aws / amazon-sagemaker-examples

Code Issues 638 Pull requests 190 Discussions Actions Projects Security

Example 📓 Jupyter notebooks that demonstrate how to build, train, and deploy machine learning models using 🧠 Amazon SageMaker.

sagemaker-examples.readthedocs.io

Apache-2.0 license

Code of conduct

Security policy

8.7k stars 6.3k forks 265 watching Activity

Public repository

main

Branches Tags

rdamazon Revert "prune horovod and some repetitive notebooks (#4307)" (#4350) … ✕ 2 hours ago 2,983

View code

Amazon SageMaker

Amazon SageMaker Examples

Example Jupyter notebooks that demonstrate how to build, train, and deploy machine learning models using Amazon SageMaker.

📚 Background

Amazon SageMaker is a fully managed service for data science and machine learning (ML) workflows. You can use Amazon SageMaker to simplify the process of building, training, and deploying ML models.

The SageMaker example notebooks are Jupyter notebooks that demonstrate the usage of Amazon SageMaker.

🛠 Setup

The quickest setup to run example notebooks includes:

- An AWS account
- Proper IAM User and Role setup
- An Amazon SageMaker Notebook Instance
- An S3 bucket

💻 Usage

These example notebooks are automatically loaded into SageMaker Notebook Instances. They can be accessed by clicking on the `SageMaker Examples` tab in Jupyter or the SageMaker logo in JupyterLab.

Although most examples utilize key Amazon SageMaker functionality like distributed, managed training or real-time hosted endpoints, these notebooks can be run outside of Amazon SageMaker Notebook Instances with minimal modification (updating IAM role definition and installing the necessary libraries).

As of February 7, 2022, the default branch is named "main". See our announcement for details and how to update your existing clone.

📓 Examples

Introduction to geospatial capabilities

These examples introduce SageMaker geospatial capabilities which makes it easy to build, train, and deploy ML models using geospatial data.

- Monitoring Lake Drought with SageMaker Geospatial Capabilities shows how to monitor Lake Mead drought using SageMaker geospatial capabilities.
- Digital Farming with Amazon SageMaker Geospatial Capabilities shows how geospatial capabilities can help accelerating, optimizing, and easing the processing of the geospatial data for the Digital Farming use cases.
- Assess wildfire damage with Amazon SageMaker Geospatial Capabilities demonstrates how Amazon SageMaker geospatial capabilities can be used to identify and assess vegetation loss caused by the Dixie wildfire in Northern California.
- How to use Vector Enrichment Jobs for Map Matching shows how to use vector enrichtment operations with Amazon SageMaker Geospatial capabilities to snap GPS coordinates to road segments.
- How to use Vector Enrichment Jobs for Reverse Geocoding shows how to use Amazon SageMaker Geospatial capabilities for reverse geocoding to obtain human readable addresses from data with latitude/longitude information.

- Monitoring Glacier Melting with SageMaker Geospatial Capabilities shows how to monitor glacier melting at Mount Shasta using SageMaker geospatial capabilities.
- SageMaker Pipelines with Amazon SageMaker Geospatial Capabilities shows how a geospatial data processing workflow can be automated by using an Amazon SageMaker Pipeline.

Introduction to Ground Truth Labeling Jobs

These examples provide quick walkthroughs to get you up and running with the labeling job workflow for Amazon SageMaker Ground Truth.

- Bring your own model for SageMaker labeling workflows with active learning is an end-to-end example that shows how to bring your custom training, inference logic and active learning to the Amazon SageMaker ecosystem.
- From Unlabeled Data to a Deployed Machine Learning Model: A SageMaker Ground Truth Demonstration for Image Classification is an end-to-end example that starts with an unlabeled dataset, labels it using the Ground Truth API, analyzes the results, trains an image classification neural net using the annotated dataset, and finally uses the trained model to perform batch and online inference.
- Ground Truth Object Detection Tutorial is a similar end-to-end example but for an object detection task.
- Basic Data Analysis of an Image Classification Output Manifest presents charts to visualize the number of annotations for each class, differentiating between human annotations and automatic labels (if your job used auto-labeling). It also displays sample images in each class, and creates a pdf which concisely displays the full results.
- Training a Machine Learning Model Using an Output Manifest introduces the concept of an "augmented manifest" and demonstrates that the output file of a labeling job can be immediately used as the input file to train a SageMaker machine learning model.
- Annotation Consolidation demonstrates Amazon SageMaker Ground Truth annotation consolidation techniques for image classification for a completed labeling job.

Introduction to Applying Machine Learning

These examples provide a gentle introduction to machine learning concepts as they are applied in practical use cases across a variety of sectors.

- Predicting Customer Churn uses customer interaction and service usage data to find those most likely to churn, and then walks through the cost/benefit trade-offs of providing retention incentives. This uses Amazon SageMaker's implementation of XGBoost to create a highly predictive model.
- Cancer Prediction predicts Breast Cancer based on features derived from images, using SageMaker's Linear Learner.
- Ensembling predicts income using two Amazon SageMaker models to show the advantages in ensembling.
- Video Game Sales develops a binary prediction model for the success of video games based on review scores.
- MXNet Gluon Recommender System uses neural network embeddings for non-linear matrix factorization to predict user movie ratings on Amazon digital reviews.

- Fair Linear Learner is an example of an effective way to create fair linear models with respect to sensitive features.
- Population Segmentation of US Census Data using PCA and Kmeans analyzes US census data and reduces dimensionality using PCA then clusters US counties using KMeans to identify segments of similar counties.
- Document Embedding using Object2Vec is an example to embed a large collection of documents in a common low-dimensional space, so that the semantic distances between these documents are preserved.
- Traffic violations forecasting using DeepAR is an example to use daily traffic violation data to predict pattern and seasonality to use Amazon DeepAR alogorithm.
- Visual Inspection Automation with Pre-trained Amazon SageMaker Models is an example for fine-tuning pre-trained Amazon Sagemaker models on a target dataset.
- Create SageMaker Models Using the PyTorch Model Zoo contains an example notebook to create a SageMaker model leveraging the PyTorch Model Zoo and visualize the results.
- Deep Demand Forecasting provides an end-to-end solution for Demand Forecasting task using three state-of-the-art time series algorithms LSTNet, Prophet, and SageMaker DeepAR, which are available in GluonTS and Amazon SageMaker.
- Fraud Detection Using Graph Neural Networks is an example to identify fraudulent transactions from transaction and user identity datasets.
- Identify key insights from textual document contains comphrensive notebooks for five natural language processing tasks Document Summarization, Text Classification, Question and Answering, Name Entity Recognition, and Semantic Relation Extracion.
- Synthetic Churn Prediction with Text contains an example notebook to train, deploy and use a churn prediction model that processed numerical, categorical and textual features to make its prediction.
- Credit Card Fraud Detector is an example of the core of a credit card fraud detection system using SageMaker with Random Cut Forest and XGBoost.
- Churn Prediction Multimodality of Text and Tabular is an example notebook to train and deploy a churn prediction model that uses state-of-the-art natural language processing model to find useful signals in text. In addition to textual inputs, this model uses traditional structured data inputs such as numerical and categorical fields.

SageMaker Automatic Model Tuning

These examples introduce SageMaker's hyperparameter tuning functionality which helps deliver the best possible predictions by running a large number of training jobs to determine which hyperparameter values are the most impactful.

- XGBoost Tuning shows how to use SageMaker hyperparameter tuning to improve your model fit.
- BlazingText Tuning shows how to use SageMaker hyperparameter tuning with the BlazingText built-in algorithm and 20_newsgroups dataset..
- TensorFlow Tuning shows how to use SageMaker hyperparameter tuning with the pre-built TensorFlow container and MNIST dataset.
- MXNet Tuning shows how to use SageMaker hyperparameter tuning with the pre-built MXNet container and MNIST dataset.

- HuggingFace Tuning shows how to use SageMaker hyperparameter tuning with the pre-built HuggingFace container and 20_newsgroups dataset.
- Keras BYO Tuning shows how to use SageMaker hyperparameter tuning with a custom container running a Keras convolutional network on CIFAR-10 data.
- R BYO Tuning shows how to use SageMaker hyperparameter tuning with the custom container from the Bring Your Own R Algorithm example.
- Analyzing Results is a shared notebook that can be used after each of the above notebooks to provide analysis on how training jobs with different hyperparameters performed.
- Model tuning for distributed training shows how to use SageMaker hyperparameter tuning with Hyperband strategy for optimizing model in distributed training.

SageMaker Autopilot

These examples introduce SageMaker Autopilot. Autopilot automatically performs feature engineering, model selection, model tuning (hyperparameter optimization) and allows you to directly deploy the best model to an endpoint to serve inference requests.

- Customer Churn AutoML shows how to use SageMaker Autopilot to automatically train a model for the Predicting Customer Churn task.
- Targeted Direct Marketing AutoML shows how to use SageMaker Autopilot to automatically train a model.
- Housing Prices AutoML shows how to use SageMaker Autopilot for a linear regression problem (predict housing prices).
- Portfolio Churn Prediction with Amazon SageMaker Autopilot and Neo4j shows how to use SageMaker Autopilot with graph embeddings to predict investment portfolio churn.
- Move Amazon SageMaker Autopilot ML models from experimentation to production using Amazon SageMaker Pipelines shows how to use SageMaker Autopilot in combination with SageMaker Pipelines for end-to-end AutoML training automation.
- Amazon SageMaker Autopilot models to serverless endpoints shows how to deploy Autopilot generated models to serverless endpoints.

Introduction to Amazon Algorithms

These examples provide quick walkthroughs to get you up and running with Amazon SageMaker's custom developed algorithms. Most of these algorithms can train on distributed hardware, scale incredibly well, and are faster and cheaper than popular alternatives.

- k-means is our introductory example for Amazon SageMaker. It walks through the process of clustering MNIST images of handwritten digits using Amazon SageMaker k-means.
- Factorization Machines showcases Amazon SageMaker's implementation of the algorithm to predict whether a handwritten digit from the MNIST dataset is a 0 or not using a binary classifier.
- Latent Dirichlet Allocation (LDA) introduces topic modeling using Amazon SageMaker Latent Dirichlet Allocation (LDA) on a synthetic dataset.
- Linear Learner predicts whether a handwritten digit from the MNIST dataset is a 0 or not using a binary classifier from Amazon SageMaker Linear Learner.

- Neural Topic Model (NTM) uses Amazon SageMaker Neural Topic Model (NTM) to uncover topics in documents from a synthetic data source, where topic distributions are known.
- Principal Components Analysis (PCA) uses Amazon SageMaker PCA to calculate eigendigits from MNIST.
- Seq2Seq uses the Amazon SageMaker Seq2Seq algorithm that's built on top of Sockeye, which is a sequence-to-sequence framework for Neural Machine Translation based on MXNet. Seq2Seq implements state-of-the-art encoder-decoder architectures which can also be used for tasks like Abstractive Summarization in addition to Machine Translation. This notebook shows translation from English to German text.
- Image Classification includes full training and transfer learning examples of Amazon SageMaker's Image Classification algorithm. This uses a ResNet deep convolutional neural network to classify images from the caltech dataset.
- XGBoost for regression predicts the age of abalone (Abalone dataset) using regression from Amazon SageMaker's implementation of XGBoost.
- XGBoost for multi-class classification uses Amazon SageMaker's implementation of XGBoost to classify handwritten digits from the MNIST dataset as one of the ten digits using a multi-class classifier. Both single machine and distributed use-cases are presented.
- DeepAR for time series forecasting illustrates how to use the Amazon SageMaker DeepAR algorithm for time series forecasting on a synthetically generated data set.
- BlazingText Word2Vec generates Word2Vec embeddings from a cleaned text dump of Wikipedia articles using SageMaker's fast and scalable BlazingText implementation.
- Object detection for bird images demonstrates how to use the Amazon SageMaker Object Detection algorithm with a public dataset of Bird images.
- Object2Vec for movie recommendation demonstrates how Object2Vec can be used to model data consisting of pairs of singleton tokens using movie recommendation as a running example.
- Object2Vec for multi-label classification shows how ObjectToVec algorithm can train on data consisting of pairs of sequences and singleton tokens using the setting of genre prediction of movies based on their plot descriptions.
- Object2Vec for sentence similarity explains how to train Object2Vec using sequence pairs as input using sentence similarity analysis as the application.
- IP Insights for suspicious logins shows how to train IP Insights on a login events for a web server to identify suspicious login attempts.
- Semantic Segmentation shows how to train a semantic segmentation algorithm using the Amazon SageMaker Semantic Segmentation algorithm. It also demonstrates how to host the model and produce segmentation masks and probability of segmentation.
- JumpStart Instance Segmentation demonstrates how to use a pre-trained Instance Segmentation model available in JumpStart for inference.
- JumpStart Semantic Segmentation demonstrates how to use a pre-trained Semantic Segmentation model available in JumpStart for inference, how to finetune the pre-trained model on a custom dataset using JumpStart transfer learning algorithm, and how to use fine-tuned model for inference.
- JumpStart Text Generation shows how to use JumpStart to generate text that appears indistinguishable from the hand-written text.
- JumpStart Text Summarization shows how to use JumpStart to summarize the text to contain only the important information.

- JumpStart Image Embedding demonstrates how to use a pre-trained model available in JumpStart for image embedding.
- JumpStart Text Embedding demonstrates how to use a pre-trained model available in JumpStart for text embedding.
- JumpStart Object Detection demonstrates how to use a pre-trained Object Detection model available in JumpStart for inference, how to finetune the pre-trained model on a custom dataset using JumpStart transfer learning algorithm, and how to use fine-tuned model for inference.
- JumpStart Machine Translation demonstrates how to translate text from one language to another language in JumpStart.
- JumpStart Named Entity Recognition demonstrates how to identify named entities such as names, locations etc. in the text in JumpStart.
- JumpStart Text to Image demonstrates how to generate image conditioned on text in JumpStart.
- JumpStart Upscaling demonstrates how to enhance image quality with Stable Diffusion models in JumpStart.
- JumpStart Inpainting demonstrates how to inpaint an image with Stable Diffusion models in JumpStart.
- In-context learning with AlexaTM 20B demonstrates how to use AlexaTM 20B for in-context-learning in JumpStart.

Amazon SageMaker RL

The following provide examples demonstrating different capabilities of Amazon SageMaker RL.

- Cartpole using Coach demonstrates the simplest usecase of Amazon SageMaker RL using Intel's RL Coach.
- AWS DeepRacer demonstrates AWS DeepRacer trainig using RL Coach in the Gazebo environment.
- HVAC using EnergyPlus demonstrates the training of HVAC systems using the EnergyPlus environment.
- Knapsack Problem demonstrates how to solve the knapsack problem using a custom environment.
- Mountain Car Mountain car is a classic RL problem. This notebook explains how to solve this using the OpenAI Gym environment.
- Distributed Neural Network Compression This notebook explains how to compress ResNets using RL, using a custom environment and the RLLib toolkit.
- Portfolio Management This notebook uses a custom Gym environment to manage multiple financial investments.
- Autoscaling demonstrates how to adjust load depending on demand. This uses RL Coach and a custom environment.
- Roboschool is an open source physics simulator that is commonly used to train RL policies for robotic systems. This notebook demonstrates training a few agents using it.
- Stable Baselines In this notebook example, we will make the HalfCheetah agent learn to walk using the stable-baselines, which are a set of improved implementations of Reinforcement Learning (RL) algorithms based on OpenAI Baselines.
- Travelling Salesman is a classic NP hard problem, which this notebook solves with AWS SageMaker RL.

- Tic-tac-toe is a simple implementation of a custom Gym environment to train and deploy an RL agent in Coach that then plays tic-tac-toe interactively in a Jupyter Notebook.
- Unity Game Agent shows how to use RL algorithms to train an agent to play Unity3D game.

Scientific Details of Algorithms

These examples provide more thorough mathematical treatment on a select group of algorithms.

- Streaming Median sequentially introduces concepts used in streaming algorithms, which many SageMaker algorithms rely on to deliver speed and scalability.
- Latent Dirichlet Allocation (LDA) dives into Amazon SageMaker's spectral decomposition approach to LDA.
- Linear Learner features shows how to use the class weights and loss functions features of the SageMaker Linear Learner algorithm to improve performance on a credit card fraud prediction task

Amazon SageMaker Debugger

These examples provide and introduction to SageMaker Debugger which allows debugging and monitoring capabilities for training of machine learning and deep learning algorithms. Note that although these notebooks focus on a specific framework, the same approach works with all the frameworks that Amazon SageMaker Debugger supports. The notebooks below are listed in the order in which we recommend you review them.

- Using a built-in rule with TensorFlow
- Using a custom rule with TensorFlow Keras
- Interactive tensor analysis in notebook with MXNet
- Visualizing Debugging Tensors of MXNet training
- Real-time analysis in notebook with MXNet
- Using a built in rule with XGBoost
- Real-time analysis in notebook with XGBoost
- Using SageMaker Debugger with Managed Spot Training and MXNet
- Reacting to CloudWatch Events from Rules to take an action based on status with TensorFlow
- Using SageMaker Debugger with a custom PyTorch container

Amazon SageMaker Distributed Training

These examples provide an introduction to SageMaker Distributed Training Libraries for data parallelism and model parallelism. The libraries are optimized for the SageMaker training environment, help adapt your distributed training jobs to SageMaker, and improve training speed and throughput. More examples for models such as BERT and YOLOv5 can be found in distributed_training/.

- Train GPT-2 with Sharded Data Parallel shows how to train GPT-2 with near-linear scaling using Sharded Data Parallelism technique in SageMaker Model Parallelism Library.
- Train EleutherAI GPT-J with Model Parallel shows how to train EleutherAI GPT-J with PyTorch and Tensor Parallelism technique in the SageMaker Model Parallelism Library.

- Train MaskRCNN with Data Parallel shows how to train MaskRCNN with PyTorch and SageMaker Data Parallelism Library.

Amazon SageMaker Clarify

These examples provide an introduction to SageMaker Clarify which provides machine learning developers with greater visibility into their training data and models so they can identify and limit bias and explain predictions.

- Fairness and Explainability with SageMaker Clarify shows how to use SageMaker Clarify Processor API to measure the pre-training bias of a dataset and post-training bias of a model, and explain the importance of the input features on the model's decision.
- Amazon SageMaker Clarify Model Monitors shows how to use SageMaker Clarify Model Monitor API to schedule bias monitor to monitor predictions for bias drift on a regular basis, and schedule explainability monitor to monitor predictions for feature attribution drift on a regular basis.

Publishing content from RStudio on Amazon SageMaker to RStudio Connect

These examples show you how to run R examples, and publish applications in RStudio on Amazon SageMaker to RStudio Connect.

- Publishing R Markdown shows how you can author an R Markdown document (.Rmd, .Rpres) within RStudio on Amazon SageMaker and publish to RStudio Connect for wide consumption.
- Publishing R Shiny Apps shows how you can author an R Shiny application within RStudio on Amazon SageMaker and publish to RStudio Connect for wide consumption.
- Publishing Streamlit Apps shows how you can author a streamlit application withing Amazon SageMaker Studio and publish to RStudio Connect for wide consumption.

Advanced Amazon SageMaker Functionality

These examples showcase unique functionality available in Amazon SageMaker. They cover a broad range of topics and utilize a variety of methods, but aim to provide the user with sufficient insight or inspiration to develop within Amazon SageMaker.

- Data Distribution Types showcases the difference between two methods for sending data from S3 to Amazon SageMaker Training instances. This has particular implication for scalability and accuracy of distributed training.
- Encrypting Your Data shows how to use Server Side KMS encrypted data with Amazon SageMaker training. The IAM role used for S3 access needs to have permissions to encrypt and decrypt data with the KMS key.
- Using Parquet Data shows how to bring Parquet data sitting in S3 into an Amazon SageMaker Notebook and convert it into the recordIO-protobuf format that many SageMaker algorithms consume.
- Connecting to Redshift demonstrates how to copy data from Redshift to S3 and vice-versa without leaving Amazon SageMaker Notebooks.
- Bring Your Own XGBoost Model shows how to use Amazon SageMaker Algorithms containers to bring a pre-trained model to a realtime hosted endpoint without ever needing to think about REST APIs.

- Bring Your Own k-means Model shows how to take a model that's been fit elsewhere and use Amazon SageMaker Algorithms containers to host it.
- Bring Your Own R Algorithm shows how to bring your own algorithm container to Amazon SageMaker using the R language.
- Installing the R Kernel shows how to install the R kernel into an Amazon SageMaker Notebook Instance.
- Bring Your Own scikit Algorithm provides a detailed walkthrough on how to package a scikit learn algorithm for training and production-ready hosting.
- Bring Your Own MXNet Model shows how to bring a model trained anywhere using MXNet into Amazon SageMaker.
- Bring Your Own TensorFlow Model shows how to bring a model trained anywhere using TensorFlow into Amazon SageMaker.
- Bring Your Own Model train and deploy BERTopic shows how to bring a model through an external library, how to train it and deploy it into Amazon SageMaker by extending the pytorch base containers.
- Experiment Management Capabilities with Search shows how to organize Training Jobs into projects, and track relationships between Models, Endpoints, and Training Jobs.
- Host Multiple Models with Your Own Algorithm shows how to deploy multiple models to a realtime hosted endpoint with your own custom algorithm.
- Host Multiple Models with XGBoost shows how to deploy multiple models to a realtime hosted endpoint using a multi-model enabled XGBoost container.
- Host Multiple Models with SKLearn shows how to deploy multiple models to a realtime hosted endpoint using a multi-model enabled SKLearn container.
- Host Multimodal HuggingFace Model shows how to host an instruction based image editing model from HuggingFace as a SageMaker endpoint using single core or multi-core GPU based instances. Inference Recommender is used to run load tests and compare the performance of instances.
- SageMaker Training and Inference with Script Mode shows how to use custom training and inference scripts, similar to those you would use outside of SageMaker, with SageMaker's prebuilt containers for various frameworks like Scikit-learn, PyTorch, and XGBoost.
- Host Models with NVidia Triton Server shows how to deploy models to a realtime hosted endpoint using Triton as the model inference server.
- Heterogenous Clusters Training in TensorFlow or PyTorch shows how to train using TensorFlow tf.data.service (distributed data pipeline) or Pytorch (with gRPC) on top of Amazon SageMaker Heterogenous clusters to overcome CPU bottlenecks by including different instance types (GPU/CPU) in the same training job.

Amazon SageMaker Neo Compilation Jobs

These examples provide an introduction to how to use Neo to compile and optimize deep learning models.

- GluonCV SSD Mobilenet shows how to train GluonCV SSD MobileNet and use Amazon SageMaker Neo to compile and optimize the trained model.
- Image Classification Adapts from image classification including Neo API and comparison against the uncompiled baseline.

- MNIST with MXNet Adapts from MXNet MNIST including Neo API and comparison against the uncompiled baseline.
- Deploying pre-trained PyTorch vision models shows how to use Amazon SageMaker Neo to compile and optimize pre-trained PyTorch models from TorchVision.
- Distributed TensorFlow includes Neo API and comparison against the uncompiled baseline.
- Predicting Customer Churn Adapts from XGBoost customer churn including Neo API and comparison against the uncompiled baseline.

Amazon SageMaker Processing

These examples show you how to use SageMaker Processing jobs to run data processing workloads.

- Scikit-Learn Data Processing and Model Evaluation shows how to use SageMaker Processing and the Scikit-Learn container to run data preprocessing and model evaluation workloads.
- Feature transformation with Amazon SageMaker Processing and SparkML shows how to use SageMaker Processing to run data processing workloads using SparkML prior to training.
- Feature transformation with Amazon SageMaker Processing and Dask shows how to use SageMaker

README.md

built-in Spark container on SageMaker Processing using the SageMaker Python SDK.

Amazon SageMaker Pipelines

These examples show you how to use SageMaker Pipelines to create, automate and manage end-to-end Machine Learning workflows.

- Amazon Comprehend with SageMaker Pipelines shows how to deploy a custom text classification using Amazon Comprehend and SageMaker Pipelines.
- Amazon Forecast with SageMaker Pipelines shows how you can create a dataset, dataset group and predictor with Amazon Forecast and SageMaker Pipelines.
- Multi-model SageMaker Pipeline with Hyperparamater Tuning and Experiments shows how you can generate a regression model by training real estate data from Athena using Data Wrangler, and uses multiple algorithms both from a custom container and a SageMaker container in a single pipeline.
- SageMaker Pipeline Local Mode with FrameworkProcessor and BYOC for PyTorch with sagemaker-training-toolkig
- SageMaker Pipeline Step Caching shows how you can leverage pipeline step caching while building pipelines and shows expected cache hit / cache miss behavior.
- Native AutoML step in SageMaker Pipelines shows how you can use SageMaker Autopilot with a native AutoML step in SageMaker Pipelines for end-to-end AutoML training automation.

Amazon SageMaker Pre-Built Framework Containers and the Python SDK

Pre-Built Deep Learning Framework Containers

These examples show you how to train and host in pre-built deep learning framework containers using the SageMaker Python SDK.

- Chainer CIFAR-10 trains a VGG image classification network on CIFAR-10 using Chainer (both single machine and multi-machine versions are included)
- Chainer MNIST trains a basic neural network on MNIST using Chainer (shows how to use local mode)
- Chainer sentiment analysis trains a LSTM network with embeddings to predict text sentiment using Chainer
- IRIS with Scikit-learn trains a Scikit-learn classifier on IRIS data
- Model Registry and Batch Transform with Scikit-learn trains a Scikit-learn Random Forest model, registers it in Model Registry, and runs a Batch Transform Job.
- MNIST with MXNet Gluon trains a basic neural network on the MNIST handwritten digit dataset using MXNet Gluon
- MNIST with MXNet trains a basic neural network on the MNIST handwritten digit data using MXNet's symbolic syntax
- Sentiment Analysis with MXNet Gluon trains a text classifier using embeddings with MXNet Gluon
- TensorFlow training and serving trains a basic neural network on MNIST
- TensorFlow with Horovod trains on MNIST using Horovod for distributed training
- TensorFlow using shell commands shows how to use a shell script for the container's entry point

Pre-Built Machine Learning Framework Containers

These examples show you how to build Machine Learning models with frameworks like Apache Spark or Scikit-learn using SageMaker Python SDK.

- Inference with SparkML Serving shows how to build an ML model with Apache Spark using Amazon EMR on Abalone dataset and deploy in SageMaker with SageMaker SparkML Serving.
- Pipeline Inference with Scikit-learn and LinearLearner builds a ML pipeline using Scikit-learn preprocessing and LinearLearner algorithm in single endpoint

Using Amazon SageMaker with Apache Spark

These examples show how to use Amazon SageMaker for model training, hosting, and inference through Apache Spark using SageMaker Spark. SageMaker Spark allows you to interleave Spark Pipeline stages with Pipeline stages that interact with Amazon SageMaker.

- MNIST with SageMaker PySpark
- Parameterize spark configuration in pipeline PySparkProcessor execution shows how you can define spark-configuration in different pipeline PysparkProcessor executions

Using Amazon SageMaker with Amazon Keyspaces (for Apache Cassandra)

These examples show how to use Amazon SageMaker to read data from Amazon Keyspaces.

- Train Machine Learning Models using Amazon Keyspaces as a Data Source

AWS Marketplace

Create algorithms/model packages for listing in AWS Marketplace for machine learning.

These example notebooks show you how to package a model or algorithm for listing in AWS Marketplace for machine learning.

- Creating Marketplace Products
  - Creating a Model Package - Listing on AWS Marketplace provides a detailed walkthrough on how to package a pre-trained model as a SageMaker Model Package that can be listed on AWS Marketplace.
  - Creating Algorithm and Model Package - Listing on AWS Marketplace provides a detailed walkthrough on how to package a scikit learn algorithm to create SageMaker Algorithm and SageMaker Model Package entities that can be used with the enhanced SageMaker Train/Transform/Hosting/Tuning APIs and listed on AWS Marketplace.

Once you have created an algorithm or a model package to be listed in the AWS Marketplace, the next step is to list it in AWS Marketplace, and provide a sample notebook that customers can use to try your algorithm or model package.

- Curate your AWS Marketplace model package listing and sample notebook provides instructions on how to craft a sample notebook to be associated with your listing and how to curate a good AWS Marketplace listing that makes it easy for AWS customers to consume your model package.
- Curate your AWS Marketplace algorithm listing and sample notebook provides instructions on how to craft a sample notebook to be associated with your listing and how to curate a good AWS Marketplace listing that makes it easy for your customers to consume your algorithm.

Use algorithms, data, and model packages from AWS Marketplace.

These examples show you how to use model-packages and algorithms from AWS Marketplace and dataset products from AWS Data Exchange, for machine learning.

- Using Algorithms
  - Using Algorithm From AWS Marketplace provides a detailed walkthrough on how to use Algorithm with the enhanced SageMaker Train/Transform/Hosting/Tuning APIs by choosing a canonical product listed on AWS Marketplace.
  - Using AutoML algorithm provides a detailed walkthrough on how to use AutoML algorithm from AWS Marketplace.
- Using Model Packages
  - Using Model Packages From AWS Marketplace is a generic notebook which provides sample code snippets you can modify and use for performing inference on Model Packages from AWS Marketplace, using Amazon SageMaker.
  - Using Amazon Demo product From AWS Marketplace provides a detailed walkthrough on how to use Model Package entities with the enhanced SageMaker Transform/Hosting APIs by choosing a canonical product listed on AWS Marketplace.
  - Using models for extracting vehicle metadata provides a detailed walkthrough on how to use pre-trained models from AWS Marketplace for extracting metadata for a sample use-case of auto-insurance claim processing.
  - Using models for identifying non-compliance at a workplace provides a detailed walkthrough on how to use pre-trained models from AWS Marketplace for extracting metadata for a sample use-

case of generating summary reports for identifying non-compliance at a construction/industrial workplace.
  - Creative writing using GPT-2 Text Generation will show you how to use AWS Marketplace GPT-2-XL pre-trained model on Amazon SageMaker to generate text based on your prompt to help you author prose and poetry.
  - Amazon Augmented AI with AWS Marketplace ML models will show you how to use AWS Marketplace pre-trained ML models with Amazon Augmented AI to implement human-in-loop workflow reviews with your ML model predictions.
  - Monitoring data quality in third-party models from AWS Marketplace will show you how to perform Data Quality monitoring on a pre-trained third-party model from AWS Marketplace.
  - Evaluating ML models from AWS Marketplace for person counting use case will show you how to use two AWS Marketplace GluonCV pre-trained ML models for person counting use case and evaluate each model for performance in different types of crowd images.
  - Preprocessing audio data using a pre-trained machine learning model demonstrates the usage of a pre-trained audio track separation model to create synthetic features and improve an acoustic classification model.
- Using Dataset Products
  - Using Dataset Product from AWS Data Exchange with ML model from AWS Marketplace is a sample notebook which shows how a dataset from AWS Data Exchange can be used with an ML Model Package from AWS Marketplace.
  - Using Shutterstock Image Datasets to train Image Classification Models provides a detailed walkthrough on how to use the Free Sample: Images & Metadata of "Whole Foods" Shoppers from Shutterstock's Image Datasets to train a multi-label image classification model using Shutterstock's pre-labeled image assets. You can learn more about this implementation from this blog post.

⚖️ License

This library is licensed under the Apache 2.0 License. For more details, please take a look at the LICENSE file.

🤝 Contributing

Although we're extremely excited to receive contributions from the community, we're still working on the best mechanism to take in examples from external sources. Please bear with us in the short-term if pull requests take longer than expected or are closed. Please read our contributing guidelines if you'd like to open an issue or submit a pull request.

Releases 1

Examples checkpoint prior to website launch Latest
on Oct 19, 2020

Packages

No packages published

Contributors 502

+ 491 contributors

Languages

● **Jupyter Notebook** 94.2% ● **Python** 4.8% ● **Roff** 0.7% ● **Shell** 0.2% ● **Dockerfile** 0.1% ● **HTML** 0.0%