수십 년 전에 만들어졌습니다. 하지만 최근 정보의 디지털화가 급증함에 따라 조직들은 기계 학습 파이프라인에서 쉽게 사용할 수 있는 대량의 데이터를 축적했습니다.

더욱이 우리 세대는 소셜 미디어에 연결된 휴대폰과 컴퓨터 사용에 더 많은 시간을 소비하기에 더 많은 데이터가 생성되었습니다. 의학 분야에서도 X-Ray 이미지는 실제 필름 대신 디지털 레코드로 저장되고 있습니다.

한편 데이터의 양은 폭발적으로 증가했지만, 로지스틱 회귀, 의사 결정 트리, 서포트 벡터 머신 등과 같은 기계 학습 알고리즘 성능은 더 많은 데이터가 공급되어도 큰 변화가 없었습니다.

상황이 급변하면서 소규모 신경망에서 애플리케이션 정확도가 향상되었고 중간 규모의 신경망에서도 애플리케이션 정확도가 향상되었으며, 딥 러닝 신경망은 더 많은 데이터를 공급받으면서 지속적으로 정확도가 향상되었습니다. 하지만 딥 러닝 신경망에 더 많은 계층과 데이터를 도입함으로써 구현 가능한 한계까지는 아직 도달하지도 못했습니다.

신경망에 계층을 추가하고 더 많은 데이터를 공급하면 딥 러닝 애플리케이션의 정확도를 향상시킬 수 있었습니다. 하지만 딥 러닝 신경망의 훈련에는 강력하고 비용이 많이 드는 컴퓨팅 인프라에 대한 액세스가 필요하므로 이것이 극복해야 할 과제였습니다. 클라우드 컴퓨팅은 경제적이고 탄력적인 방식으로 온디맨드 GPU를 제공하여 원하는 수준의 모델 정확도를 실현하는 데 필요한 대규모 실험을 가능케 함으로써 이 문제를 해결해 주었습니다.

확률적 그래프 모델, 계획 알고리즘, 검색 알고리즘, 지식 그래프 등 광범위한 기계 학습의 범주에서 꾸준히 개선되고 있는 다른 도구가 많지만, 딥 러닝은 급속도로 향상되면서 꾸준히 새로운 지평을 열어가고 있습니다.

이 안내서에서는 딥 러닝 프로젝트를 지원하기 위해 Amazon Web Services(AWS)가 제공하는 고유한 가치 제안을 설명합니다. 딥 러닝 분야의 AWS 혁신 기술을 활용하여, AWS 최적화 컴퓨팅, 스토리지 및 네트워크 인프라를 기반으로 딥 러닝 작업의 훈련 시간을 개선할 수 있습니다. 딥 러닝 작업은 AWS 서비스를 사용하고 AWS에서 딥 러닝 인프라 및 플랫폼을 관리하는 데 수반되는 획일적인 작업의 부담을 덜어주므로 생산성과 민첩성이 향상됩니다. AWS는 딥 러닝 프로젝트의 탐색적 특성을 지원하기 위해 요구되는 가장 깊이 있고 광범위한 기능과 유연성을 제공합니다.

이 안내서에서는 딥 러닝용 AWS 서비스 사용자를 딥 러닝 엔지니어와 딥 러닝 과학자로 지칭합니다. 딥 러닝 프로젝트 실행시, 딥 러닝 엔지니어와 딥 러닝 사이언티스트는 직책명은 다르지만 하나의 팀으로 구성합니다.

## 딥 러닝 환경

다음 그림은 이 안내서에서 다루는 딥 러닝 환경의 범위를 시각적으로 보여 줍니다. 다이어그램의 여러 부분에 대한 자세한 설명은 다음 표를 참조하십시오.

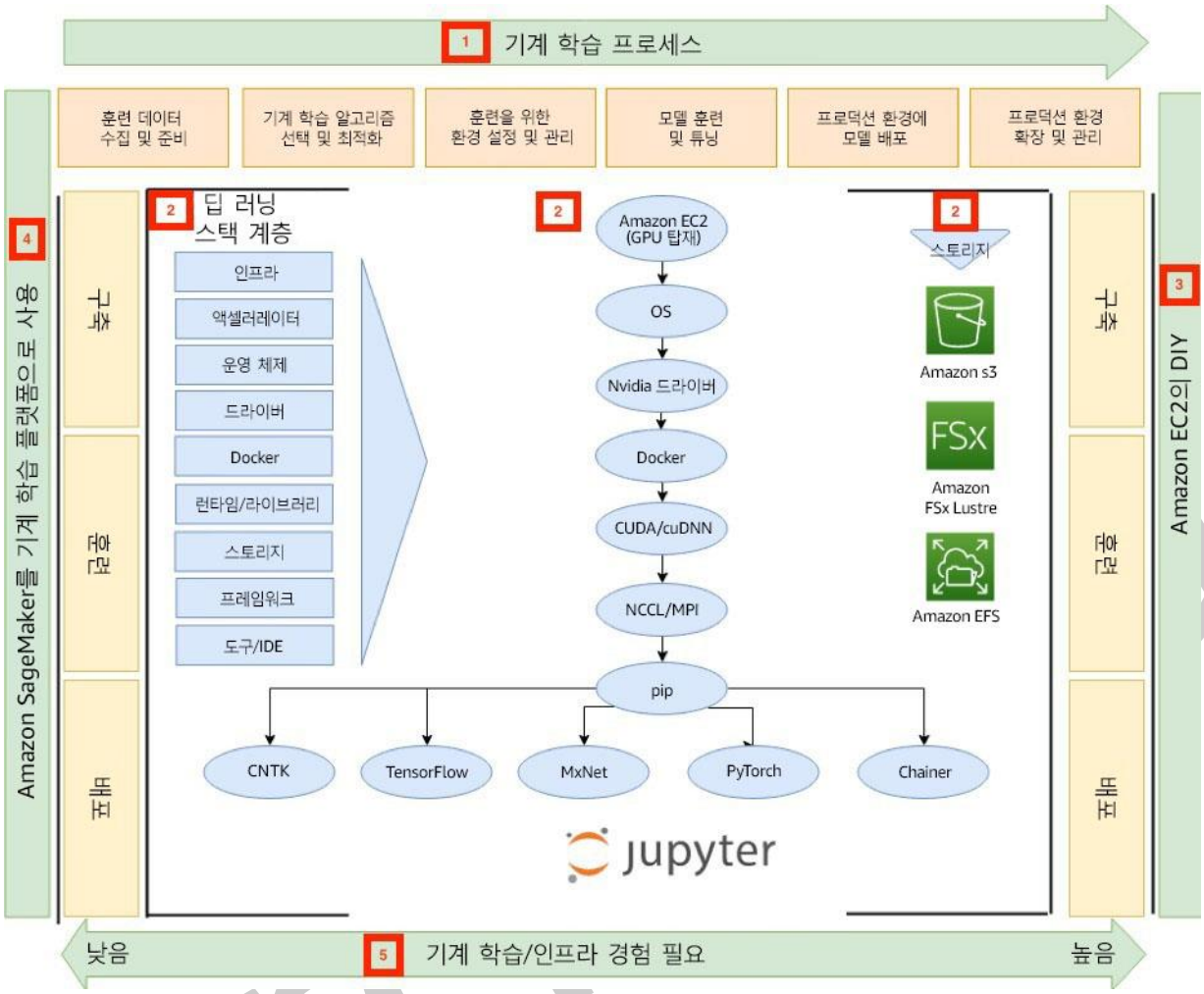


그림 1: 딥 러닝 환경

표 1: 딥 러닝 환경 다이어그램 설명

| 레이블 | 설명 |
|---|---|
| 1 | 딥 러닝 프로젝트를 실행하기 위해서는 여섯 단계가 필요합니다. 이 여섯 단계는 구축, 훈련 및 배포를 위한 딥 러닝 프로세스에서 다룹니다. |
| 2 | 딥 러닝 환경의 구축, 훈련 및 배포 작업을 지원하는 데 필요한 다양한 계층입니다. 그림의 계층은 인프라부터 딥 러닝 프로젝트에 필요한 도구까지 포함합니다. |

| | |
|---|---|
| 3 | AWS 컴퓨팅, 스토리지 및 네트워크 기술 빌딩 블록을 사용하여 딥 러닝에 필요한 구성 요소와 특성을 구축 및 관리하는 역할을 고객이 직접 담당하는 Do-it-yourself(DIY) 옵션입니다. |
| 4 | Amazon SageMaker 는 데이터를 레이블링하여 준비하고, 알고리즘을 선택하고, 모델을 훈련하며, 배포를 위해 미세 조정 및 최적화하고, 예측을 수행하며, 조치를 취하는 전체 딥 러닝 워크플로를 다루는 완전관리형 서비스입니다. 훨씬 더 적은 작업 부담과 저렴한 비용으로 모델을 빠르게 프로덕션 환경에 적용할 수 있습니다. |
| 5 | 사용 편의성 및 고객과 AWS 간 공동 책임 모델을 고려하여 딥 러닝 환경을 설정하는 데 필요한 인프라 경험의 척도입니다. 완전관리형 서비스의 경우 AWS가 스택의 주요 부분을 관리하므로 사용이 간편합니다. DIY 옵션의 경우 고객이 대부분의 스택을 관리해야 하므로 더 까다롭습니다. |

**참고:** 완전관리형(4) 옵션과 DIY(3) 옵션 사이에는 완전관리형 컨테이너 서비스와 자체 관리형 딥 러닝 워크플로 서비스(Kubeflow 등)를 사용하는 부분관리형 접근 방식이 있습니다. 이 부분 관리형 접근 방식은 Kubernetes를 기반으로 인프라를 표준화하려는 조직에서 사용됩니다. 자세한 내용은 DIY 부분 관리형 솔루션: AWS에서 Kubeflow와 함께 Kubernetes 사용을 참조하십시오.

## 안내서 사용

이 안내서는 아래에서 설명하는 바와 같이 7개의 섹션으로 구성되어 있습니다.

1. 구축, 훈련 및 배포를 위한 딥 러닝 프로세스
2. 딥 러닝 프로젝트의 당면 과제
3. 딥 러닝을 위한 고도로 최적화된 AWS 기술 빌딩 블록
4. 코드, 데이터 및 모델 버전 관리
5. 재훈련 및 재배포를 위한 딥 러닝 프로세스 자동화
6. 대규모 딥 러닝을 위한 패턴
7. AWS 지침

딥 러닝 프로세스와 딥 러닝 스택에 대해 잘 모르는 경우 전체 안내서를 순서대로 읽으십시오.

AWS 딥 러닝 빌딩 블록, 딥 러닝 당면 과제 및 딥 러닝 프로세스에 익숙한 경우 섹션 4, 5, 6 및 7로 건너뛰어도 됩니다.

# 구축, 훈련 및 배포를 위한 딥 러닝 프로세스

다음 이미지는 딥 러닝 프로세스의 여섯 단계를 보여줍니다. 이어지는 섹션에서는 딥 러닝 프로세스의 각 단계별로 자세한 정보를 제공하고, 인프라 성능, 병목 지점, 확장성, 안정성 및 사용 편의성과 관련하여 해결해야 할 문제를 설명합니다.

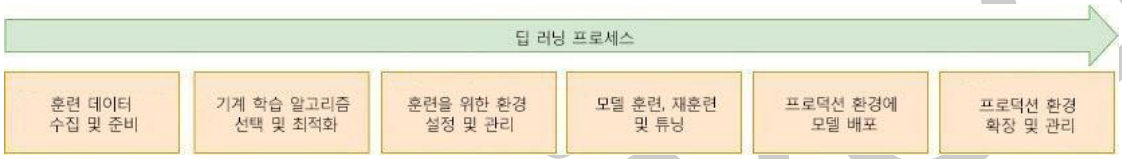


그림 2: 딥 러닝 프로세스의 여섯 단계

## 1단계. 데이터 수집

딥 러닝은 데이터 수집 및 준비 단계에 있어서 기존 기계 학습과 다릅니다. 기존의 기계 학습 구현에서는 특성 추출(feature engineering)이 병목 현상을 일으키는 장애가 되는 경우가 많지만, 딥 러닝(특히 이미지 인식 및 자연어 처리[NLP] 사용 사례)에서는 훈련 과정에서 신경망이 특성을 자동으로 생성할 수 있습니다. 이러한 특성은 딥 네트워크의 각 노드 계층이 샘플을 그리는 데 사용하는 입력을 반복적으로 재구성하여 특성을 학습하도록 함으로써 추출되며, 이를 통해 네트워크의 추측과 입력 데이터 자체의 확률 분포 간 차이를 최소화할 수 있습니다.

그러나 처음부터 새로 훈련할 때는 성능이 우수한 모델을 개발하기 위해 많은 양의 훈련 데이터가 있어야 하므로, 레이블이 지정된 데이터가 상당히 많이 필요하게 됩니다. 특히 새로운 애플리케이션 또는 딥 러닝 구현의 새로운 사용 사례의 경우 레이블이 지정된 데이터가 충분하지 않을 수 있습니다.

이 백서의 딥 러닝을 위한 고도로 최적화된 AWS 기술 빌딩 블록 섹션에서 이러한 딥 러닝의 고유한 문제를 해결하는 방법을 설명합니다. 첫 단계로, 아래의 다이어그램을 사용하여 데이터 수집 프로세스를 평가합니다.

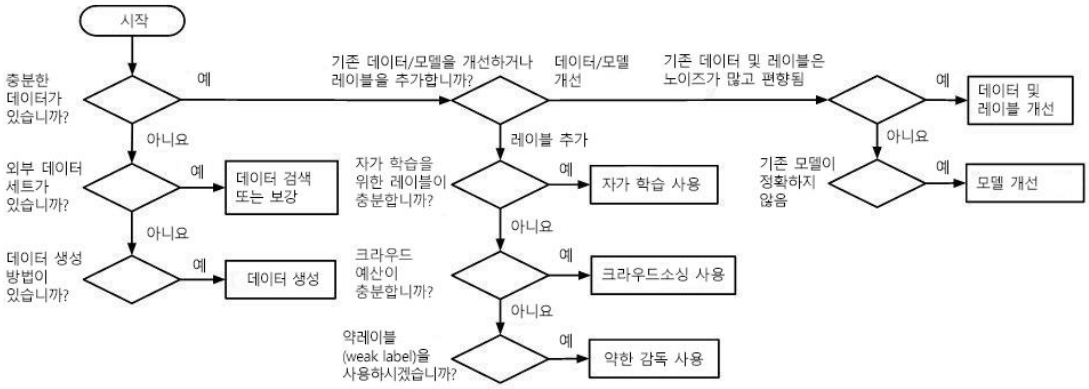


그림 3: 데이터 수집 평가[1]

## 데이터 전 처리

데이터 전 처리는 데이터 정제, 데이터 통합, 데이터 변환 및 데이터 감소로 구성되며, 훈련 프로세스를 시작하기 전에 부정확하거나, 누락되거나, 노이즈가 있거나, 일관되지 않은 데이터를 줄이는 것을 목적으로 합니다. AWS는 특성 추출을 실행하는 것 외에, 데이터 전 처리 단계를 수행할 때 사용할 수 있는 AWS Glue, Amazon EMR, AWS Lambda, Amazon SageMaker, AWS Batch, AWS Marketplace 등의 다양한 도구와 서비스를 제공합니다. 이러한 도구의 사용은 AWS의 빅 데이터 분석 옵션 백서에서 자세히 설명합니다.

무엇보다, 수많은 오픈 소스 딥 러닝 프레임워크가 광범위하게 사용됨에 따라 개별 프레임워크를 지원하기 위한 다양한 파일 형식이 등장하고 있다는 점에 주목해야 합니다. 데이터 수집 프로세스의 파일 형식을 선택하는 작업은 데이터 전 처리에서 중요한 단계이며 딥 러닝 구현을 수행하기 위해 선택하는 프레임워크에 따라 크게 달라집니다. 표준 형식의 예로는 RecordIO, TFRecords, Hierarchical Data Format(HDF5), pth, N5, LMDB(Light Memory Mapped Database) 등이 있습니다.

# 2단계. 알고리즘 선택 및 최적화

딥 러닝 구현 환경 내에는 서로 구분되는 다양한 네트워크 아키텍처와 딥 러닝 알고리즘이 있습니다. 사용 가능한 모든 네트워크 아키텍처와 훈련 알고리즘을 설명하는 것은 본 백서의 범위를 벗어납니다. 여기서는 가장 일반적으로 사용되는 네트워크 아키텍처 세 가지와 현재 가장 많이 사용되는 몇 가지 인기 있는 훈련 알고리즘에 대해 간략하게 설명하겠습니다.

## 딥 러닝 네트워크 아키텍처

- MLP(다중 계층 퍼셉트론)(피드포워드 신경망[FFNN])
- CNN(합성곱 신경망)
- RNN(순환 신경망)

간단히 말하면, 해결하려는 구체적인 사용 사례에 따라 네트워크 아키텍처를 선택합니다. 다음 표는 개별 네트워크 아키텍처에 사용 사례를 매핑하는 의사 결정 매트릭스입니다.

표 2: 네트워크 아키텍처에 사용 사례 매핑

| MLP(FFNN) | CNN | RNN(LSTM) |
|---|---|---|
| 테이블형 데이터 세트 | 이미지 데이터 | 텍스트 데이터 |
| 분류 예측 문제 | 분류 예측 문제 | 음성 데이터 |
| 회귀 예측 문제 | 회귀 예측 문제 | 분류 예측 문제 |
| | | 회귀 예측 문제 |

## 딥 러닝 알고리즘

대부분의 딥 러닝 모델은 경사 하강법 및 오차역전파법을 사용하여, 훈련 프로세스를 진행하는 중에 오류의 전체적인 변화에 대한 각 파라미터의 기여도를 나타내는 편미분을 사용하여 신경망의 파라미터를 최적화합니다. 딥 러닝 알고리즘의 훈련 성능에 관한 최적화

기법을 찾는 것은 지속적으로 진행되는 연구 주제이자, 여전히 진화하고 있는 주제입니다. 딥 러닝 네트워크의 훈련 성능을 개선할 목적으로, 모멘텀, AdaGrad(적응형 경사법 알고리즘), Adam(적응형 모멘텀 추정), Gadam(유전자 진화 Adam) 등, 기존 경사 하강법 최적화 알고리즘을 기반으로 새롭게 변형된 최적화 알고리즘이 많이 등장했습니다.

## 3단계. 훈련을 위한 환경 설정 및 관리

훈련 작업을 위한 딥 러닝 환경을 설계하고 관리하기는 어려울 수 있습니다. 딥 러닝 훈련 작업은 기존 기계 학습 구현과 다릅니다. 대부분의 신경망에서 나타나는 복잡성, 데이터 세트의 높은 차원 수, 마지막으로 대량의 훈련 데이터를 사용하여 대규모 모델을 훈련하는 데 필요한 인프라의 규모 등으로 인해 문제가 발생합니다. 이러한 문제를 해결하려면 컴퓨팅 및 스토리지 인프라의 높은 탄력성과 성능이 요구됩니다.

AWS에서 신경망을 완전히 새로 구축하는 경우 TensorFlow, PyTorch, Apache MXNet, Chainer, Microsoft Cognitive Toolkit, Gluon, Horovod 및 Keras로 사전 구성된 AWS Deep Learning Amazon Machine Image(AWS DL AMI)를 활용하면 프레임워크와 도구를 대규모로 신속하게 배포하고 실행할 수 있습니다. 또한 TensorFlow 및 Apache MXNet을 지원하는 딥 러닝 프레임워크와 함께 사전 구성된 AWS Deep Learning Containers (AWS DL Containers)를 사용하여 Amazon Elastic Kubernetes Service(Amazon EKS), 자체 관리형 Kubernetes, Amazon Elastic 컨테이너 서비스(Amazon ECS)에서 실행하거나 Amazon Elastic Compute Cloud(Amazon EC2)에서 직접 실행할 수 있습니다. 마지막으로 Python용 AWS SDK를 활용할 수 있습니다. 이 SDK는 다양한 기계 학습 및 딥 러닝 프레임워크가 Amazon SageMaker에서 모델을 훈련하고 배포할 수 있도록 오픈 소스 API 및 컨테이너를 제공합니다. 이 백서의 후반부에서는 이러한 서비스를 사용하는 가장 일반적인 솔루션과 패턴을 설명합니다.

## 4단계. 모델 훈련, 재훈련 및 튜닝

"좋은" 해가 여러 개 있는 비볼록 오류 공간(nonconvex error space)에서 함수 근사화를 통해 입력에서 출력으로의 매핑 함수를 학습하는 방식으로 모델의 학습이 이루어지기 때문에 신경망 훈련은 기존 기계 학습 구현 방식과 다릅니다.
단순한 선형 회귀 모델의 경우처럼 폐쇄형 해를 통해 최적의 가중치 집합을 직접 계산할 수 없고 전역 수렴이 보장되지 않으므로 신경망 훈련은 어려울 수 있으며 일반적으로 다른 기계 학습 알고리즘보다 훨씬 더 많은 데이터 및 컴퓨팅 리소스가 필요합니다.

AWS는 신경망의 훈련 프로세스를 간소화할 수 있는 다양한 도구와 서비스를 제공합니다. 이 백서에서는 Amazon EC2에서 자체 관리형 딥 러닝 환경을 실행하거나, Amazon EKS 또는 Amazon ECS에서 딥 러닝 환경을 실행하거나, 딥 러닝을 지원하는 완전관리형 서비스인 Amazon SageMaker를 사용하는 등의 다양한 옵션을 설명합니다. 이들 환경은 모두 고도로 맞춤화된 GPU 기반 하드웨어를 사용하여 훈련 시간과 훈련 비용을 줄여 줍니다.

2단계. 알고리즘 선택 및 최적화에서 설명하는 모델 설계 방식 외에 훈련 프로세스를 시작하기 전에 하이퍼파라미터를 설정하는 옵션도 있습니다. 최적의 하이퍼파라미터를 찾으려면 반복적인 프로세스를 거쳐야 합니다. 딥 러닝 구현에서는 검색 공간의 높은 차원 수와 복잡성 때문에 이 프로세스에 노동력과 비용이 많이 소요될 수 있습니다. 그리드 검색, 임의 검색, 베이지안 최적화 등의 기법을 통해 최적의 하이퍼파라미터 설정을 찾는 다양한 전략이 개발되어 있습니다.

하이퍼파라미터 튜닝은 Amazon SageMaker에서 턴키(turn key) 기능으로 제공됩니다.

## 5단계. 프로덕션 환경에 모델 배포

기계 학습 모델을 프로덕션 환경에 배포하는 작업은 전체 기계 학습 파이프라인에서 가장 어려운 부분으로 꼽히는 경우가 많습니다. 이는 기계 학습 워크로드 배포가 기존 소프트웨어 배포와 다르기 때문입니다.

첫째, 모델에서 배치 추론(오프라인)과 실시간 추론(온라인) 중 어떤 추론 유형을 제공하는지 고려해야 합니다. 이름에서 알 수 있듯이 배치 추론에서는 관측치 집합에 대해 비동기식으로 예측을 생성합니다. 반복적인 일정(예: 매시간, 매일, 매주)에 따라 배치 작업이 생성됩니다. 생성된 예측 데이터는 데이터 리포지토리 또는 파일 스토리지의 플랫 파일에 저장되고 최종 사용자가 사용할 수 있습니다. 실시간 추론은 요청 시 실시간/동기식으로 예측을 생성하는 프로세스로, 대부분 런타임에 데이터를 한 번 관측합니다.

둘째, 모델을 재훈련하는 방식을 고려해야 합니다. 모델을 정확하게 예측하려면, 예측을 위해 모델에 제공되는 데이터의 분포가 모델을 훈련하는 데 사용된 데이터와 비슷해야 합니다. 하지만 대부분의 기계 학습 배포에서 데이터 분포는 시간이 지남에 따라 변화하기 마련이므로 모델 배포는 일회성 작업이 아니라 지속적인 프로세스가 되어야 합니다. 수신데이터를 지속적으로 모니터링하면서 데이터의 분포가 원래 훈련 데이터의 분포에서 크게 벗어나는 경우 최신 데이터로 모델을 재훈련하는 것이 바람직합니다. 사용 사례에 따라, 상황별 대응보다 주기적인 자동 재학습이 적합할 수 있습니다. 예를 들어 데이터 분산의 변화를 감지하기 위해 데이터를 모니터링하는 데 따른 오버헤드가 높은 경우 주기적으로 모델을 훈련하는 것과 같은 단순한 전략이 더 효과적일 수 있습니다.

셋째, 모델 버전 관리를 구현하고 수요를 충족하기 위해 확장 가능한 인프라를 구축하는 것은 딥 러닝에만 국한되는 방식은 아니지만 딥 러닝에서는 추가적으로 고려할 사항이 있습니다. 버전 관리는 모델에서 생성된 출력을 해당 모델의 특정 버전에 연결할 수 있도록 지원하여 검증을 가능케 할 뿐만 아니라 모델과 훈련 파일 간의 추적을 가능케 한다는 점에서 필수적인 기능입니다. 추론 엔드포인트에 GPU 코어의 일부를 할당하는 기능 외에, 추론 엔드포인트에 사용되는 컴퓨팅 파워의 양을 동적으로 조정하는 기능까지 지원함으로써 용량을 오버프로비저닝하지 않고도 애플리케이션의 수요를 충족할 수 있습니다.

넷째, 프로덕션 환경에 모델을 구현하기 위한 마지막 단계로서, 시간 경과에 따른 모델 성능을 감사할 도구를 구현해야 합니다. 감사 솔루션은 시간 경과에 따라 객관적인 평가 지표를 정확하게 관측하고, 실패를 감지하고, 시간이 지나면서 모델의 정확도가 저하될 경우

피드백 루프를 구성할 수 있어야 합니다. 이 안내서에서는 감사 솔루션에 대해서는 다루지 않습니다.

마지막으로, AWS에서 제공하는 다양한 모델 배포 및 모델/데이터 버전 관리 접근 방식에 대해 다음 두 단원, 즉 코드, 데이터 및 모델 버전 관리 및 대규모 딥 러닝을 위한 패턴에서 보다 자세히 알아보겠습니다.

## 6단계. 프로덕션 환경 확장 및 관리

효과적인 기계 학습 시스템을 구축하고 배포하는 작업은 반복적인 프로세스이며, 시스템을 변경하는 속도는 아키텍처의 확장에 직접적으로 영향을 미치는 동시에 팀의 생산성에도 영향을 미칩니다. 딥 러닝 구현 시 확장성과 관리 용이성을 실현하기 위해 가장 중요한 고려 사항 세 가지는 모듈화, 계층화된 오퍼링, 그리고 수많은 프레임워크 중 적합한 프레임워크를 선택하는 것입니다. 분리되고 프레임워크에 구애받지 않는 이 접근 방식은 딥 러닝 엔지니어와 과학자가 원하는 도구를 제공하는 동시에 조직 내의 구체적인 사용 사례와 기술에 맞춤화됩니다.

AWS는 모든 일반적인 기계 학습 워크플로를 아우르는 광범위한 서비스 포트폴리오를 제공하는 동시에 일반적이지 않은 맞춤형 사용 사례를 지원할 수 있는 유연성도 제공합니다. AWS 기계 학습 스택의 범위와 깊이에 대해 논의하기 전에 현재 기계 학습 전문가가 직면하는 가장 일반적인 문제를 살펴보겠습니다.

# 딥 러닝 프로젝트의 당면 과제

## 소프트웨어 관리

딥 러닝 프로젝트는 기계 학습 프레임워크에 의존합니다. 딥 러닝 프레임워크는 오픈 소스 기반인 경우가 많고 프레임워크 코드 개선에 적극적으로 참여하는 커뮤니티에 의해 유지 보수됩니다. 변경 사항이 자주 발생하며 때로는 완전히 새롭게 적용되는 것일 수도

있습니다. 경우에 따라 사용자 정의 연산자를 작성하여 갑작스러운 성능 요구 사항을 충족하도록 프레임워크를 일부 수정해야 합니다.

기계 학습 프레임워크를 구축, 테스트, 유지 관리하는 작업을 수행해야 합니다. 완전히 새롭게 적용되는 변경 사항인 경우 스크립트도 변경해야 합니다. 그러나 오픈 소스 인공 지능 커뮤니티의 최신 기술을 활용하고 내부 프로젝트의 요구 사항을 지원하는 것이 무엇보다 중요합니다.

## 성능 최적화

딥 러닝의 전체 스택은 여러 계층으로 구성되어 있습니다. 스택에서 최대한의 성능을 끌어내려면 드라이버, 라이브러리 및 종속성을 비롯한 소프트웨어의 모든 계층을 미세하게 조정해야 합니다. 소프트웨어의 계층을 잘못 튜닝하면 모델의 훈련 시간이 길어지고 훈련 비용이 증가할 수 있습니다.

딥 러닝 스택을 튜닝하려면 여러 번의 반복 테스트와 전문 기술이 필요합니다. 대부분의 경우 튜닝은 훈련 스택과 추론 스택 모두에서 필요합니다. 스택마다 네트워크, CPU, 스토리지 I/O 등 병목 지점이 서로 다를 수 있으며, 이는 튜닝을 통해 해결해야 합니다.

## 협업 개발

대부분의 경우 협업을 통해 딥 러닝 프로젝트를 진행하는 딥 러닝 엔지니어와 딥 러닝 과학자로 팀이 구성됩니다. 이 같은 팀은 특정 표준을 준수하여 협업을 진행하고 서로의 작업에 대한 피드백을 제공해야 합니다. 일관성을 유지하기 위해서는 프로젝트 개념 증명(PoC) 단계부터 프로덕션 단계까지 지속적으로 모델의 성능을 추적하는 것이 중요합니다. 다양한 실무자가 훈련 중에 사용하는 다양한 스택의 데이터 세트와 하드웨어 버전 및 소프트웨어 구성에는 일관성이 요구됩니다. 훈련과 추론 스택 간에도 일관성이 요구되며, 스택과 해당 스택에서 산출된 결과를 재현할 수 있어야 합니다.

## 인프라 관리

모델의 가치를 입증하려면 적절한 하이퍼파라미터와 대규모 데이터 세트로 훈련해야 합니다. 최적의 하이퍼파라미터를 검색하려면 대규모 데이터 세트에 대해 여러 작업을 동시에 실행해야 합니다. 이를 위해서는 작업 스케줄러, 오케스트레이터 및 모니터링 도구를 사용해야 하며, 결과적으로 중앙 집중식 IT 팀이 관리하는 IT 자산에 대한 종속성이 나타나게 됩니다. 모델의 첫 번째 버전 개발이 완료된 후에도 DevOps 팀은 새로운 데이터로 모델을 재훈련하고, 모델 배포에 사용되는 엔드포인트를 모니터링 및 지원할 인프라를 지원해야 합니다.

## 확장성

딥 러닝 훈련 워크로드와 추론 워크로드는 일시적으로 급증하기도 합니다. 프로젝트를 진행하는 동안 특정 기간에 특정 실험을 수백, 수천 개의 인스턴스로 확장해야 할 수 있습니다. 추론과 관련해서는 연중 특별 이벤트 기간에 추론의 일시적 급증이 예상되는 경우가 대표적인 예입니다.
훈련과 추론을 위한 실험을 지원하고 일시적 워크로드 급증에 대비하기 위해 필요한 고성능 컴퓨팅을 사전에 계획하고 예측하기는 쉽지 않습니다.

딥 러닝에 최적화된 AWS 서비스는 딥 러닝 엔지니어와 딥 러닝 과학자가 직면한 위와 같은 문제를 해결합니다. 그러면 개별 빌딩 블록을 자세히 살펴보겠습니다.

# 딥 러닝을 위한 고도로 최적화된 AWS 기술 빌딩 블록

다음 그림은 AWS의 개별 딥 러닝 계층 구성을 간략하게 보여줍니다.

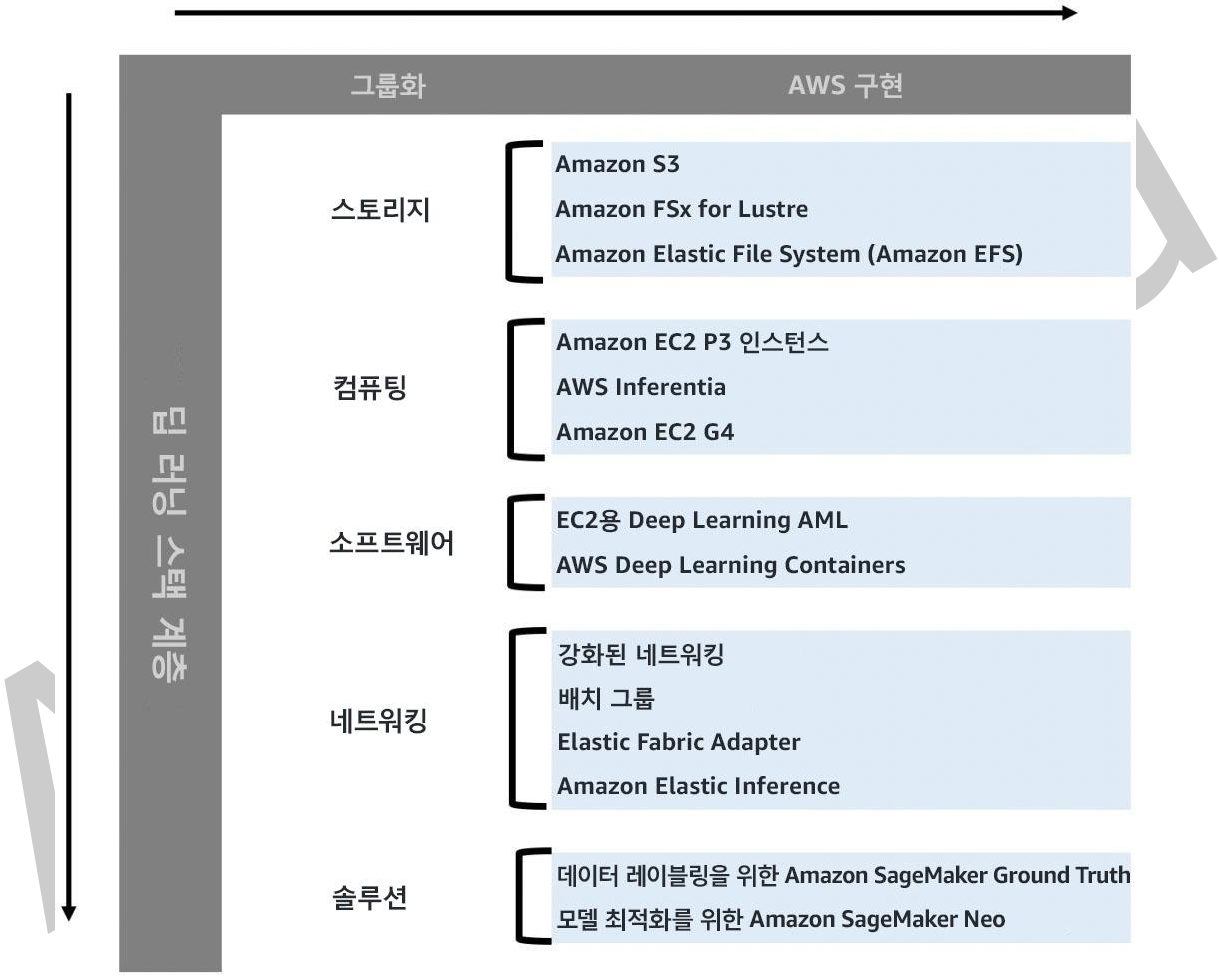


그림 4: 딥 러닝을 위한 빌딩 블록

# 스토리지

## Amazon Simple Storage Service(Amazon S3)

데이터를 획득하고, 다양한 딥 러닝 프로젝트를 지원하기 위해 엔터프라이즈 환경 전체에서 해당 데이터를 탐색 및 소비할 수 있도록 하는 것은 중요한 전략적 이니셔티브입니다. 이를 위해서는 데이터 수집, 추출/변환/로드(ETL) 수행, 데이터 시각화, 데이터 랭글링 등의 작업을 통해 딥 러닝 모델 훈련을 위한 고품질 훈련 데이터 세트를 개발해야 합니다.

Amazon Simple Storage Service(Amazon S3) 를 중앙 스토리지 계층으로 사용하여 딥 러닝을 위한 데이터를 저장하고 대중화할 수 있습니다. 애플리케이션에서 Amazon S3 를 딥 러닝 훈련 작업을 위한 스토리지 티어로 사용하여 초당 수천 건의 트랜잭션을 손쉽게 실현할 수 있습니다. Amazon S3는 높은 요청 속도를 지원하도록 자동으로 확장됩니다.

Amazon S3에서 객체를 수집하고 읽는 동안 Amazon EC2 와 Amazon S3 간에 요구되는 처리량을 고려해야 합니다. 분산된 방식으로 여러 Amazon EC2 인스턴스를 사용하여 더 높은 성능을 실현할 수 있습니다.

Amazon SageMaker는 훈련 작업 및 배치 추론에 사용된 데이터와 훈련된 모델을 저장하기 위한 스토리지 티어로 Amazon S3를 사용합니다. Amazon SageMaker는 Amazon S3에서 Amazon SageMaker 훈련 인스턴스의 로컬 Amazon Elastic Block Store(Amazon EBS) 볼륨으로 데이터를 읽는 작업 시 배치 모드와 파이프 모드를 모두 지원합니다.

Amazon EC2 에서 자체 컴퓨팅 클러스터를 관리하려는 DIY 고객은 Amazon S3를 스토리지 계층으로 사용하거나 Amazon S3에서 지연 로딩과 함께 하이드레이션된 Amazon FSx for Lustre 를 사용하여 딥 러닝 작업을 위한 데이터 캐싱 계층을 구축할 수 있습니다. 두 옵션 모두 DIY 환경에 사용할 수 있습니다. 가격과 성능 간의 균형을 유지해야 합니다.

## Amazon FSx for Lustre

Amazon FSx for Lustre 는 오픈 소스 Lustre를 기반으로 합니다. Lustre는 분산 훈련을 위한 딥 러닝 데이터 캐싱 계층으로 사용 가능한 확장성이 뛰어나며 고도로 분산된 고도의 병렬 오픈 소스 파일 시스템입니다.

Lustre는 고성능 기능과 개방형 라이선스 때문에 딥 러닝 워크로드에 널리 사용되고 있습니다. Lustre 파일 시스템은 확장성이 뛰어나며, 수만 개의 클라이언트 노트, PB 규모의 데이터, 초당 TB 규모의 집계 I/O 처리량을 지원하는 여러 컴퓨팅 클러스터에 사용할 수 있습니다.

딥 러닝 신경망을 훈련하는 경우 Lustre는 짧은 지연 시간으로 소스 데이터를 빠르게 획득할 수 있는 기능을 제공합니다. 하지만 Lustre 클러스터를 설정하기가 어려울 수 있습니다.

Amazon FSx for Lustre 는 Lustre 파일 시스템 설정 및 관리의 복잡성을 간소화하고, 몇 분 만에 파일 시스템을 생성하고, 원하는 수의 클라이언트에 탑재하고, 바로 액세스를 시작할 수 있는 환경을 제공합니다. Amazon FSx for Lustre 는 완전관리형 서비스이므로 별도로 유지 관리할 필요가 없습니다. 임시 사용을 위해 독립 실행형 파일 시스템을 구축하거나, S3 버킷을 Amazon FSx for Lustre와 연동하여 Lustre 파일 시스템인 것처럼 S3 버킷의 데이터에 액세스할 수 있습니다. Amazon FSx for Lustre 는 높은 수준의 처리량, IOPS 및 일관되게 짧은 지연 시간이 요구되는 워크로드에 적합하도록 설계되었습니다.

Amazon FSx for Lustre 의 고유한 기능 중 하나는 Amazon S3와 완벽하게 통합되어 실제 파일 시스템으로 데이터를 지연 로드하는 기능입니다. 고객이 S3 버킷에서 어떤 객체를 로드해야 할지 모르는 경우 Amazon FSx for Lustre 는 객체 자체의 이름, 날짜, 크기 등으로 구성된 메타데이터만 로드하고 필요할 때까지 실제 파일 데이터를 로드하지 않습니다. 기본적으로 Amazon S3 객체는 애플리케이션에 의해 처음으로 액세스될 때만 파일 시스템으로 로드됩니다. 애플리케이션이 파일 시스템에 아직 로드되지 않은 객체에 액세스할 경우 Amazon FSx for Lustre 는 Amazon S3에서 해당 객체를 자동으로 로드합니다.

## Amazon Elastic File System(Amazon EFS)

스토리지 솔루션을 선택할 때에는 데이터 로컬화와 중앙 관리형 스토리지 솔루션의 장단점을 고려해야 합니다. Amazon EFS 는 최고 수준의 처리량이 요구되는 고도로 병렬화된 확장 워크로드부터 지연 시간에 민감한 단일 스레드 워크로드까지, 광범위한 사용 사례를 효과적으로 지원할 수 있습니다. 하지만 중앙 위치에서 배치 처리를 실행할 때는 대개 Amazon EFS가 스토리지 솔루션으로 가장 적합합니다. Amazon EFS를 사용하면 스토리지를 프로비저닝하거나 네트워크 파일 시스템을 직접 관리할 필요 없이 노트북 환경에서도 직접 대규모 기계 학습 데이터 세트 또는 공유 코드에 손쉽게 액세스할 수 있습니다.

Amazon EFS는 수집되는 데이터의 양이 늘어나면 자동으로 확장됩니다. 데이터는 여러 가용 영역에 중복 저장되며, 데이터가 증가함에 따라 초당 10GB 이상의 처리량으로 성능이 확장됩니다. Amazon EFS는 여러 가용 영역에 걸쳐 수천 개의 Amazon EC2 인스턴스에 동시에 탑재할 수 있습니다.

아래 다이어그램에서 보듯이 여러 가용 영역에 구축된 최대 수천 개의 Amazon EC2 인스턴스를 파일 시스템에 동시에 연결할 수 있습니다. 또한 여러 Amazon SageMaker Jupyter 노트북에 탑재할 수도 있습니다. 이 기능은 딥 러닝 엔지니어와 딥 러닝 과학자가 협업에 필요한 데이터와 코드를 공유하는 데 Amazon EFS를 사용할 수 있도록 합니다. 또한 Amazon EFS를 분산 훈련 작업의 훈련 데이터 세트를 위한 캐싱 계층으로 사용할 수도 있습니다.

다음 그림은 모든 휘발성 컴퓨팅 노드에 Amazon EFS 엔드포인트를 추가함으로써, 중앙에서 액세스 가능한 스토리지 솔루션을 탑재하는 방법을 보여줍니다. 가장 중요한 점은 이 엔드포인트가 애플리케이션에 영향을 주지 않으면서 온디맨드로 페타바이트까지 확장할 수 있고, 파일을 추가 및 제거함에 따라 자동으로 확장 및 축소된다는 것입니다.

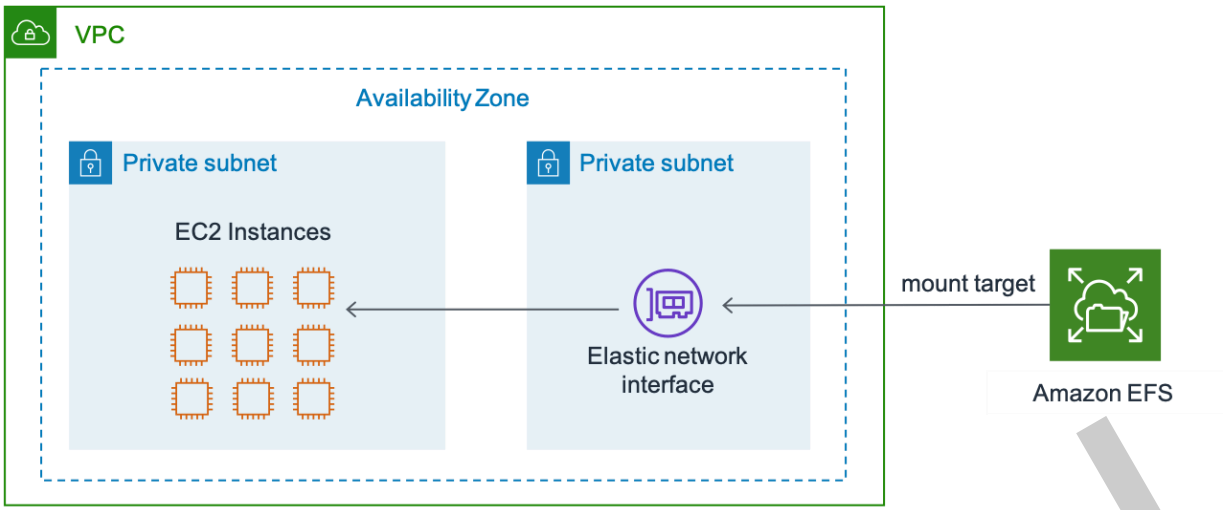


그림 5: 파일 시스템에 연결된 여러 EC2 인스턴스



# 컴퓨팅

## Amazon EC2 P3 인스턴스

신경망의 계산 집약적인 부분은 많은 행렬 및 벡터 연산에서 비롯됩니다. 연산을 하나씩 개별적으로 수행하는 대신 모든 연산을 동시에 수행하여 이 프로세스의 처리 속도를 높일 수 있습니다. 여러 간단한 계산을 병렬로 처리하는 데 더 효과적인 GPU가 신경망을 훈련하는 데 CPU 대신 사용되는 이유가 여기에 있습니다.

더 많은 계층을 신경망에 추가하고(특정 한도까지), 훈련 데이터를 늘리면 딥 러닝 모델의 성능이 향상된다는 것이 입증되었습니다. GPU는 수천 개의 단순 코어로 이루어져 있으며 수천 개의 동시 스레드를 실행할 수 있습니다. GPU는 복잡한 신경망을 훈련하는 데 소요되는 훈련 시간을 단축해 주었습니다. 신경망 아키텍처를 사용하여 복잡한 모델을 구축하는 프로젝트에서는 기본 요구 사항으로서 비용 효율적인 고성능 GPU 인프라에 대한 액세스와 가용성이 요구됩니다. GPU 기반 Amazon EC2 P3 인스턴스 는 현재 클라우드에서 사용되는 다른 GPU 대안과 비교하여 최고 수준의 가격 대비 성능을 제공합니다.

차세대 EC2 컴퓨팅 최적화 GPU 인스턴스인 Amazon EC2 P3 인스턴스는 최대 8개의 최신 세대 NVIDIA Tesla V100 GPU로 구동되며 딥 러닝 애플리케이션에 적합합니다.

Amazon EC2 P3 인스턴스 는 맞춤형 인텔 제온 E5 프로세서를 사용한 64개의 vCPU, 488GB RAM, Elastic Network Adapter(ENA) 기술을 적용한 최대 25Gbps의 집계 네트워크 대역폭을 활용하여 딥 러닝을 위한 강력한 플랫폼을 제공합니다. ENA는 뒤쪽의 섹션에서 자세히 설명합니다.

GPU는 CPU보다 속도가 빠르며, 훈련 작업 중에 네트워크와 CPU를 포화시킬 수 있습니다. 훈련 인스턴스의 네트워크 파이프 크기와 vCPU 수는 병목 지점이 될 수 있으며 GPU 사용률을 높이는 데 제한 요인으로 작용할 수 있습니다.

P3 인스턴스 패밀리의 최신 Amazon EC2 P3dn.24xlarge GPU 인스턴스는 P3.16xlarge 인스턴스의 최대 4배에 달하는 네트워크 대역폭을 제공하며 앞서 언급한 제한 사항을 해결하기 위해 특별히 설계되었습니다.

위의 Amazon EC2 P3 인스턴스 개선 사항은 단일 인스턴스에서 성능을 최적화할 뿐만 아니라 딥 러닝 모델 훈련 시간도 단축합니다. 이는 훈련 인스턴스 간에 최대 100Gbps의 네트워크 처리량을 활용하여, 여러 인스턴스에 걸쳐 개별 작업을 확장하는 방식으로 구현됩니다.

AWS는 100Gbps의 네트워킹 처리량을 제공하는 최초의 클라우드 공급자입니다. 이 같은 높은 처리 성능은 데이터 전송 병목 지점을 없애고 GPU 사용률을 최적화하여 인스턴스 성능을 극대화해 줍니다. GPU당 16GB~32GB인 GPU 메모리를 두 배로 늘리면 더 높은 수준의 대규모 기계 학습 모델을 훈련할 수 있을 뿐만 아니라 이미지 분류 및 객체 감지 시스템을 위한 4k 이미지와 같은 대규모 데이터 배치를 처리하는 유연성도 확보할 수 있습니다.

P3 인스턴스의 구성과 요금 정보를 비교하려면 Amazon EC2 P3 인스턴스 제품 세부 정보를 참조하십시오.

## AWS Inferentia

훈련된 기계 학습 모델을 사용하여 예측(추론이라고 하는 프로세스)을 수행하는 데 애플리케이션의 컴퓨팅 비용 중 최대 90%가 소요됩니다. 추론은 기계 학습의 값을 구하는 단계입니다. 이 단계에서 음성이 인식되고, 텍스트가 번역되고, 비디오 내의 객체가 인식되며, 제조 결함이 발견되고, 자동차의 자율 주행이 이루어집니다.

Amazon Elastic Inference는 코드 변경 없이 적절한 양의 GPU 지원 추론 가속을 Amazon EC2 또는 Amazon SageMaker 인스턴스 유형에 연결함으로써 이 문제를 해결합니다. Amazon Elastic Inference를 사용하면 애플리케이션의 전체 CPU 및 메모리 요구 사항에 가장 적합한 인스턴스 유형을 선택한 다음 리소스를 효율적으로 사용하는 데 필요한 추론 가속화 용량을 개별적으로 구성하여 추론 실행 비용을 줄일 수 있습니다.

하지만 일부 추론 워크로드의 경우 전체 GPU를 사용해야 하거나 짧은 지연 시간이 요구됩니다. 이 문제를 저렴한 비용으로 해결하려면 특화된 전용 추론 칩이 필요합니다.

AWS Inferentia 는 저렴한 비용으로 뛰어난 성능을 제공하도록 설계된 기계 학습 추론 칩입니다. AWS Inferentia 하드웨어 및 소프트웨어는 다양한 추론 사용 사례와 최첨단 신경망을 지원합니다.

AWS Inferentia 는 TensorFlow, Apache MXNet 및 PyTorch 딥 러닝 프레임워크뿐만 아니라 ONNX 형식을 사용하는 모델도 지원합니다.

각 AWS Inferentia 칩은 복잡한 모델에서 빠른 예측을 수행할 수 있도록 수백 TOPS(초당 테라 연산)의 추론 처리량을 제공합니다. 또한 여러 AWS Inferentia 칩을 함께 사용하여 수천 TOPS의 처리량으로 성능을 더욱 높일 수도 있습니다. AWS Inferentia 는 Amazon SageMaker, Amazon EC2, Amazon Elastic Inference와 함께 사용할 수 있습니다. AWS Inferentia가 제공될 경우 알림을 받으려면 여기에서 가입하십시오.

### Amazon EC2 G4

이제 우리는 모든 고객 상호 작용이 백엔드의 인공 지능에 의해 구동되는 시대로 발전하고 있습니다. 고객의 요구를 충족하거나 뛰어넘으려면 인공 지능 기반 제품 및 서비스를 비용 효율적으로 확장할 수 있는 컴퓨팅 플랫폼이 필요합니다.

NVIDIA® Tesla® T4 GPU 는 세계에서 가장 발전된 추론 액셀러레이터입니다. NVIDIA Turing™ Tensor Core로 구동되는 T4는 혁신적인 다중 정밀 추론 성능을 제공하여 모던 인공 지능의 다양한 애플리케이션에서 성능을 가속화합니다. T4는 확장 서버에 최적화되어 있으며, 실시간으로 최첨단 추론을 제공하도록 특별히 설계되었습니다.

응답성은 대화형 인공 지능, 추천 시스템, 시각 검색 등의 서비스에 대한 사용자 참여를 유도하는 데 핵심적인 요소입니다. 모델이 더 정확하고 복잡해짐에 따라 올바른 답을 제공하는 데 필요한 컴퓨팅 용량도 기하급수적으로 늘어납니다. Tesla T4는 짧은 지연 시간으로 처리량을 최대 40배 개선하여 더 많은 요청을 실시간으로 처리할 수 있도록 합니다.

새로운 Amazon EC2 G4 인스턴스는 T4 기반 GPU를 탑재하여 AWS 고객에게 다양한 인공 지능 서비스를 비용 효율적으로 배포할 수 있는 다목적 플랫폼을 제공합니다. 고객은 AWS Marketplace를 통해 NVIDIA CUDA-X 인공 지능 라이브러리를 비롯한 NVIDIA GPU 가속 소프트웨어와 G4 인스턴스를 페어링하여 딥 러닝 추론을 가속화할 수 있습니다.

새로운 T4 기반 G4 인스턴스를 사용하면 기계 학습 추론의 편의성과 비용 효율성을 높일 수 있습니다.

Amazon EC2 G4는 평가판으로 제공됩니다.

## 소프트웨어

### AWS Deep Learning AMI

숙련된 기계 학습 전문가라도 딥 러닝을 시작하는 것은 시간이 많이 들고 번거로울 수 있습니다.

AWS Deep Learning AMI는 널리 사용되는 Python 패키지 및 Anaconda 플랫폼을 설치하는 것은 물론, 사전 구성된 CUDA 및 cuDNN 드라이버를 통한 최신 NVIDIA GPU 가속화 기술, 인텔 MKL(Math Kernel Library)을 제공하여 개발 및 모델 훈련을 가속화합니다.

AWS Deep Learning AMI는 기계 학습 전문가 및 연구원에게 규모에 관계없이 클라우드에서 딥 러닝을 가속화할 수 있는 인프라와 도구를 제공합니다. TensorFlow, PyTorch, Apache MXNet, Chainer, Gluon, Horovod, Keras 등 인기 있는 딥 러닝 프레임워크 및 인터페이스가 사전 설치된 Amazon EC2 인스턴스를 빠르게 시작하여 정교한 사용자 지정 인공 지능 모델을 훈련하고 새로운 알고리즘을 실험할 수 있습니다.

Deep Learning AMI는 Conda AMI와 Base AMI라는 두 가지 버전으로 제공됩니다.

별도의 가상 환경에 딥 러닝 프레임워크 pip 패키지를 사전 설치하려는 개발자의 경우 Ubuntu, Amazon Linux 및 Windows 2016 버전으로 제공되는 Conda 기반 AMI를 활용할 수 있습니다. AMI의 환경은 상호 격리된 독립형 샌드박스로 작동합니다. 또한 AMI는 Jupyter 노트북에 연결되는 시각적 인터페이스를 제공하므로 Jupyter 노트북 브라우저에서 클릭 한 번으로 바로 환경을 전환하고 원하는 환경에서 노트북을 시작하고 환경을 재구성할 수 있습니다.

프라이빗 딥 러닝 엔진 리포지토리 또는 딥 러닝 엔진의 맞춤형 빌드를 설치할 초기 환경을 원하는 개발자의 경우 Ubuntu 및 Amazon Linux 버전으로 제공되는 Base AMI을 활용하면 됩니다. Base AMI는 딥 러닝을 위한 기본 구성 요소가 사전 설치되어 제공됩니다. Base AMI에는 NVIDIA CUDA 라이브러리, GPU 드라이버 및 시스템 라이브러리가 포함되어 있어 Amazon EC2 인스턴스에서 기계 학습을 가속화하고 확장할 수 있습니다. Base AMI는 CUDA 9 환경이 기본적으로 설치된 상태로 제공됩니다. 하지만 간단한 한 줄 명령을 사용하여 CUDA 8 환경으로 전환할 수 있습니다.

## AWS Deep Learning Containers

AWS는 딥 러닝 훈련 및 추론을 가속화할 수 있는 다양한 컴퓨팅 옵션을 제공합니다. 고객은 Amazon SageMaker를 사용하여 완전관리형 서비스를 사용하거나 Deep Learning AMI를 사용하여 DIY 접근 방식을 이용할 수 있습니다.

DIY는 프레임워크 수준에서 작업하는 연구원과 응용 기계 학습 실무자 사이에서 널리 사용되는 옵션입니다.

여러 환경에서 일관되게 실행되는 맞춤형 기계 학습 환경을 배포할 수 있는 Docker 컨테이너가 지난 몇 년간 널리 사용되고 있습니다. Docker 컨테이너를 구축하고 테스트하는 것은 어렵고 오류가 발생하기 쉽습니다. 소프트웨어 종속성 및 버전 호환성 문제로 인해 Docker 컨테이너를 구축하는 데는 며칠이 걸립니다. 또한 전체 인스턴스 클러스터에 걸쳐 기계 학습 작업을 확장하고 분산하려면 Docker 컨테이너 이미지를 최적화하는 전문 기술이 필요합니다. 그리고 새 버전의 소프트웨어 또는 드라이버가 릴리스되면 이 프로세스를 반복해야 합니다.

AWS는 AWS Deep Learning Containers(AWS DL Containers)를 통해 고급 기계 학습 실무자를 위한 DIY 서비스를 확장하고, TensorFlow 및 Apache MXNet과 같은 프레임워크로 사전 구성된 딥 러닝용 Docker 컨테이너 이미지를 제공하고 있습니다. 즉, 딥 러닝을 위한 Docker 컨테이너 구축 및 최적화와 관련된 획일적인 작업을 AWS가 처리합니다. AWS DL Containers는 Amazon Elastic Container Service(Amazon ECS) 및 Amazon Elastic Kubernetes Service(Amazon EKS)와 완벽하게 통합됩니다. 클릭 한 번으로 Amazon ECS 및 Amazon EKS에 AWS DL Containers 를 배포하고 이를 사용하여 여러 프레임워크에서 기계 학습 작업을 확장 및 가속화할 수 있습니다. Amazon ECS 및 Amazon EKS 는 가상 머신 클러스터에서 AWS DLContainers 를 배포하고 확장하는 데 필요한 모든 컨테이너 오케스트레이션을 처리합니다. 현재 AWS DL Containers는 TensorFlow 및 Apace MXNet에 사용할 수 있습니다.

CPU 및 GPU 지원, Python 2.7 및 3.6 지원, TensorFlow for Inference and Training에서 분산 훈련을 위한 Horovod 지원 등 다양한 설정의 컨테이너 이미지가 있습니다.

AWS DL Containers는 AWS Marketplace 또는 Amazon ECS 콘솔에서 사용할 수 있습니다.

AWS DL Containers 버전 2.0 for TensorFlow Docker 이미지는 Amazon SageMaker, Amazon EC2, Amazon ECS 및 Amazon EKS에서 테스트되었습니다. 단일 Docker 이미지를 AWS의 여러 플랫폼에 사용할 수 있습니다.

# 네트워킹

## 향상된 네트워킹

향상된 네트워킹에서는 SR-IOV(단일 루트 I/O 가상화)를 사용하여 지원되는 인스턴스 유형에서 고성능 네트워킹 기능을 제공합니다. SR-IOV는 기존의 가상 네트워크 인터페이스에 비해 높은 I/O 성능과 낮은 CPU 사용률을 제공하는 디바이스 가상화 방법입니다. 향상된 네트워킹을 통해 대역폭과 초당 패킷(PPS) 성능을 높이고, 인스턴스 간 지연 시간을 낮게 유지할 수 있습니다. 딥 러닝에 사용되는 대부분의 인스턴스 유형은 향상된 네트워킹을 제공하는 Elastic Network Adapter(ENA)를 지원합니다.

ENA는 C5, M5, P3 및 X1 인스턴스와 같은 최신 프로세서와 호환되도록 설계되었습니다. 이러한 프로세서에는 많은 수의 가상 CPU(X1의 경우 128개)가 있으므로 네트워크 어댑터와 같은 공유 리소스를 효율적으로 사용하는 것이 중요합니다. ENA는 높은 처리량과 뛰어난 PPS(초당 패킷) 성능을 제공하면서 여러 가지 방법으로 호스트 프로세서의 로드를 최소화하고 여러 vCPU에 패킷 처리 워크로드를 더욱 효과적으로 분산합니다. 다음은 이러한 향상된 성능을 실현하는 몇 가지 기능입니다.

- **체크섬 생성** – ENA는 하드웨어에서 IPv4 헤더 체크섬 생성과 TCP/UDP 부분 체크섬 생성을 처리합니다.
- **다중 대기열 디바이스 인터페이스** – ENA는 다중 전송 및 수신 대기열을 사용하여 내부 오버헤드를 줄이고 확장성을 개선합니다. 대기열이 여러 개 있으면 수신 및 발신 패킷을 특정 vCPU에 매핑하는 프로세스가 간소화되고 가속화됩니다.

- **수신 측 스티어링** – ENA는 수신되는 패킷을 적절한 vCPU로 전달하여 처리할 수 있습니다. 이 기술은 병목 현상을 줄이고 캐시 효율성을 높입니다.

이러한 모든 기능은 가능한 한 많은 워크로드를 프로세서 외부로 오프로드하고, 네트워크 패킷과 해당 패킷을 생성 또는 처리하는 vCPU 간에 짧고 효율적인 경로를 생성하도록 설계되었습니다.

## 배치 그룹(Placement Group)

배치 그룹은 Amazon EC2 인스턴스 간의 지연 시간을 줄이는 AWS 솔루션입니다. 동일한 가용 영역에서 실행되는 인스턴스를 최대한 가깝게 배치하여 지연 시간을 줄이고 처리량을 개선하는 메커니즘입니다.

## Elastic Fabric Adapter

Elastic Fabric Adapter(EFA) 는 고객이 AWS에서 높은 수준의 인스턴스 간 통신이 요구되는 HPC(고성능 컴퓨팅) 애플리케이션(예: 대규모 딥 러닝)을 실행할 수 있도록 지원하는 Amazon EC2 인스턴스용 네트워크 인터페이스입니다. 이 인터페이스는 맞춤형 운영 체제 우회 기술을 사용하여 인스턴스 간 통신 성능을 높이는데, 이는 HPC 애플리케이션을 확장하는 데 있어 매우 중요합니다. EFA를 사용하면 메시지 전달 인터페이스(Message Passing Interface, MPI)와 같은 인기 있는 HPC 기술을 사용하는 HPC 애플리케이션을 수천 개의 CPU 코어로 확장할 수 있습니다. EFA는 개방형 표준 libfabric API를 지원하므로, 지원되는 MPI 라이브러리를 사용하는 애플리케이션을 최소한으로만 수정하여 AWS로 마이그레이션할 수 있습니다. EFA는 추가 비용 없이 C5n.18xl 및 P3dn.24xl 인스턴스에서 활성화할 수 있는 선택적 EC2 네트워킹 기능으로 제공됩니다.

Open MPI 3.1.3(이상)을 사용하거나, NCCL용 OFI 드라이버를 통해 NCCL(2.3.8 이상)을 사용할 수 있습니다.

인스턴스에서 EFA 를 사용하여 VPC 서브넷 내에서 통신할 수 있으며, 보안 그룹에는 보안 그룹 내의 모든 트래픽 흐름을 허용하는 수신 및 송신 규칙이 있어야 합니다. 각 인스턴스에는 단일 EFA를 구축할 수 있으며, 이 EFA는 인스턴스가 시작되거나 중지될 때 연결할 수 있습니다.

## Amazon Elastic Inference

Amazon Elastic Inference 를 사용하면 저렴한 GPU 기반 가속을 Amazon EC2 및 Amazon SageMaker 인스턴스에 연결하여 딥 러닝 추론의 실행 비용을 최대 75% 절감할 수 있습니다. 현재 Amazon Elastic Inference 는 TensorFlow, Apache MXNet 및 ONNX 모델을 지원하며 곧 추가될 프레임워크에도 지원이 제공될 예정입니다. 다른 딥 러닝 프레임워크를 사용하는 경우, ONNX를 사용하여 모델을 내보낸 다음 MXNet으로 모델을 가져올 수 있습니다. 그런 다음 해당 모델을 Amazon Elastic Inference 에서 MXNet 모델로 사용할 수 있습니다.

Amazon Elastic Inference는 AWS의 향상된 TensorFlow 서비스 또는 Apache MXNet과 함께 사용하도록 설계되었습니다. 이러한 향상된 버전의 프레임워크는 Amazon SageMaker Python SDK를 사용할 때 컨테이너에 자동으로 구축되며, 이진 파일로 다운로드하여 자체 Docker 컨테이너로 가져올 수도 있습니다.

일반적으로, 모델이 복잡해서 Amazon SageMaker 사전 구축 컨테이너가 지원하지 않는 프레임워크로 확장해야 하는 경우가 아니면, 맞춤형 컨테이너를 생성할 필요는 없습니다.

Amazon Elastic Inference 액셀러레이터는 엔드포인트에서 Amazon EC2 인스턴스와 함께 작동하여 추론 호출을 가속화하는 네트워크 연결 디바이스입니다. 모델이 엔드포인트로서 배포되면 기계 학습 프레임워크가 Amazon EC2 인스턴스와 액셀러레이터 리소스의 조합을 사용하여 추론 호출을 실행합니다.

호스팅된 엔드포인트에서 Amazon Elastic Inference 를 사용하려면 필요에 따라 다음 중 하나를 사용할 수 있습니다.

- Amazon SageMaker Python SDK TensorFlow – TensorFlow를 사용하려 하며, 맞춤형 컨테이너를 구축할 필요가 없는 경우 사용합니다.
- Amazon SageMaker Python SDK MXNet – MXNet을 사용하려고 하며, 맞춤형 컨테이너를 구축할 필요가 없는 경우 사용합니다.
- Amazon SageMaker SDK for Python(Boto 3) - 맞춤형 컨테이너를 구축해야 하는 경우 사용합니다.

일반적으로, 모델이 복잡해서 Amazon SageMaker 사전 구축 컨테이너가 지원하지 않는 프레임워크로로 확장해야 하는 경우가 아니면, 맞춤형 컨테이너를 생성할 필요는 없습니다.

다음과 같은 Amazon Elastic Inference 액셀러레이터 유형이 지원됩니다. Amazon Elastic Inference 액셀러레이터 유형을 사용하여 엔드포인트 또는 노트북 인스턴스를 구성할 수 있습니다.

표 3: Elastic Inference 액셀러레이터 유형[2]

| 액셀러레이터 유형 | F32 처리량(TFLOPS) | F16 처리량(TFLOPS) | 메모리(GB) |
|---|---|---|---|
| **ml.eia1.medium** | 1 | 8 | 1 |
| **ml.eia1.large** | 2 | 16 | 2 |
| **ml.eia1.xlarge** | 4 | 32 | 4 |

호스팅된 모델에 적합한 액셀러레이터 유형을 선택할 때는 다음 요인을 고려하십시오.

- 모델, 입력 텐서 및 배치 크기에 따라 필요한 액셀러레이터 메모리 양이 결정됩니다. 최소한 훈련된 모델의 파일 크기와 같은 양의 메모리를 제공하는 액셀러레이터 유형으로 시작합니다.
- CPU 컴퓨팅 리소스, GPU 기반 가속화 및 CPU 메모리에 대한 수요는 딥 러닝 모델의 종류마다 크게 다릅니다. 애플리케이션의 지연 시간 및 처리량 요구 사항에 따라 필요한 컴퓨팅 및 가속화의 양도 결정됩니다. 다양한 인스턴스 유형 및

Amazon Elastic Inference 액셀러레이터 크기 구성을 철저히 테스트하여 애플리케이션의 성능 요구 사항에 가장 적합한 구성을 선택해야 합니다.

# 솔루션

## 데이터 레이블링을 위한 Amazon SageMaker Ground Truth

다른 형태의 기계 학습에 비해, 지도 학습이 기계 학습 분야를 계속 주도하고 있습니다. 모델 훈련 주기에 더 많은 데이터를 제공하면 기계 학습 모델 성능이 지속적으로 개선됩니다. 하지만 정확한 레이블로 훈련 데이터 세트를 구축하는 작업은 까다롭고 비용이 많이 소요됩니다.

Amazon SageMaker Ground Truth 는 데이터를 수집하고 레이블을 지정하는 기계 학습 프로세스의 첫 번째 단계에 도움이 됩니다. Amazon SageMaker Ground Truth는 능동 학습(active learning)을 기반으로 한 자동 데이터 레이블링 기술과 Mechanical Turk를 사용한 크라우드 소싱 데이터 레이블링 프로세스를 결합합니다. 능동 학습을 활용하여 학습할 속성을 식별한 다음, 크라우드 소싱된 인력을 활용하여 레이블링을 수행할 수 있습니다. 능동 학습은 모델을 훈련하는 데 필요한 레이블링된 데이터의 양을 크게 줄이는 방법론입니다. 이를 위해 전문가의 레이블링 작업 우선 순위를 정합니다.

능동 학습 모델은 레이블이 없는 데이터를 살펴보고 신뢰도를 기준으로 순위가 매겨진 답을 계산합니다. 그런 다음 모델은 가장 낮은 신뢰도 점수를 레이블이 지정된 데이터와 비교합니다. 마지막으로, 모델은 동일한 데이터가 다시 관측된 경우 정답을 계산할 가능성이 높아지도록 자체적으로 조정됩니다.

Amazon SageMaker Ground Truth 는 능동 학습 기능과 Mechanical Turk 인력에 대한 액세스 외에 레이블 관리 및 워크플로 관리 기능도 지원합니다. 선택적으로 레이블링 작업을 위한 프라이빗 및 하이브리드 인력도 설정할 수 있습니다.

## 모델 최적화를 위한 Amazon SageMaker Neo

훈련된 모델이 준비되면 클라우드, 엣지 또는 모바일 디바이스에 배포할 수 있습니다. 추론 요청은 클라이언트 HTTP 스택을 통해 네트워크에서 웹 서버 및 애플리케이션 서버 스택을 거쳐 최종적으로 추론 엔드포인트로 전달되어야 합니다. 위의 모든 계층에서 발생하는 지연 시간을 고려할 때, 추론을 컴퓨팅하고 그 결과를 클라이언트에 제공하는 과정에서 부분적으로 남는 시간이 있을 수 있습니다. 따라서 항상 추론 엔드포인트에서 최대 성능을 구현하는 것이 바람직합니다.

추론 엔드포인트의 성능을 개선하는 것은 복잡한 문제입니다. 첫째, 모델의 계산 그래프는 컴퓨팅 집약적 작업입니다. 둘째, 추론을 위한 기계 학습 모델을 최적화하려면 모델이 배포된 특정 하드웨어 및 소프트웨어의 구성을 튜닝해야 합니다. 최적의 성능을 실현하려면 하드웨어 아키텍처, 명령 세트, 메모리 액세스 패턴, 입력 데이터 셰이프 등을 파악해야 합니다.

기존 소프트웨어의 경우 컴파일러와 프로파일러가 튜닝을 처리합니다. 하지만 딥러닝 프로세스에서 이같은 튜닝작업은 수작업으로 수정과 확인을 반복해야 하는 번거로운 작업입니다.

Amazon SageMaker Neo 는 ARM, 인텔 및 NVIDIA 프로세서에 배포하기 위해 TensorFlow, Apache MXNet, PyTorch, ONNX 및 XGBoost 모델을 자동으로 최적화하여 특정 소프트웨어 및 하드웨어 구성에 맞게 모델을 튜닝하는 데 소요되는 시간과 노력을 없애 줍니다. 지원되는 딥 러닝 프레임워크, 모델 형식 및 칩셋은 앞으로 꾸준히 추가될 예정입니다.

Amazon SageMaker Neo 는 컴파일러와 런타임으로 구성됩니다. 먼저, Amazon SageMaker Neo API는 모델을 읽고 표준 형식으로 파싱합니다. 프레임워크별 함수 및 연산을 프레임워크에 구애받지 않는 중간 표현으로 변환합니다. 다음으로, 모델 그래프에서 일련의 최적화 작업을 수행합니다. 그런 다음, 최적화된 연산에 대한 이진 코드를 생성합니다. 또한 Amazon SageMaker Neo 는 컴파일된 모델을 로드하고 실행하는 데 사용되는 각 대상 플랫폼 및 소스 프레임워크를 위한 린 런타임을 제공합니다. 마지막으로,

Amazon SageMaker Neo 는 Apache 소프트웨어 라이선스에 따라 Neo-AI 프로젝트로 오픈 소스 코드로도 사용할 수 있으므로 다양한 디바이스와 애플리케이션에 맞게 소프트웨어를 사용자 지정할 수 있습니다.

# 코드, 데이터 및 모델 버전 관리

## Git을 사용한 코드 버전 관리

훈련, 전 처리 및 추론 스크립트는 전체 딥 러닝 스택에서 가장 작은 구성 요소입니다. 위의 스크립트 외에, AWS Step Functions와 같은 서비스를 사용하여 모델 재훈련 및 모델 배포를 위한 딥 러닝 파이프라인을 스크립팅할 수도 있습니다. Git 기반 리포지토리를 사용하거나, 안전한 Git 기반 리포지토리를 호스팅하는 완전관리형 소스 제어 서비스인 AWS CodeCommit을 사용하여 위의 모든 스크립트의 버전을 관리할 수 있습니다.

## Amazon S3에서의 데이터 버전 관리

딥 러닝에서 모델 재훈련에 사용된 데이터의 사본은 모델의 편향과 드리프트를 설명하고 문제를 해결하는 데 있어 중요한 역할을 합니다. Amazon S3 를 사용하여 훈련 데이터의 버전을 지정할 수 있습니다. 모든 신규 또는 업데이트된 훈련 데이터 세트에 대해 새 Amazon S3 객체 또는 Amazon S3 객체의 새 버전을 생성합니다. Amazon S3 객체의 명명 규칙을 사용하거나 객체 태그를 사용하여 여러 훈련 데이터 세트 버전을 추적할 수 있습니다. 선택적으로, Amazon DynamoDB 테이블로 데이터 세트 S3 객체 위치 및 데이터 메타데이터를 푸시하고, 테이블을 인덱싱하여 데이터 검색을 지원할 수 있습니다.

## Amazon S3에서의 모델 버전 관리

모델 훈련에는 비용과 시간이 많이 소요됩니다. 새 버전의 모델 성능과 정확성을 비교할 수 있도록 원본 모델을 유지하는 것이 중요합니다. 훈련된 모델은 버전 관리가 활성화된 데이터 파일로서 Amazon S3에 유지할 수 있는 특수한 데이터 파일이라고 할 수 있습니다. Amazon S3를 사용하면 모든 유형의 데이터 파일에 태깅하고 버전을 지정할 수 있습니다.

# 재훈련 및 재배포를 위한 딥 러닝 프로세스 자동화

기능 프로토타입을 시연한 후에는 모델을 프로덕션 환경에 배포하고 훈련된 모델을 사용하여 예측을 제공하기 위한 엔드포인트를 생성할 차례입니다. 프로토타입 작성 과정에서 구축, 훈련 및 배포의 모든 단계는 Jupyter 노트북에서 수동으로 수행됩니다. 하지만 프로덕션 환경에서는 배포 시 정밀도, 일관성 및 안정성이 요구됩니다. 프로덕션 파이프라인에 수동으로 개입하면 인적 오류가 발생하여 가동 중지가 발생할 수 있습니다. 재훈련 및 재배포와 관련된 모든 단계를 자동화함으로써 이 같은 인적 오류를 해결할 수 있습니다. 아래에서는 AWS에서 딥 러닝 프로덕션 파이프라인을 자동화하는 데 사용할 수 있는 솔루션 및 서비스를 설명합니다.

## Amazon SageMaker를 위한 AWS Step Functions

AWS Step Functions 를 사용하면 기계 학습 워크플로의 여러 단계를 오케스트레이션하여 프로덕션 환경에서 모델 배포를 원활하게 수행할 수 있습니다. AWS Step Functions 는 워크플로를 알기 쉽고, 다른 사용자에게 설명하기 쉬우며, 변경하기 쉬운 상태 시스템 다이어그램으로 변환합니다. 또한, 발생하는 각 실행 단계를 모니터링할 수 있습니다.

현재 Amazon SageMaker 는 두 가지 서비스 통합 패턴을 지원합니다.

- Amazon SageMaker 인스턴스를 호출하고 HTTP 응답을 수신한 직후 AWS Step Functions 가 다음 상태로 진행되도록 합니다.
- Amazon SageMaker 인스턴스를 호출하고 AWS Step Functions 가 작업이 완료될 때까지 대기하도록 합니다.

## Amazon SageMaker를 위한 Apache Airflow

Apache Airflow 는 프로그래밍 방식으로 워크플로를 작성, 예약 및 모니터링할 수 있는 오픈 소스 기반의 대체 플랫폼입니다. Apache Airflow를 사용하면 Amazon SageMaker 훈련,

하이퍼파라미터 튜닝, 배치 변환 및 엔드포인트 배포를 위한 워크플로를 구축할 수 있습니다. 원하는 Amazon SageMaker 딥 러닝 프레임워크 또는 Amazon SageMaker 알고리즘을 사용하여 Airflow에서 이러한 작업을 수행할 수 있습니다.

Airflow SageMaker 연산자 또는 Airflow Python 연산자를 사용하여 Amazon SageMaker 워크플로를 구축할 수 있습니다.

오픈 소스 AWS CloudFormation 템플릿인 Turbine을 사용하여 AWS에 Airflow 리소스 스택을 생성할 수도 있습니다.

## Kubernetes의 Kubeflow Pipelines

Amazon SageMaker를 사용하지 않는 DIY 고객이 기존에 투자한 Kubernetes를 AWS에서 활용하는 경우 Kubeflow Pipelines를 사용할 수 있습니다. Kubeflow Pipelines는 Docker 컨테이너를 기반으로 이식성이 높고 확장성이 뛰어난 기계 학습 워크플로를 구축 및 배포하는 플랫폼입니다. 파이프라인은 워크플로 내의 모든 구성 요소와 구성 요소간 상호 연동을 그래프 형식으로 표시한 정보 등의 기계 학습 워크플로를 기술하기 위한 도구입니다. 실무자가 Kubernetes를 활용한 구축, 훈련 및 배포에 사용하는 인기 있는 도구로, Kubernetes와 기본적으로 통합됩니다.

또한 Amazon SageMaker 는 물론, Amazon EMR , Amazon Athena 등 데이터 정리 및 변환에 사용되는 다른 AWS 서비스와도 통합되는 Kubeflow용 AWS 파이프라인 구성 요소도 있습니다. 이 접근 방식은 통합 제어 플레인(마이크로서비스 아키텍처와 인공 지능/기계 학습 서비스 릴리스 통합)을 원하면서 Amazon EKS, Amazon FSx for Lustre 및 Amazon SageMaker 등 작업에 가장 적합하고 획일적이고 부담스러운 작업을 처리하는 데 도움이 될 수 있는 다양한 AWS 서비스를 활용하고자 하는 고객에게 적합합니다.

# 대규모 딥 러닝을 위한 패턴

이 섹션에서는 조직에서 딥 러닝 도입을 확대하는 데 활용할 수 있는 다양한 솔루션과 패턴을 설명합니다. 이 같은 옵션에는 완전관리형, AWS 서비스를 사용한 DIY 또는 하이브리드 접근 방식이 있습니다. 다른 AWS 서비스를 사용하여 맞춤형 환경을 구축하는 데에도 이러한 옵션을 활용할 수 있습니다.

## AWS 기반의 딥 러닝을 위한 옵션

이 섹션에서는 AWS 기반의 딥 러닝을 위한 솔루션으로서 완전관리형 플랫폼과 DIY 플랫폼에 대해 설명합니다. 다음 그림과 같이, AWS는 AWS와 고객 간의 서로 다른 수준의 공동 책임에 따라 딥 러닝을 위한 광범위한 솔루션을 포괄하는 다양한 옵션을 제공합니다.


그림 6: 공동 책임 모델 스펙트럼으로 본 딥 러닝 솔루션

스펙트럼의 왼쪽에는 이벤트 기반 딥 러닝을 위한 완전관리형 서비스인 Amazon SageMaker가 있습니다. 이 서비스는 클릭 한 번으로 구축, 훈련, 배포할 수 있는 환경을 제공합니다.
스펙트럼의 오른쪽에는 자체 관리형 완전 맞춤형 수동 딥 러닝 솔루션이 있으며, 이 솔루션은 구축, 훈련 및 배포를 위한 여러 단계를 수반합니다. 이 두 옵션 사이에는 Amazon EKS와 같은 관리형 서비스를 사용하여 인프라 관리를 간소화하는 동시에 딥 러닝 워크플로를 지원할 자체 관리형 Kubeflow를 배포할 수 있는 부분 관리형 솔루션도 있습니다.

자체 관리형 DIY 솔루션과 맞춤형 DIY 솔루션의 차이점에 대해 자세히 알아보기 위해 딥 러닝 환경의 일회성 설정과 관련된 운영 단계를 살펴보겠습니다.

DIY 환경에서는 딥 러닝 엔지니어와 과학자에게 딥 러닝 환경의 가용성을 보장하기 위해 다음 운영 문제를 관리해야 합니다.

- 드라이버 설치
- 업데이트 조율
- 네트워크 고장
- 예약되지 않은 재부팅
- 정전
- 장비 장애
- 응답 문제
- 디스크 공간 부족
- 케이블 문제
- VPN을 통해 서버에 연결할 수 없음
- 서버의 보호되지 않은 물리적 경계

Amazon SageMaker와 같은 AWS 관리형 서비스의 경우, 모두 내결함성을 보장하고 설계상 가용성이 높기 때문에 운영 문제가 발생할 가능성이 최소화됩니다.

예를 들어 Amazon EC2를 기반으로 딥 러닝 환경을 완전히 새로 구축하는 맞춤형 DIY 환경에서는 TensorFlow 및 Horovod로 분산 훈련을 하고 가치를 창출하기 위해서는 다음 작업들을 수행해야 합니다. 이러한 작업에는 일회성 소프트웨어 설치, 지속적인 작업 예약 작업 등이 있습니다.

1. 헤드 노드에 SSH로 로그인
2. MPI 설치
3. Anaconda, Python 3 설치

4. 가상 환경 생성
5. MKL을 사용하여 인텔 최적화 TensorFlow 설치
6. 분산 훈련용 Horovod 설치
7. SLURM 작업 스크립트 작성(SLURM은 HPC 작업을 지원하는 오픈 소스 스케줄링 소프트웨어)
8. SLURM 작업 제출
9. 인사이트 확보

Amazon SageMaker와 같은 AWS 관리형 서비스의 경우, 위의 모든 단계는 자동화되어 있으며 AWS Management Console에서의 클릭 한 번, 단일 API 호출, 단일 CLI 명령 또는 이벤트에 의해 트리거됩니다.

각 솔루션을 자세히 살펴보면서 각각의 구성 요소와 가치를 알아보겠습니다.

## 완전관리형 솔루션 – Amazon SageMaker 사용

조직 전체에 걸쳐 딥 러닝을 확장할 수 있는 솔루션을 찾는 고객에게 Amazon SageMaker는 딥 러닝 프로세스와 관련된 다양한 단계를 포괄적으로 지원하는 솔루션을 제공합니다. Amazon SageMaker는 완전관리형 서비스의 형태로 기본 지원 기능을 제공할 뿐만 아니라, 최신 드라이버와 프레임워크를 활용하도록 딥 러닝 스택을 맞춤형으로 구성할 수 있는 유연성도 제공합니다. Amazon SageMaker가 제공하는 편의성과 유연성은 프레임워크 수준에서 작업하는 고급 딥 러닝 엔지니어와 딥 러닝 과학자, 딥 러닝 프로젝트에 참여하는 데이터 사이언티스트 및 개발자의 요구 사항을 충족합니다.

다음 차트는 다양한 Amazon SageMaker 구성 요소가 딥 러닝 환경에 어떻게 적용되어 포괄적인 딥 러닝 프로세스를 제공하는지 보여줍니다.

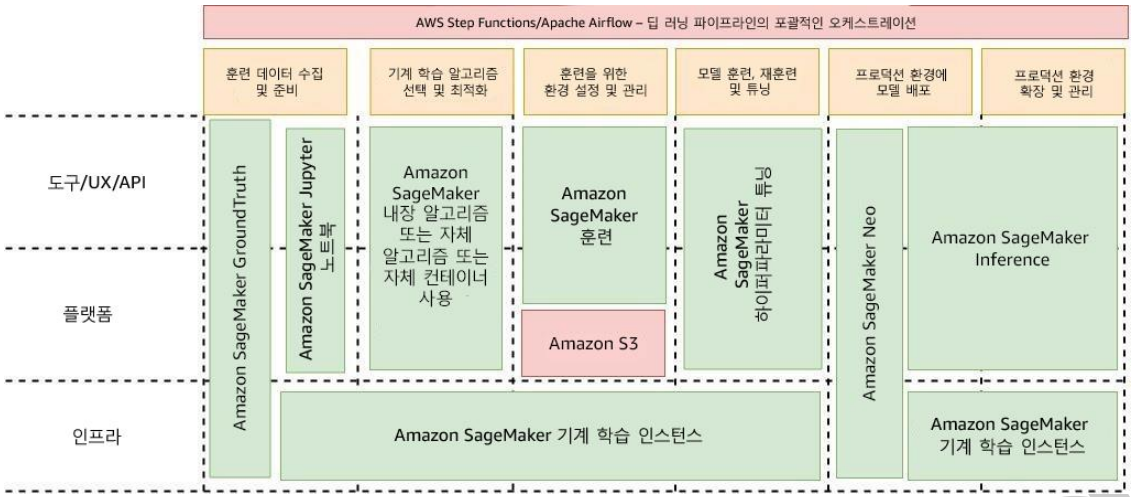


그림 7: 딥 러닝을 위한 Amazon SageMaker 솔루션

Amazon SageMaker는 딥 러닝 환경을 구축, 훈련 및 배포하는 데 필요한 인프라 생성 및 종료 작업을 완전히 추상화합니다. NCCL, CUDA 및 CUDNN과 같은 공용 드라이버 설치, 공용 데이터 처리 라이브러리 설치, 현재 딥 러닝에 사용되는 공용 프레임워크의 설치 작업 등을 인프라의 일부로서 처리합니다. 노트북, 훈련 및 추론 환경 생성 기능은 API 호출, CLI 명령, Amazon SageMaker SDK 또는 AWS Management Console의 옵션으로 제공됩니다. Amazon SageMaker를 사용하면 일련의 작업을 ML 파이프라인의 자동화된 워크플로로 자동화할 수 있습니다(예: 일관성 있는 모델의 새로운 버전 재훈련 및 재배포).

Amazon SageMaker는 데이터 레이블링, 하이퍼파라미터 튜닝, 실시간 및 배치 추론과 같은 일반적인 딥 러닝 작업을 간소화합니다. 이러한 작업을 간소화하면 제어 능력을 잃지 않으면서 조직 내 딥 러닝 도입을 가속화하고 확장할 수 있습니다. 완전관리형 표준 도구 세트를 사용하면 딥 러닝 프로젝트의 협업을 개선하고 출시 시간을 단축할 수 있습니다.

Amazon SageMaker는 조직의 딥 러닝 엔지니어와 과학자 및 DevOps 전문가 모두에게 단일 플랫폼을 제공하여, 알고리즘을 설계하고 데이터를 바탕으로 모델을 훈련하는 모델 작성자와 모델 배포 및 모니터링을 담당하는 DevOps 팀 간에 원활한 협업이 가능합니다.

Amazon SageMaker는 널리 사용되는 대부분의 프레임워크에 사전 구축된 컨테이너 이미지를 제공합니다. 기존 Amazon SageMaker 컨테이너 이미지를 확장하거나 자체 이미지를 구축할 수 있습니다. 프레임워크 수준에서 작업하는 고급 딥 러닝 엔지니어와 과학자는 TF(TensorFlow)와 같은 사용자 지정 DL 프레임워크를 사용해 특정 사용 사례의 딥 러닝 훈련을 가속화하기 위해 맞춤형 연산자를 시도하거나 성능 개선을 위해 단일 인스턴스에서 2개의 TF 프로세스를 실행할 수 있습니다. Amazon SageMaker를 사용하면 스크립트 모드로 환경을 구성 및 사용자 지정하고 자체 컨테이너 모드를 가져올 수 있습니다. 스크립트 모드를 사용하면 사용자 지정 스크립트(예: TF 스크립트)를 가져와 사전 구축된 TF AWS DL 컨테이너에서 실행할 수 있습니다. 기존 보유 컨테이너를 사용하면 사용자 지정 TF 빌드로 컨테이너를 완전히 새로 구축하여 Amazon SageMaker에서 실행할 수 있으므로 유연성과 제어 능력이 극대화됩니다.

## DIY 부분 관리형 솔루션: AWS에서 Kubeflow와 함께 Kubernetes 사용

이 패턴은 Kubernetes를 인프라 계층으로 표준화한 고객이 Kubernetes에 대한 기존 투자를 활용하여 딥 러닝 훈련 및 추론 작업을 실행하려는 경우에 적용됩니다. 이 설정을 적용할 경우 Kubernetes를 관리해야 하는 획일적이고 부담스러운 작업이 많아지고 운영 복잡성이 커집니다. Kubernetes 제어 플레인을 관리하기 어려울 수 있습니다. 하지만 Kubernetes 제어 플레인 관리를 간소화하기 위해 Kubernetes용 관리형 서비스인 Amazon EKS를 사용하여 AWS로 Kubernetes 마스터 관리 작업을 오프로드할 기회가 있습니다.

Amazon EKS Kubernetes 클러스터는 영구 클러스터라는 점에 유의해야 합니다. 하지만 고객은 훈련 및 추론작업을 성공적으로 완료하기 위해 충분한 리소스를 제공할 수 있도록 온디맨드로 자원을 확장하거나 축소하는 정책을 적용할 수 있습니다. 훈련작업은 Kubernetes 클러스터 인프라에서 동작하며, 클러스터에 할당된 리소스를 소비합니다. 컴퓨팅 파워가 더 필요한 경우 Kubernetes 클러스터에 작업자 노드를 추가하여 용량 요구 사항을 충족해야 합니다.

노드 수준 AutoScaler(Kubernetes 추가 기능)를 활용하여 GPU 노드를 확장(인/아웃)할 수 있습니다.

추론 엔드포인트를 확장하는 데에는 Kubernetes 수준에서 수평 포드(POD) 확장 기능을 사용할 수 있습니다. POD 사이드카 패턴을 이용하여 TensorBoard와 같은 훈련 시각화를 위한 추가 소프트웨어를 배포할 수도 있습니다. Kubernetes POD는 동일한 호스트에 함께 배포되는 컨테이너 그룹입니다.

Kubernetes 클러스터는 컨테이너 환경에서 딥 러닝 훈련 및 추론을 실행하기 위한 인프라 계층을 제공합니다. 하지만 이 환경을 딥 러닝 프로젝트에 쉽게 사용하려면 더 많은 구성 요소를 배포하고 관리해야 합니다.

Kubernetes와 완벽하게 통합되고 딥 러닝 엔지니어와 과학자가 Kubernetes에서 구축, 훈련 및 배포할 수 있는 통합 API 플랫폼을 제공하는 Kubeflow를 사용할 수 있습니다. Kubeflow는 AWS의 관리형 서비스가 아닙니다. Kubeflow는 AWS의 Amazon EKS 또는 Kubernetes에서 직접 배포하고 관리해야 하는 Kubernetes 추가 기능 패키지입니다.

처음에 Kubernetes 및 Kubeflow는 TensorFlow만 지원했습니다. 하지만 MXNet, PyTorch, Chainer 등 다른 프레임워크로 지원이 확대되고 있습니다. 커뮤니티에서 Kubeflow를 자주 업데이트하므로 Kubeflow의 최신 버전을 적용하려면 스크립트를 자주 변경해야 할 수도 있습니다.

스토리지 계층의 경우 Amazon S3, Amazon EFS 또는 Amazon FSx for Lustre(Amazon S3의 데이터에서 하이드레이션됨)를 선택할 수 있습니다. 딥 러닝과 최고 수준의 성능을 지원하려면 Amazon S3의 데이터를 하이드레이션하여 Amazon FSx for Lustre를 사용하는 것이 좋습니다. 하지만 분산 훈련에 사용되는 노드 수에 따라 Amazon S3 및 Amazon EFS를 사용하도록 선택할 수도 있습니다. Amazon S3 및 Amazon EFS를 사용할 경우 Amazon FSx for Lustre만큼 성능이 높지 않습니다.

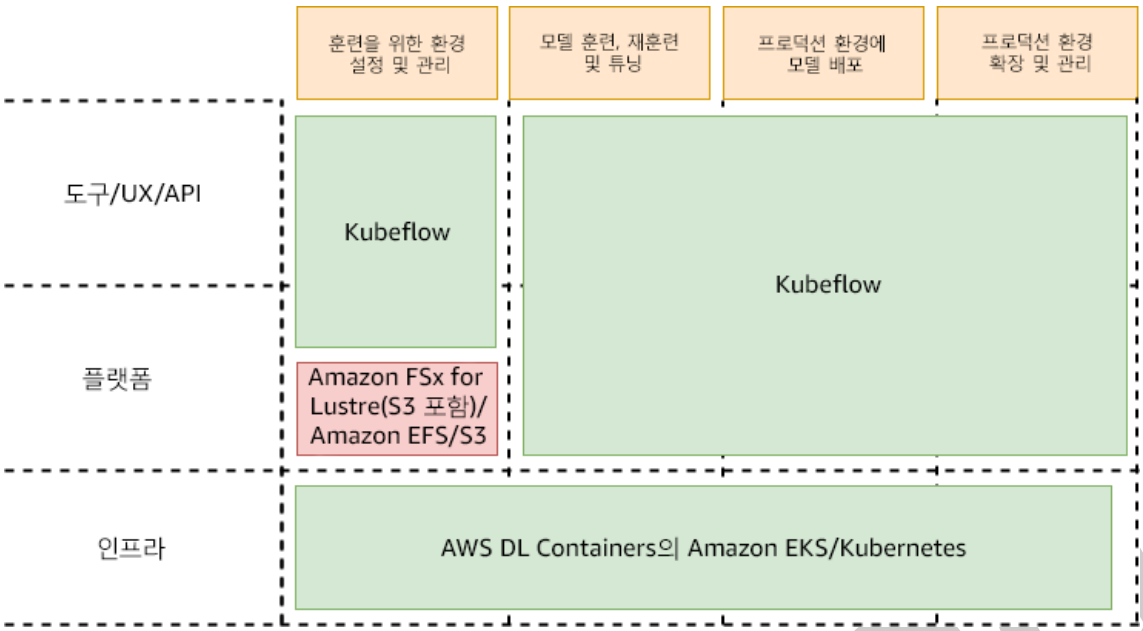


그림 8: Amazon EKS 및 자체 관리형 Kubeflow DL 계층 및 DL 프로세스 매핑

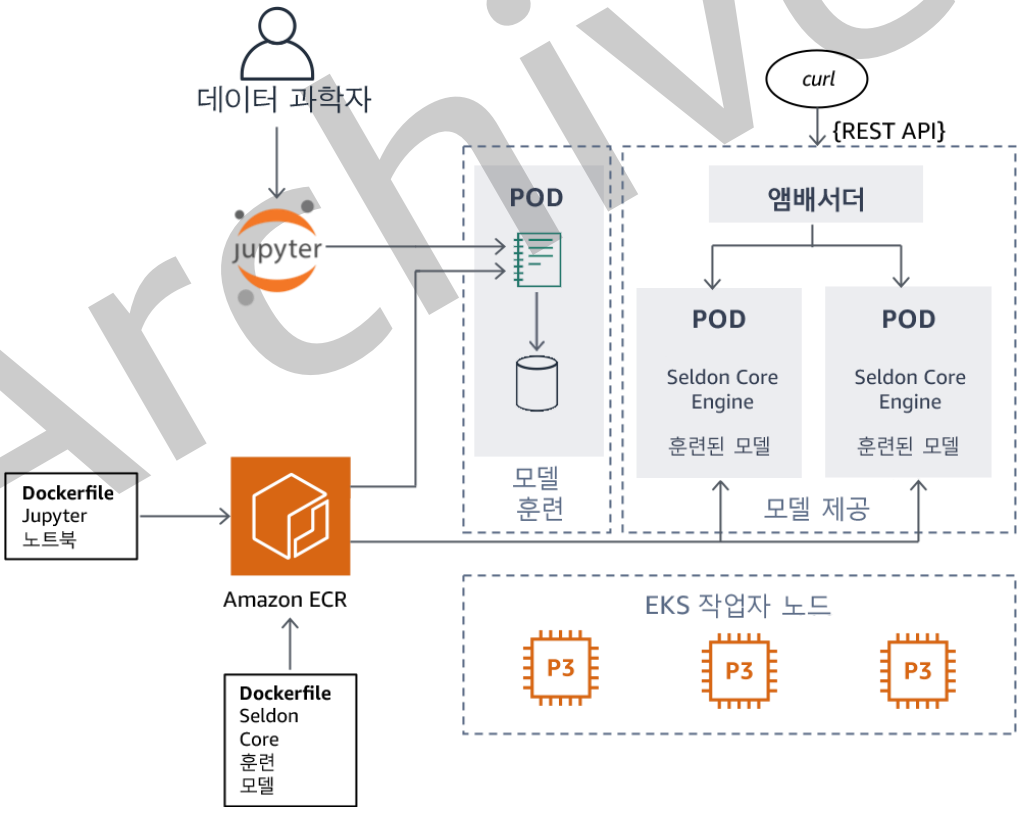


그림 9: 자체 관리형 Kubeflow를 사용하는 Amazon EKS개념 다이어그램[3]

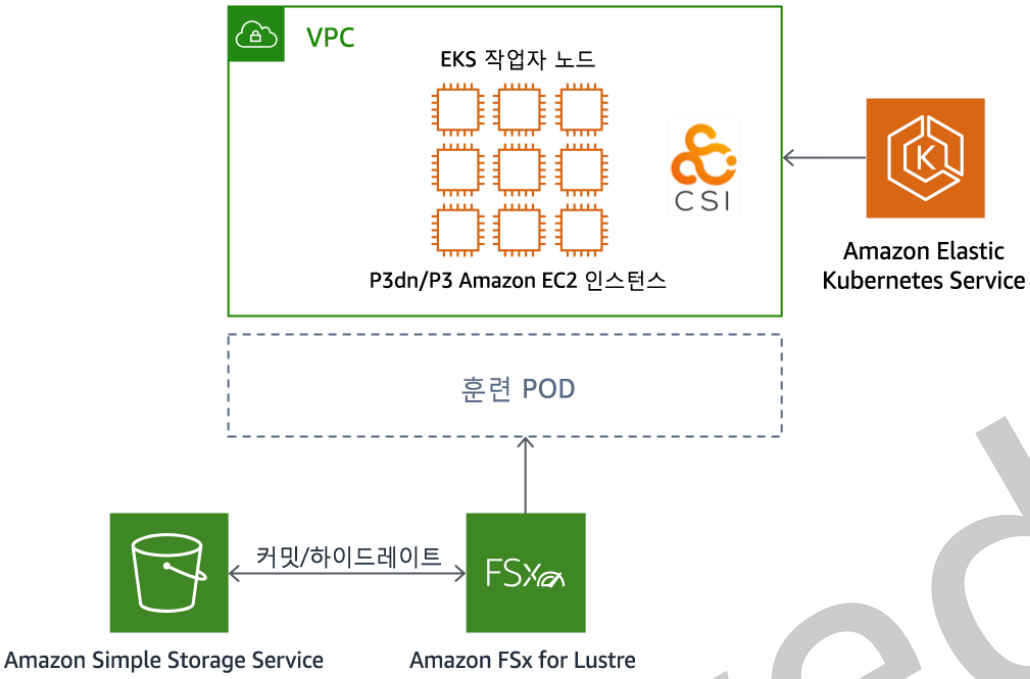


그림 10: Amazon S3에서 하이드레이션된 Amazon FSx for Lustre를 사용하는 Amazon EKS

Amazon SageMaker 구성 요소를 이 스택과 함께 데이터 레이블링 및 모델 최적화와 같은 작업에 사용할 수 있습니다. 포괄적인 딥 러닝 워크플로 또는 워크플로의 일부를 구현하는 데 Amazon SageMaker 를 사용할 수 있습니다. Kubernetes 및 Kubeflow 스택에서는 데이터 레이블링 및 주석 작업에 Amazon SageMaker Ground Truth 를 사용하고 모델 최적화에 Amazon SageMaker Neo 를 사용할 수 있습니다.

### DIY 솔루션의 추가 고려 사항

Amazon EKS는 딥 러닝용으로 특별히 튜닝되거나 최적화되지 않은 관리형 서비스입니다. Kubeflow는 AWS에서 제공하는 관리형 서비스가 아닙니다. 모범 사례를 구현하여 Amazon EKS 및 Kubeflow 스택을 딥 러닝에 적합하도록 미세 조정하고 최적화해야 합니다. 자세한 내용은 AWS 오픈 소스 블로그에서 Best Practices for Optimizing Distributed Deep Learning Performance on Amazon EKS 를 참조하십시오.

선택적으로, 기계 학습 벤치마킹을 위한 자동화된 도구인 Amazon EKS Deep Learning Benchmark Utility를 Kubernetes 클러스터에 사용할 수도 있습니다.

선택적으로, TensorFlow 및 EKS의 MXNet에서 모델을 훈련하고 제공하기 위한 Docker 이미지 세트인 AWS Deep Learning Containers(AWS DL Containers)를 사용할 수도 있습니다. AWS DL Containers 는 TensorFlow 및 MXNet, NVIDIA CUDA(GPU 인스턴스용) 및 인텔 MKL(CPU 인스턴스용) 라이브러리를 사용하는 최적화된 환경을 제공합니다. AWS DL Containers 는 Amazon Elastic Container Registry(Amazon ECR)에서 사용할 수 있습니다.

Kubeflow와 딥 러닝용 AWS 플랫폼 간의 기본 통합을 지원하기 위한 수많은 이니셔티브가 있습니다. Kubeflow와 AWS 간의 기본 통합 기능의 전체 목록은 AWS의 Kubeflow 기능 섹션을 참조하십시오.

## DIY 자체 관리형 솔루션: Amazon EC2 사용

조직과 딥 러닝 엔지니어 및 과학자가 구축, 훈련 및 배포를 위한 컨테이너 전략을 도입하지 않을 수도 있고, 컨테이너식 환경에서 운영하는 데 필요한 기술을 갖추지 못했을 수도 있습니다. 또는 딥 러닝 엔지니어와 과학자가 연구 커뮤니티의 최신 드라이버 및 라이브러리를 사용하려고 할 수도 있으며, 컨테이너식 환경에 설치하기 위한 지침이 없을 수도 있습니다. 또한, 이 새로운 소프트웨어의 설치와 통합이 불가능하거나 쉽지 않을 수 있습니다.

이러한 유형의 시나리오에서는 Amazon EC2 를 기반으로 사용자 지정 DIY 클러스터를 설정하여 실험을 개발하고 확장할 수 있습니다.

모든 구성 요소를 완전히 새롭게 설정할 수도 있지만, 아래에서 설명하는 몇 가지 도구와 서비스를 이용하면 AWS에서 손쉽게 설치하고 훈련을 가속화할 수 있습니다.

고객은 GPU 드라이버와 널리 사용되는 프레임워크 및 라이브러리가 사전에 설치되어 있는 Deep Learning AMI 를 활용할 수 있습니다. 고객은 이 환경을 완벽하게 제어할 수 있으며 필요에 따라 손쉽게 사용자 지정할 수 있습니다. 또한 이 환경에서 사용할 CUDA 드라이버 버전을 쉽게 변경할 수 있습니다.

스토리지의 경우, 훈련 환경의 데이터 스토리지 계층으로 Amazon S3, Amazon EFS 또는 Amazon FSx for Lustre를 사용할 수 있습니다. Amazon FSx for Lustre, Amazon S3 및 Amazon EFS 중에서 선택할 때는 가격과 성능의 균형을 고려합니다.

이 패턴은 고유한 요구 사항이 적용되는 일회성 프로젝트에 유용하지만, 엔터프라이즈용 딥 러닝 플랫폼의 대안으로 고려해서는 안 됩니다. 다음 그림에서 그 차이를 명확하게 확인할 수 있습니다.

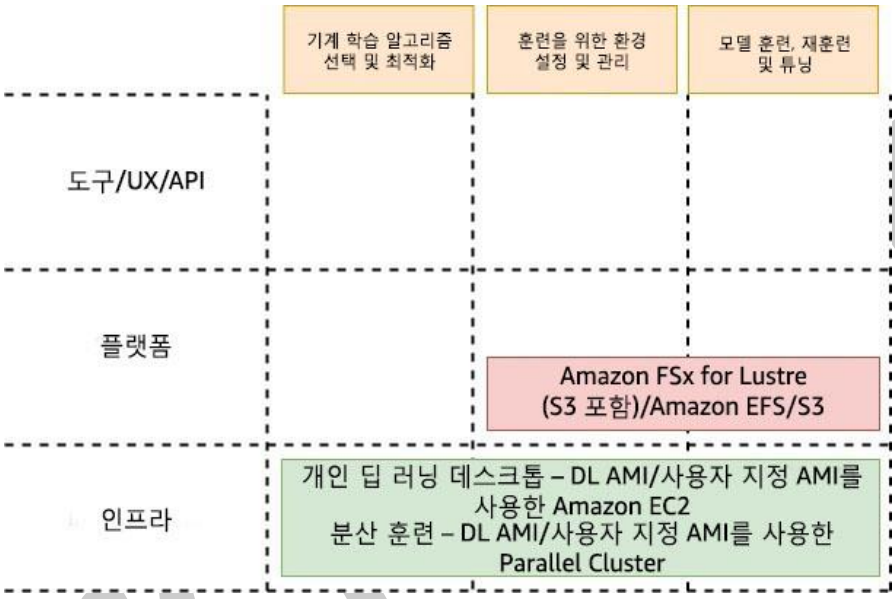


그림 11: DL AMI DL 계층 및 기계 학습 프로세스 매핑을 사용하는 Amazon EC2

Amazon EC2를 사용하는 맞춤형 환경에서, 다중 노드에서 대규모 데이터 세트에 대한 실험을 진행할 수 있도록 확장하려는 경우, AWS Parallel Cluster와 같은 도구를 사용하여 클러스터를 설정할 수 있습니다. AWS Parallel Cluster는 과학자, 연구원 및 IT 관리자가 AWS 클라우드에서 HPC(고성능 컴퓨팅) 클러스터를 손쉽게 배포하고 관리할 수 있도록 지원하는 오픈 소스 클러스터 관리 도구입니다. HPC 워크로드를 실행 중인 사용자에게 익숙한 클러스터 환경을 시작하기 위해 AWS Parallel Cluster와 많은 AWS 클라우드 네이티브 제품이 사용됩니다.

SLURM 과 같은 작업 스케줄러를 사용하여 클러스터에서 훈련 작업을 예약할 수 있습니다. 다음 그림은 스토리지 티어로 Amazon EFS와 함께 사용할 때 이 설정이 어떻게 표시되는지

보여줍니다. Amazon P3dn 인스턴스를 Elastic Network Adapter(ENA) 및 Amazon S3에서 하이드레이션된 Amazon FSx for Lustre와 함께 사용하면 클러스터의 성능을 더욱 개선할 수 있습니다.

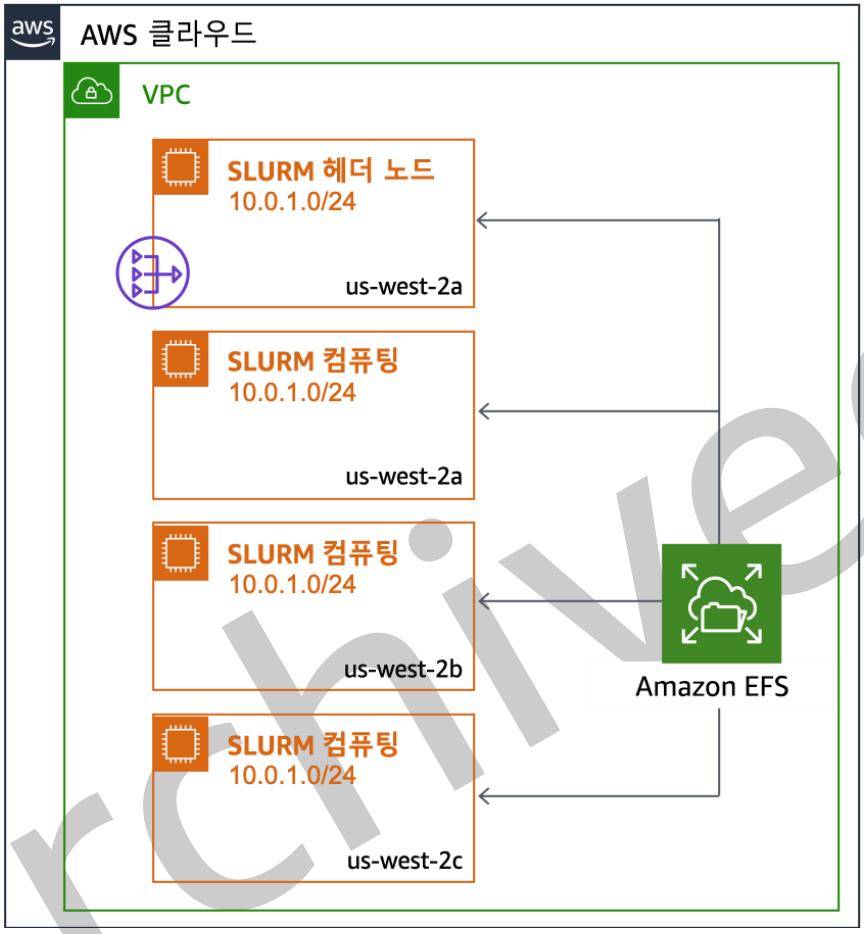


그림 12: Amazon EC2 및 Amazon EFS를 사용하는 SLURM 아키텍처



## 고급 사용 사례: Amazon SageMaker를 다른 AWS 서비스와 함께 사용

Amazon SageMaker가 제공하는 딥 러닝 솔루션의 기능을 확장하기 위해 다른 AWS 서비스를 활용해야 하는 고급 사용 사례도 있습니다. 이 섹션에서는 Amazon SageMaker와 기타 AWS 서비스를 사용하는 몇 가지 고급 사용 사례를 살펴봅니다.

## AWS Step Functions를 사용하여 포괄적인 기계 학습 파이프라인 오케스트레이션

AWS Step Functions 를 사용하면 복원력이 뛰어난 서버리스 워크플로를 구축할 수 있습니다. AWS Step Functions에서 워크플로는 유한 상태 시스템으로 구현됩니다. 상태는 '작업', '선택', '로직 브랜치', '병렬 작업 세트', '오류 핸들러' 등이 될 수 있습니다. 이 워크플로는 방향성 비순환 그래프(Directed Acyclic Graph, DAG)로 구현되며 GoTo 로직을 사용합니다. 또한 AWS Step Functions 를 사용하면 예외를 발생시키고 오류 처리를 수행하여 흐름을 더욱 견고하게 만들 수 있습니다.

AWS Step Functions에서 작업 상태는 대부분의 번거로운 작업을 수행합니다. 작업 상태는 활동 작업과 Lambda 작업의 두 가지 유형이 있습니다. 활동 작업에서 작업자는 AWS Step Functions에 작업을 요청한 다음 작업을 가져와 결과를 반환합니다. Lambda 작업은 AWS Step Functions에서 AWS Lambda 함수로 전달되는 동기식 호출입니다. Lambda 작업의 최대 제한 시간은 Lambda 함수의 최대 실행 기간에 정의된 대로 15분입니다. 또한 AWS Step Functions 에서는 승인 및 거부와 같은 인적 작업을 상태 시스템에 삽입할 수 있습니다. 워크플로에 이러한 작업을 사용하여 프로덕션 환경으로의 모델 푸시를 승인하거나 거부할 수 있습니다.

AWS Step Functions의 모든 기능을 사용하여 복잡하고 포괄적인 딥 러닝 워크플로를 구축할 수 있습니다. 새 데이터가 Amazon S3에 도달할 때 워크플로를 트리거하고, 훈련 작업을 시작하고, 새로 훈련된 모델을 배포할 수 있습니다. 알림과 오류 처리를 추가하여 워크플로를 보다 견고하고 투명하게 만들 수 있습니다. 다음 워크플로 다이어그램은 재훈련 및 재배포를 위해 AWS Step Functions 를 사용하여 구현된 포괄적인 딥 러닝 워크플로의 샘플 표현입니다.

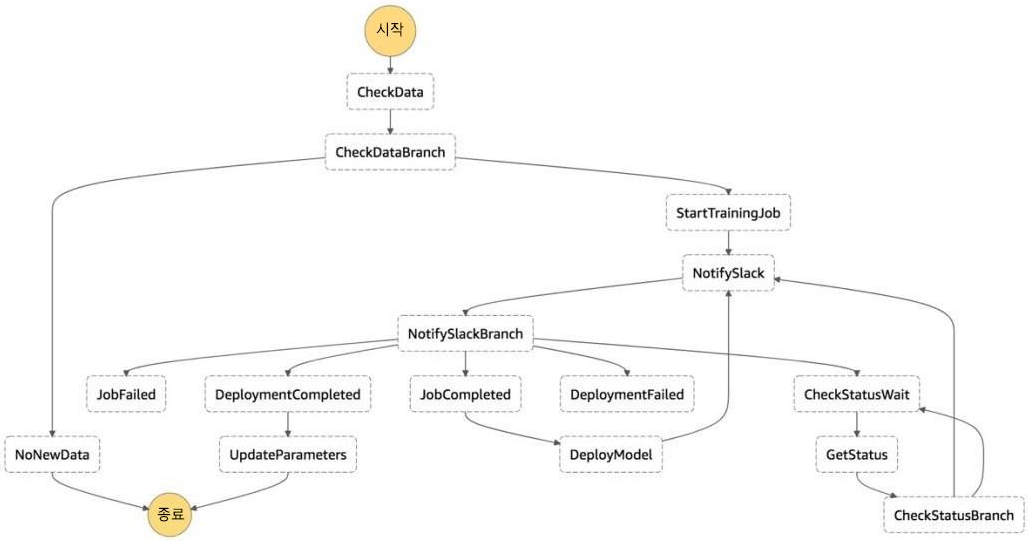


그림 13: 재훈련 및 재배포를 위한 워크플로

## 여러 AWS 리전에서 Amazon SageMaker를 백엔드로 사용하여 AWS Batch로 하이퍼스케일 딥 러닝 작업 오케스트레이션

일부 고객은 비용, 성능, 또는 규제 문제로 인해, 데이터가 생성된 리전의 독립적인 경계 내에서 로컬 데이터를 유지해야 하는 대용량 데이터 세트에 대해 훈련이 필요한 경우들이 있습니다. 이 데이터는 로컬로 생성된 자율 주행 차량의 4K 비디오 데이터 또는 로컬에서 생성되어 가장 가까운 AWS 리전으로 로컬로 전송되고 동일한 리전 내에서 레이블링된 캠페인 데이터일 수 있습니다. Amazon SageMaker를 사용하여 동일한 리전에서 로컬로 모델을 훈련할 수 있습니다. 선택적으로 여러 Amazon SageMaker 훈련 작업을 시작하여 각 리전에서 병렬로 훈련할 수 있습니다. AWS Batch 를 사용하여 여러 AWS 리전의 Amazon SageMaker 에서 실행되는 여러 작업을 중앙 리전에서 오케스트레이션하고 모니터링할 수 있습니다.

이 이벤트 중심 아키텍처는 온프레미스 환경에서 가장 가까운 AWS 리전으로 데이터가 업로드되면 훈련 작업을 트리거합니다.

Amazon S3로 들어오는 데이터를 한 중앙 공간에서 관계 테이블로 생성할 수 있습니다. 이 중앙 테이블에는 서로 다른 지리적 위치에서 실행되는 다양한 캠페인에서 소싱된 모든

데이터 파일의 인덱스가 유지됩니다. 이 중앙 테이블에서 쿼리를 실행하여 AWS Batch 배열 작업을 생성할 수 있습니다. AWS Batch 배열 작업은 일반 작업과 마찬가지로 제출됩니다. 하지만, 배열에서 실행해야 하는 하위 작업 수를 정의하기 위해 배열 크기(2~10,000)를 지정합니다. 배열 크기가 1,000인 작업을 제출하면 단일 작업이 실행되어 1,000개의 하위 작업을 생성합니다. 배열 작업은 모든 하위 작업을 관리하기 위한 상위 작업에 대한 참조 또는 포인터입니다. 이 기능을 사용하면 단일 쿼리로

큰 워크로드를 제출할 수 있습니다. 이 설정을 위해서는 2개의 Docker 이미지를 구축해야 합니다. 하나는 Amazon SageMaker 훈련용이고 다른 하나는 Amazon SageMaker API를 사용하여 여러 리전에서 훈련을 오케스트레이션하는 용도로 사용됩니다. AWS Batch에서 실행하는 오케스트레이터 이미지에는 서로 다른 AWS 리전에서 파라미터가 다른 여러 하위 작업을 생성하는 로직이 있지만, 4개 리전 모두에서 동일한 작업 구성을 사용합니다.

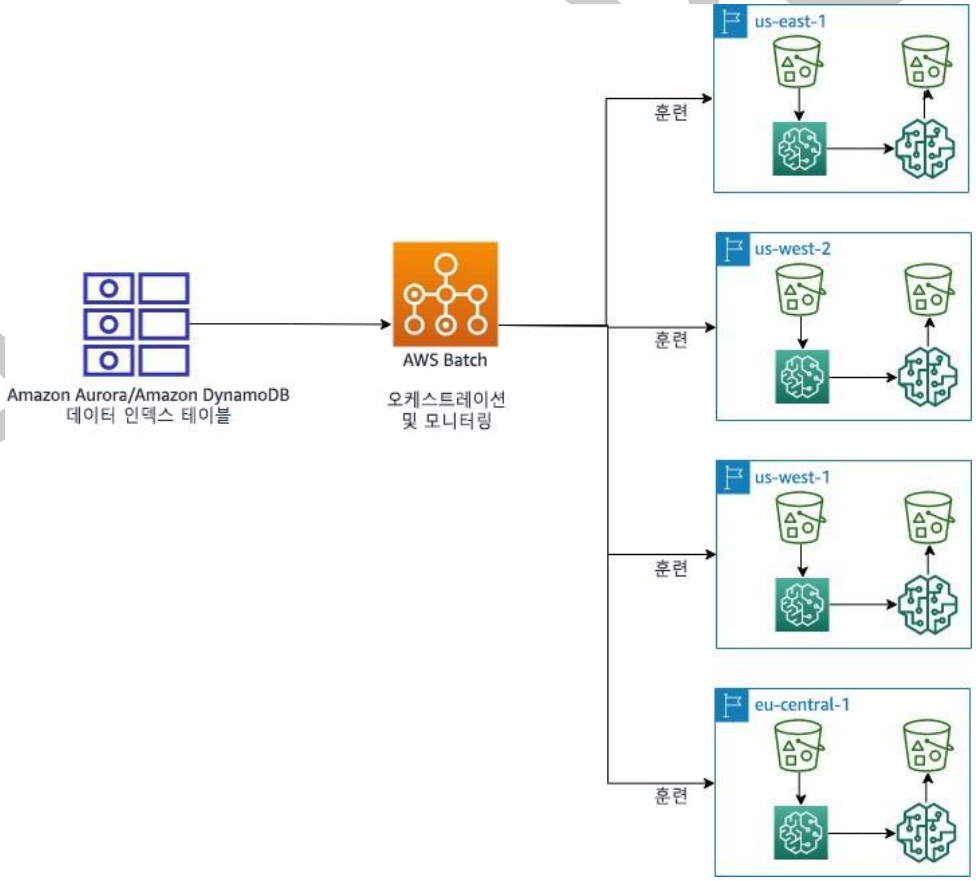


그림 14: 여러 리전에서 Amazon SageMaker 작업을 오케스트레이션하기 위한 참조 아키텍처

## Amazon S3 및 Amazon DynamoDB를 사용하여 배치 및 실시간 추론 및 훈련용 특성 저장소(Feature Store) 구축

데이터 중심 기업이 되고자 하거나 이미 데이터 중심 기업으로 변신한 조직은 데이터 레이크 솔루션을 구축하는 중이거나 분석 및 인공 지능/기계 학습을 위해 데이터를 대중화할 수 있는 데이터 레이크 솔루션을 이미 보유하고 있을 수 있습니다.

전체 조직의 데이터가 단일 리포지토리에서 관리 및 공유되므로 데이터 레이크 생성은 기계 학습 프로세스에서 중요한 단계입니다. 하지만 데이터 엔지니어가 아닌 딥 러닝 엔지니어와 과학자가 새로운 문제를 해결할 수 있는 새로운 특성을 쉽게 확보할 방법이 없다는 것이 문제입니다. 딥 러닝 엔지니어와 과학자는 데이터 레이크에 있는 엄청난 양의 데이터에서 의미 있는 특징을 어떻게 추출해야 할까요? 딥 러닝에 사용할 데이터 레이크에서 특성의 데이터 세트를 구축하는 데는 시간과 다양한 기술이 필요합니다.

특성은 관측 중인 현상의 측정 가능한 속성입니다. 특성으로는 원시 단어, 픽셀, 센서 값, 데이터 스토어의 행, CSV 파일의 필드, 집계(최소, 최대, 합계, 평균), 파생 표현(임베딩 또는 클러스터) 등이 있습니다.

다음 다이어그램에는 피처 파이프라인이 나와 있습니다. 복잡한 파이프라인을 사용하여 특성 세트를 구축하는 데 필요한 작업량이 얼마나 될지 생각해 보십시오. 고객과의 대화에서 유추된 사례 증거로 볼 때, 특성 추출은 딥 러닝 프로젝트에 소비하는 시간의 25% 이상을 차지할 수 있습니다.

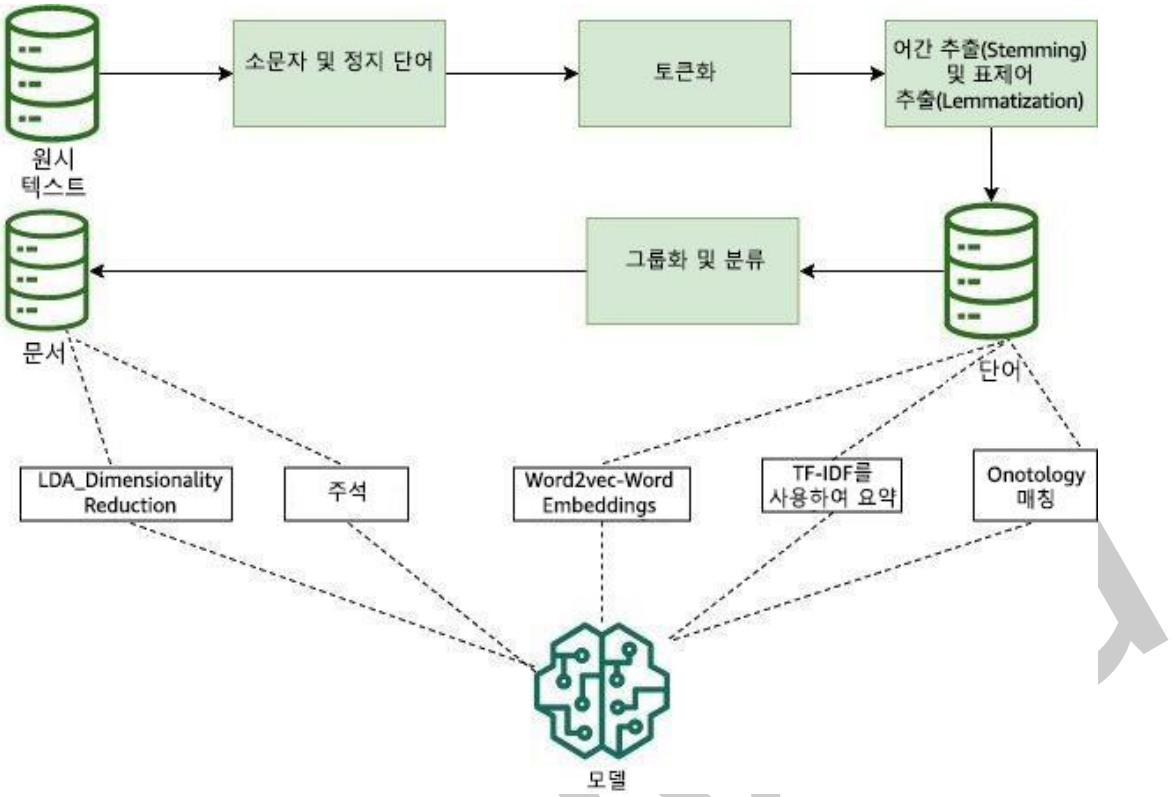


그림 15: 예제 특성 파이프라인

대부분의 기술 솔루션과 마찬가지로 이 문제를 해결할 수 있는 두 가지 방법이 있습니다. 즉, 특성 추출을 자동화하거나 기존 특성 및 모델을 재사용하는 기술을 개발하면 됩니다. 특성 추출 자동화에 대한 연구가 진행되고 있지만, 일부 조직에서는 특성 재사용에 대한 관심과 패턴이 등장하기 시작했습니다. 특성 저장소의 개념은 주목과 관심을 끌고 있습니다. 아직 초기 단계이며, 이 새로운 아이디어를 목표로 하는 솔루션을 보유한 타사 ISV 공급업체와 오픈 소스 이니셔티브가 거의 없습니다.

간단히 말해 특성 저장소는 딥 러닝 모델을 훈련, 평가 및 추론하기 위해 여러 프로젝트에서 재사용할 수 있는 딥 러닝 특성의 큐레이션된 리포지토리입니다.

딥 러닝 특성 저장소와 다른 스토어(예: 데이터 웨어하우스)의 근본적인 차이점은 특성이 딥 러닝 특성 저장소의 기본 엔터티라는 것입니다.

AWS 서비스를 사용하여 간단한 특성 저장소를 개발한 후 지속적으로 개발 및 개선 과정을 반복할 수 있습니다. 특성 저장소를 지원하려면 다음과 같은 기능이 필요합니다.

- **재사용:** 데이터 엔지니어가 개발한 기존 특성 저장소 파이프라인을 사용하여 특성 저장소의 특성을 다시 계산하고 캐싱합니다.
- **저장:** 특성 저장소에 설명, 설명서, 특성의 통계 측정치 등 특성의 메타데이터를 저장합니다.
- **검색:** 기계 학습 실무자가 API를 통해 메타데이터를 검색할 수 있도록 지원합니다.
- **거버넌스:** 거버넌스 및 액세스 제어를 위해 특성 저장소 위에 데이터 관리 계층을 추가합니다.
- **소비:** 기계 학습 실무자가 API를 사용하여 특성을 쿼리하고 소비하여 훈련 또는 실시간 추론을 위해 특성을 내보낼 수 있도록 허용합니다.

특성 저장소는 기계 학습 프로젝트 비용을 상당히 낮출 수 있습니다. 후속 프로젝트에서 큐레이트된 특성 세트를 재사용할 경우 직접적인 비용 절감 효과가 나타납니다. 비용을 절감하는 것 외에도, 특성 저장소를 재사용하면 기계 학습 실무자가 데이터 레이크에 저장된 대용량 데이터를 처리하기 보다는 큐레이트된 특성 목록을 처리하면 되기 때문에 서비스 제공에 소요되는 기간을 단축할 수 있습니다. 특성 저장소는 데이터 엔지니어로서의 역할과 기계 학습 실무 담당자 간의 추상화를 제공합니다. 또한 특성 저장소를 사용하면 훈련 및 추론의 일관성을 보장할 수 있습니다. 다음 그림은 기계 학습 프로젝트 비용과 특성 재사용 간의 관계 그래프를 보여줍니다. 그림과 같이 특성 저장소에서 더 많은 특성이 재사용됨에 따라 프로젝트 비용이 떨어집니다.

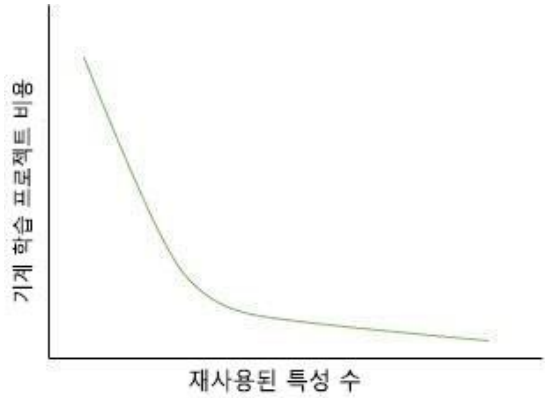


그림 16: 기계 학습 프로젝트 비용과 특성 재사용 비용

Amazon S3를 특성 저장소의 신뢰할 수 있는 소스로 사용할 수 있습니다. Amazon S3에서 제공하는 제어 기능을 사용하여 특성 저장소의 버전을 지정하고 관리할 수 있습니다. Amazon S3 를 Amazon SageMaker 훈련과 Amazon SageMaker Batch Transform 작업을 위한 스토리지 계층으로 사용할 수 있습니다. 짧은 지연 시간의 온라인 추론을 위해 특성 데이터 세트를 Amazon DynamoDB로 복사하고 추론에 사용할 수 있습니다. 특성 검색을 지원하려면 특성 메타데이터를 Amazon DynamoDB로 푸시하고 Amazon Elastic Search를 사용하여 메타데이터 검색을 지원할 수 있습니다.

다음 그림은 특성 저장소의 참조 아키텍처를 보여 줍니다.

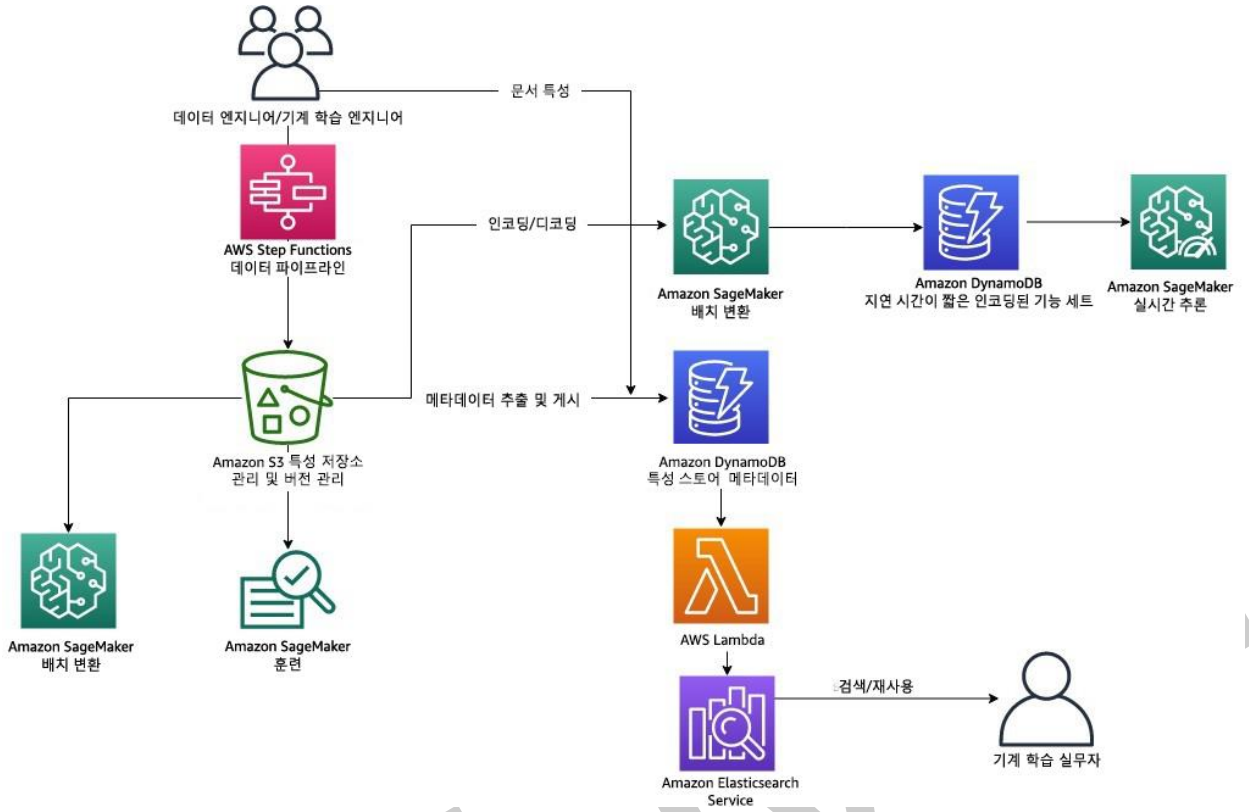


그림 17: AWS의 특성 저장소 참조 아키텍처

다음 그림은 데이터 레이크, 데이터 웨어하우스 및 특성 저장소가 엔터프라이즈 수준에서 서로 상대적으로 배치되는 방식을 보여줍니다.

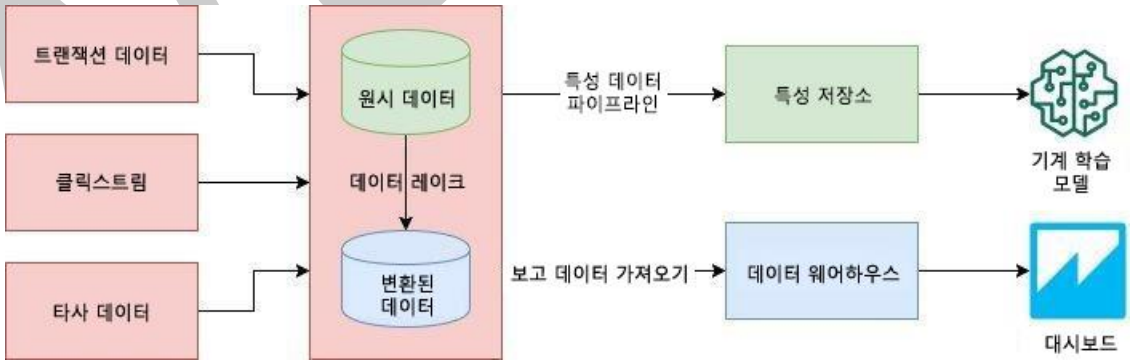


그림 18: 데이터 레이크, 데이터 웨어하우스 및 특성 저장소의 개념

# AWS 지침

이 안내서에서는 딥 러닝 스택과 딥 러닝 프로세스에 대해 설명했습니다. 딥 러닝 엔지니어와 과학자의 심층적이고 광범위한 요구 사항을 해결하는 데 사용할 수 있는 AWS 서비스에 대해 알아보았습니다. 또한 딥 러닝 엔지니어와 과학자가 조직 내에서 딥 러닝을 도입하고 확장하는 데 활용할 수 있는 설계 패턴도 설명했습니다. 기계 학습을 위한 완전관리형 서비스인 Amazon SageMaker에서 바로 사용할 수 있는 기능에 대해 설명했습니다. 또한 Amazon EC2 및 Amazon EKS와 같은 다른 AWS 서비스를 사용하여 자체적으로 딥 러닝 환경을 구축하려는 사례도 살펴보았습니다. AWS를 사용하면 자신에게 가장 적합한 접근 방식을 선택할 수 있는 유연성을 확보할 수 있습니다.

다음은 고객이 AWS 인공 지능/기계 학습 스택에서 제공하는 다양한 옵션을 활용하는 인기 있는 시나리오입니다.

스타트업이라면 고성능 GPU 컴퓨팅을 구매하고 인프라를 관리하는 데 드는 비용을 줄이려고 합니다. 개발자와 데이터 사이언티스트로 구성된 소규모 팀이 있으며 소규모 팀과 함께 새로운 딥 러닝 기능을 롤아웃하는 데 중점을 둡니다. 이 경우는 Amazon SageMaker가 이상적인 선택입니다.

조직의 연구 과학자로 일하면서 딥 러닝 기능을 사용하여 새로운 제품 기능을 개발하는 경우 더 많은 자율성, 더 많은 격리 및 더 많은 제어가 필요할 수 있습니다. 딥 러닝을 위한 표준 플랫폼을 사용하는 지침이나 정책이 조직에 없을 수 있습니다. 딥 러닝 데스크톱으로 AWS DL AMI와 함께 Amazon EC2를 계속 사용할 수 있지만, 자동 하이퍼파라미터 튜닝을 지원하는 Amazon SageMaker를 훈련에 사용하여 실험을 확장할 수도 있습니다.

변화하는 고객 선호도에 따라 딥 러닝 모델의 성능을 유지하기 위해 노력하는 제품 팀의 경우, AWS Step Functions 및 Amazon SageMaker를 사용하여 모델 재훈련 및 배포 과정을 자동화할 수 있는 포괄적인 자동 딥 러닝 파이프라인을 구현할 수 있습니다.

조직의 딥 러닝 도입을 가속화해야 하는 임무를 받은 조직의 기술 리더라면, Amazon SageMaker를 조직에 딥 러닝 모델을 구축, 훈련 및 배포하기 위한 완전관리형 서비스로 사용할 수 있습니다. Amazon SageMaker를 사용하면 소규모 딥 러닝 팀으로 더 많은 성과를 달성할 수 있습니다. 완전관리형 서비스는 운영을 간소화하고, 컴퓨팅 인프라 낭비를 없애고, 딥 러닝 엔지니어와 과학자의 생산성을 개선하고, 비용 효율적인 실험을 가능하게 하며, 출시 기간을 단축하는 데 도움이 됩니다.

Kubernetes를 통해 표준화된 인프라를 제공하기로 결정한 조직에 속한 경우 Amazon EKS를 인프라 계층으로 사용할 수 있습니다. Kubeflow와 같은 오픈 소스 구성 요소를 추가하여 Amazon EKS에서 딥 러닝 모델의 구축, 훈련 및 배포를 간소화할 수 있습니다. Amazon EKS는 영구 클러스터입니다. 클러스터는 스케일 인 및 스케일 아웃할 수 있습니다. 하지만 딥 러닝 작업을 위해 효율적이고 성능을 발휘하기 위해 튜닝이 필요한 범용 컴퓨팅 플랫폼입니다. 이를 관리하고 운영하려면 고급 전문 지식과 기술이 필요합니다. 선택적으로, Kubernetes 및 Kubeflow 스택과 더불어 데이터 레이블링 및 주석에 Amazon SageMaker Ground Truth를 사용하고 모델 최적화에 Amazon SageMaker Neo를 사용할 수 있습니다.



# 결론

이 안내서에서는 AWS에서 딥 러닝을 배포할 때 사용되는 모든 기술 빌딩 블록, 솔루션 및 패턴에 대한 포괄적인 개요 정보를 제공합니다. 또한 고객이 관심을 가질 만한 몇 가지 고급 사용 사례도 설명했습니다. AWS는 딥 러닝 연구가 빠르게 발전하고 있으며, 딥 러닝 분야에 사용되는 도구가 끊임없이 변화 및 진화하고 있음을 잘 알고 있습니다.

AWS 환경에서는 완전관리형부터 DIY에 이르기까지 대부분 딥 러닝 프로젝트의 공통적인 요구 사항과 탐색 요구 사항을 모두 아우르는 최신 옵션을 제공합니다. 그러나 사용 사례 또는 요구 사항에 따라, 프로젝트의 특별한 요구 사항을 해결하기 위해 보다 심층적인 참여와 논의가 요구될 수도 있습니다. 이러한 특별한 요구 사항이나 이 안내서에 게시된 콘텐츠에 대한 자세한 내용은 담당 AWS 계정 팀에 문의하시기 바랍니다.

# 기고자

이 문서를 작성하는 데 도움을 주신 분들입니다.

- Vikrant Kahlir, Strategic Accounts 솔루션스 아키텍트
- Christian Williams, AWS Solutions Architecture 기계 학습 전문가
- Amr Ragab, AWS Professional Services HPC 글로벌 컨설턴트

# 추가 자료

다음에서 추가 정보를 참조하십시오.

- AWS 백서 및 안내서 페이지
- AWS의 기계 학습
- AWS Machine Learning 블로그
- AWS Machine Learning 훈련

# 참고

1 Yuji Roh, Geon Heo, Steven Euijong Whang, “A Survey on Data Collection for Machine Learning: a Big Data - AI Integration Perspective”(2018년 11월): 3, https://arxiv.org/pdf/1811.03402.pdf

2 요금 정보는 Amazon SageMaker 요금 페이지에서 모델 배포를 참조하십시오.

3 https://aws.amazon.com/blogs/opensource/kubeflow-amazon-eks/

4 http://wiki.lustre.org/Introduction_to_Lustre