Algorithms and Frameworks

AWS SageMaker

Chandra Lingam

Cloud Wave LLC

SageMaker – Training and Hosting

Options	Usage Scenario
Built-in Algorithms	Training algorithms provided by SageMaker Easy to use and scale Optimized for AWS Cloud
Pre-built Container Images	Supports popular frameworks like MxNet, TensorFlow, scikit-learn, PyTorch Flexibility to use wide selection of algorithms
Extend Pre-built Container Images	Extend pre-built container images to your needs
Custom Container Images	Custom algorithm, language and framework

Built-in Algorithms

- Training algorithms provided by SageMaker
- Easy to use and scale
- Optimized for AWS Cloud
- GPU Support

BlazingText

Type	Purpose	Use
Unsupervised	Convert Word to vector (Word2Vec)	Word2Vec is a text preprocessing step for downstream NLP, Sentiment analysis, named entity recognition and translation. Words that are semantically similar have vectors that are closer to each other Example: All vegetable names are closer to each other in the vector space
Supervised	Multi-class, Multi-label Classification	Classification based on Text (single-label) Example: Spam Detection – Spam/Not-Spam A single instance can belong to many classes (multi-label) Example: movie can belong to multiple genre

Reference: SageMaker BlazingText and https://fasttext.cc/

Object2Vec

Type	Purpose	Use
Supervised	Classification, Regression	Extends Word2Vec Captures structure of sentences Learns relationship between pair of objects Example: similarity search based on Customer-Product, Movie-Ratings and so forth

Factorization Machines

Type	Purpose	Use
Supervised	Regression, Classification	Works very well with high dimensional sparse datasets Popular algorithm for building Recommender systems Collaborative Filtering Example: Movie Recommendation based on your viewing habits; Cross recommend based on similar users

K-Nearest Neighbors

Type	Purpose	Use
Supervised	Regression, Classification	Classification – Queries K-Nearest Neighbors and assigns majority class for the instance Regression - Queries K-Nearest Neighbors and returns average value for the instance Does not scale well for large datasets

Linear Learner

Type	Purpose	Use
Supervised	Regression, Classification	Linear models for regression, binary classification and multi-class classifcation

XGBoost

Type	Purpose	Use
Supervised	Regression, Classification	Gradient Boosted Trees Algorithm Very Popular Algorithm - Won several competitions

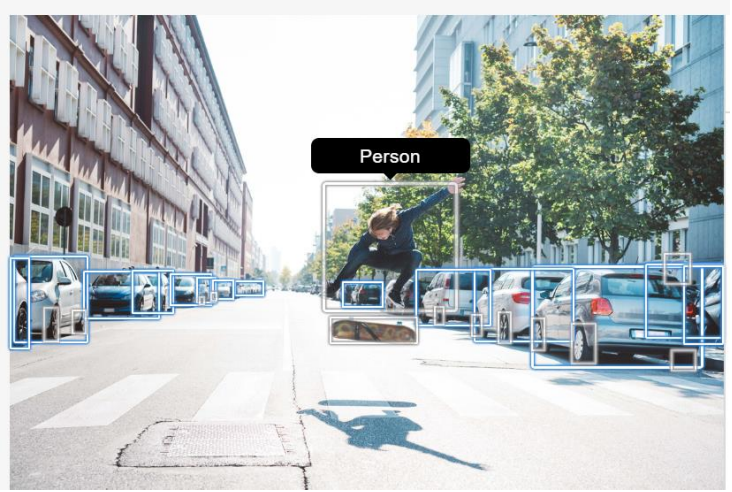
DeepAR

Type	Purpose	Use
Supervised	Timeseries Forecasting	Train multiple related time series using a single model Generate predictions for new, similar timeseries

Object Detection

Type	Purpose	Use
Supervised	Classification	Image Analysis Algorithm Detects and Classifies Objects in an Image Returns a bounding box of each object location

Object Detection



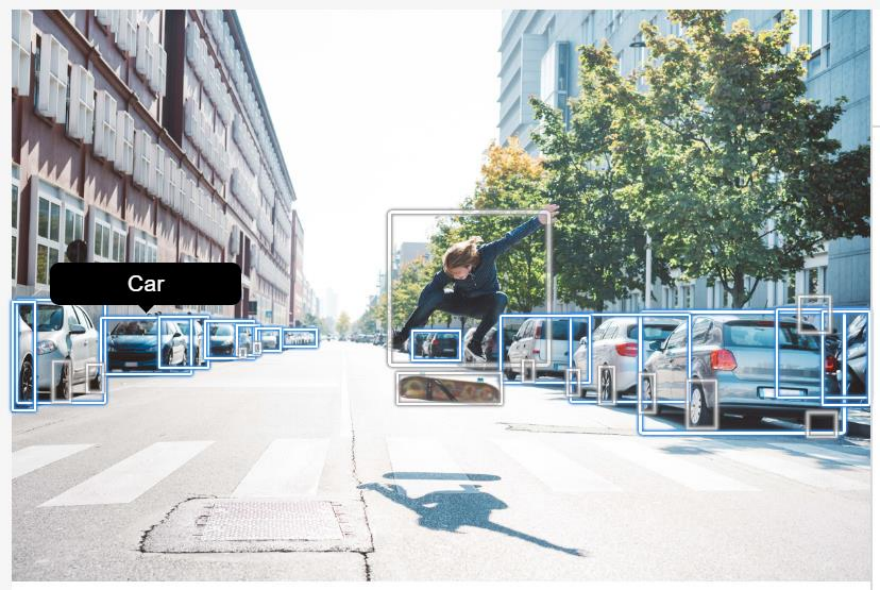


Image Courtesy: Amazon Rekognition

Image Classification

Type	Purpose	Use
Supervised	Classification	Image Analysis Algorithm Classifies entire Image Supports multi-labels

Image Classification

Bear

Butterfly

Bird

Image Courtesy: Caltech 256 Dataset
http://www.vision.caltech.edu/Image_Datasets/Caltech256/

Semantic Segmentation

Type	Purpose	Use
Supervised	Classification	Image Analysis Algorithm for Computer Vision Applications Tags each pixel in an image with a class label Example: Identify shape of car

Semantic Segmentation

Cars

Bus

People

Image Courtesy: COCO Dataset
http://cocodataset.org/#explore

Sequence to Sequence (seq2seq)

Type	Purpose	Use
Supervised	Convert sequence of tokens	Input: Sequence of tokens Output: Another sequence of tokens Examples: Text Summarization, Language Translation, Speech to Text

K-Means

Type	Purpose	Use
Unsupervised	Clustering	Identify discrete groupings within data “Members of a group are as similar as possible to one another and as different as possible from members of other groups”

Latent Dirichlet Allocation (LDA)

Type	Purpose	Use
Unsupervised	Topic Modeling	Group documents by user specified “number” of topics For documents, it assigns a probability score for each topic

Neural Topic Model (NTM)

Type	Purpose	Use
Unsupervised	Topic Modeling	Similar to LDA

Principal Component Analysis (PCA)

Type	Purpose	Use
Unsupervised	Dimensionality Reduction	"Reduces dimensionality of a dataset while retaining as much information as possible" "Returns components – a new set of features that are composites of original features and that are uncorrelated to one another" Examples: Reduce the dimensions of a dataset, visualize high dimensional datasets, remove highly correlated features

Random Cut Forest (RCF)

Type	Purpose	Use
Unsupervised	Anomaly Detection	“Anomalous points are observations that diverge from otherwise well-structured or patterned data” For each data point, RCF assigns an anomaly score Low score indicates normal data and high score indicates an anomaly

IP Insights

Type	Purpose	Use
Unsupervised	Detect unusual network activity	Learns from (entity, IPv4 address) pairs Entity can be Account ID, User ID For a given entity, IPv4 address pair, it returns a score High score indicates unusual event – a website can trigger MFA

SageMaker Ground Truth and Neo

SageMaker Ground Truth

Automatic Labeling

- Learns based on examples you provide
- Very cost-effective

Manual Labeling

- Human Labelers – Mechanical Turk
- Manages workflow

SageMaker Neo

- Run Machine Learning Algorithms anywhere in the Cloud and at Edge Locations
- Latency is critical
- Cross Compilation capability that can optimize your algorithms to run on:
    - Intel
    - NVIDIA
    - ARM
    - And other hardware

Support for ML Frameworks

- Use SageMaker to train and host models using popular frameworks
- SageMaker provides built-in container images for Apache MxNet, TensorFlow, scikit-learn, PyTorch, Chainer, SparkML and more

Containers

“Amazon SageMaker makes extensive use of *Docker containers* for build and runtime tasks”

“Amazon SageMaker provides pre-built Docker images for its built-in algorithms and the supported deep learning frameworks used for training and inference. By using containers, you can train machine learning algorithms and deploy models quickly and reliably at any scale”

Reference: https://docs.aws.amazon.com/sagemaker/latest/dg/your-algorithms.html

Use Apache Spark with SageMaker

- SageMaker Apache Spark Library in Python and Scala
- Directly read DataFrames in Spark Clusters
- SageMakerEstimator automatically converts DataFrames to Protobuf format
- Train and Host using SageMaker

Popular Framework Support

- TensorFlow
- MxNet
- scikit-learn
- PyTorch
- Chainer
- SparkML

SageMaker provides SDKs and pre-built docker images to train and host models using above frameworks

Use Your own algorithms

- Host your custom algorithms on SageMaker
- Use a runtime and language of your choice
- Build Containers that conform to SageMaker Specification
- Train and Host on SageMaker

Deep Learning AMIs

- Launch EC2 instances preconfigured with all the tools and Deep Learning Framework
- Modify DL frameworks or extend them
- Contributors to DL frameworks
- Troubleshooting framework level issues

SageMaker – Training and Hosting

Options	Usage Scenario
Built-in Algorithms	Training algorithms provided by SageMaker Easy to use and scale Optimized for AWS Cloud
Pre-built Container Images	Supports popular frameworks like MxNet, TensorFlow, scikit-learn, PyTorch Flexibility to use wide selection of algorithms
Extend Pre-built Container Images	Extend pre-built container images to your needs
Custom Container Images	Custom algorithm, language and framework

Chandra Lingam

50,000+ Students

Up-to-date Content