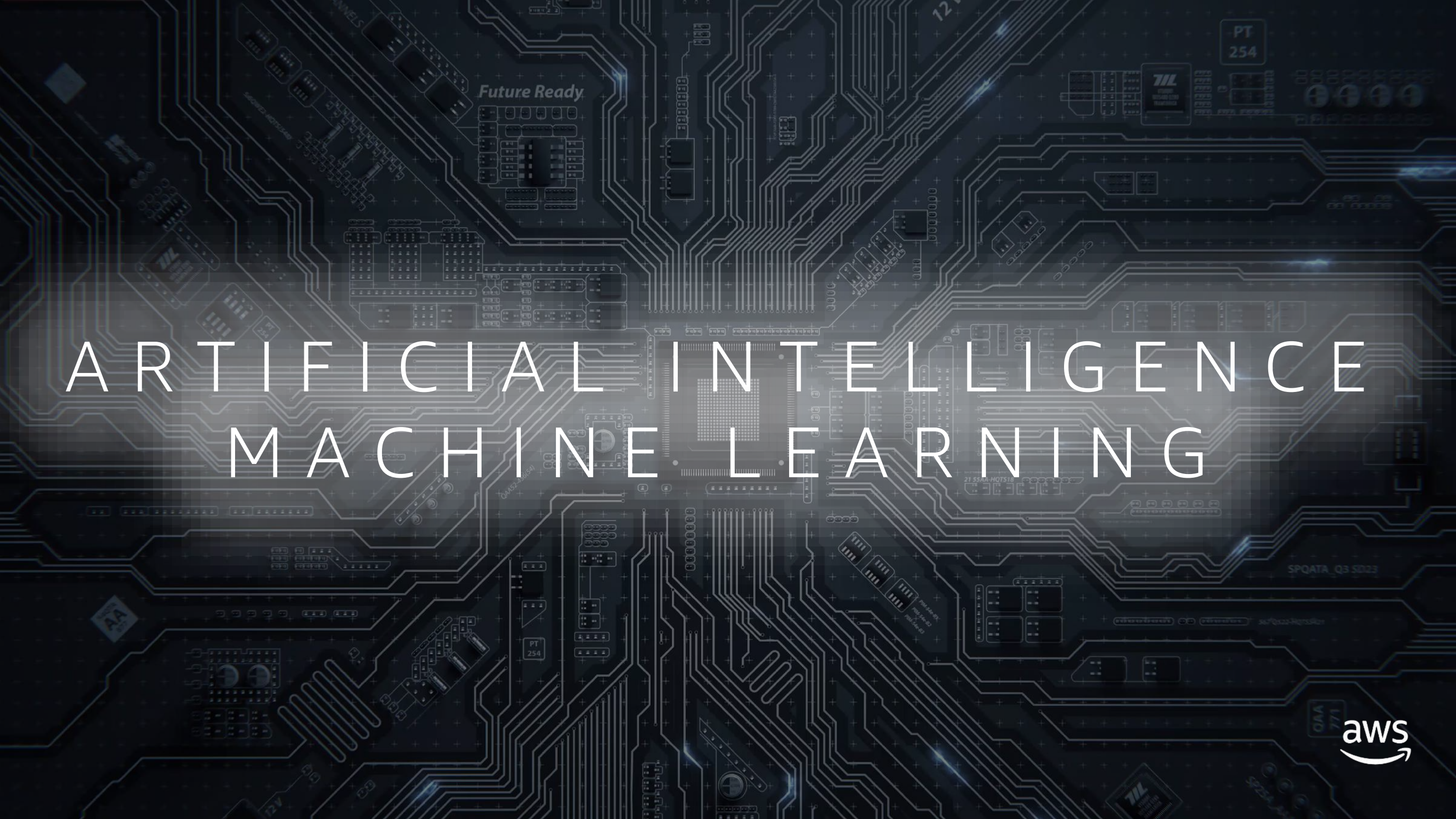




Keynote

Glenn Gore
Lead Architect
AWS

Olivier Klein
Head of Emerging Technologies, APAC
AWS

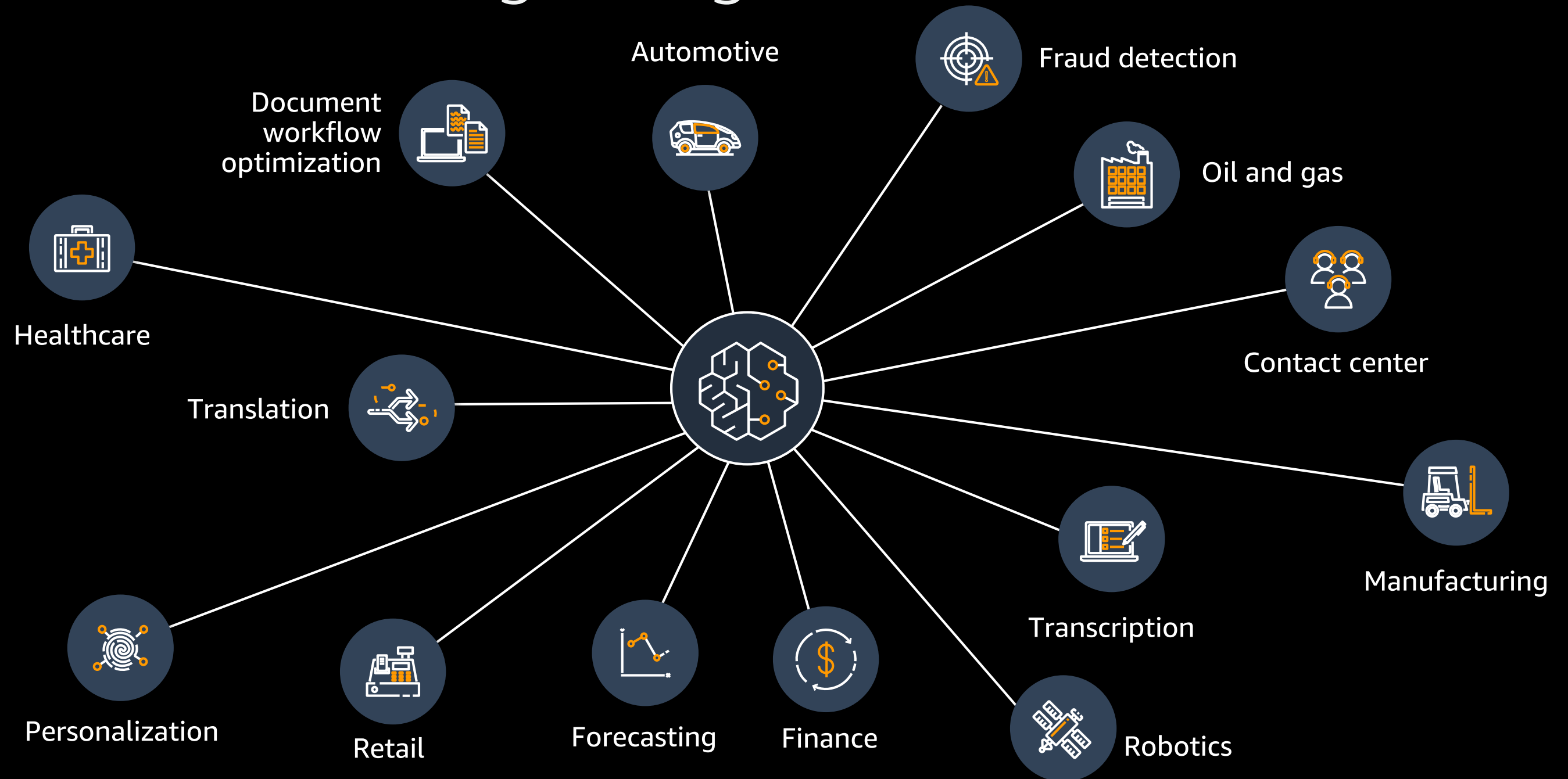


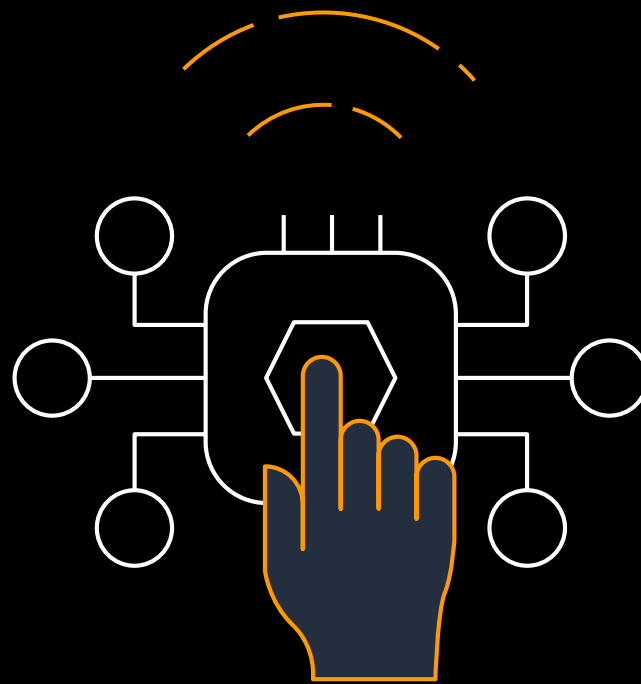
ARTIFICIAL INTELLIGENCE MACHINE LEARNING

Entering a new golden age for machine learning

Innovation is everywhere
so are
opportunities to learn

The reach of ML is growing





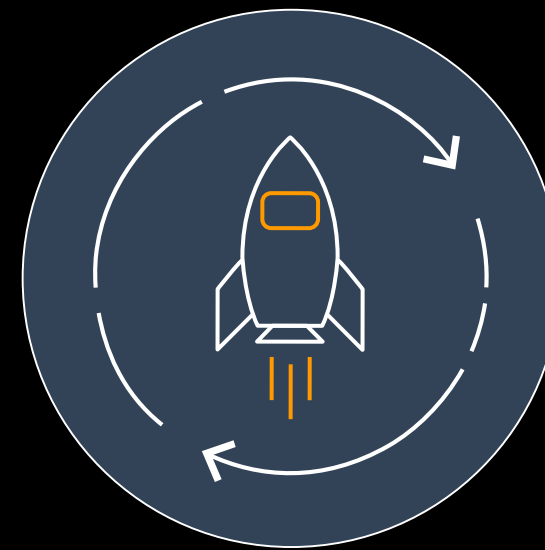
Put machine learning in the
hands of every developer



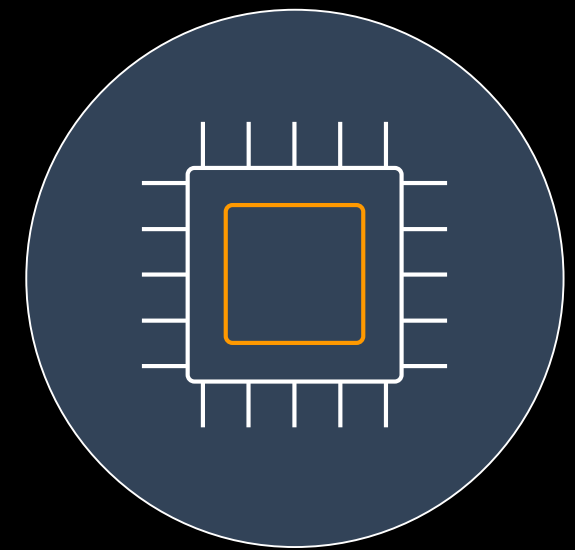
AI Services



ML Services



Frameworks



Infrastructure

AI Services

VISION



Amazon
Rekognition

SPEECH



Amazon
Polly



Amazon
Transcribe

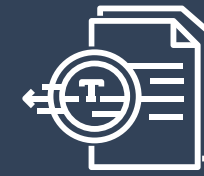
TEXT



Amazon
Comprehend



Amazon
Translate



Amazon
Textract

SEARCH



Amazon
Kendra

PERSONALIZATION



Amazon
Personalize

FRAUD



Amazon
Fraud Detector

FORECASTING



Amazon
Forecast

DEVELOPMENT



Amazon
CodeGuru

CONTACT CENTER



Amazon Connect
with Contact Lens

CHATBOT



Amazon
Lex

ML Services

End-to-end ML with AWS

Prepare

Build

Train & Tune

Deploy & Manage

Amazon SageMaker Studio

Integrated Development environment(IDE) for Machine Learning

Amazon SageMaker Autopilot

Automatically build and train models

1-Click Deployment

Supports real-time, batch & multi-model

Amazon SageMaker GroundTruth

Build and manage training dataset

Amazon SageMaker Notebooks

One-click notebooks with elastic compute

1-Click Training

Supports supervised, unsupervised & RL

Amazon SageMaker Model Monitor

Automatically detect concept drift

Amazon Sagemaker Processing

Supports Python or Spark

AWS Marketplace & In-Built Algorithms

Pre-built algorithms, models, and data

Automatic Model Tuning

One-click hyperparameter optimization

Amazon SageMaker Neo

Train once, deploy anywhere

Amazon SageMaker Experiments

Capture, organize, and compare every step

Amazon Elastic Inference

Auto scaling for 75% less

Amazon SageMaker Debugger

Debug and profile training runs

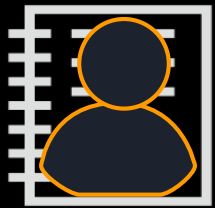
Amazon Augmented AI

Add human review of model predictions



Amazon SageMaker Studio

Fully integrated development environment (IDE) for machine learning



Collaboration at
scale

Share notebooks
without tracking
code dependencies



Easy experiment
management

Organize, track, and
compare thousands
of experiments



Automatic model
generation

Get accurate models
with full visibility &
control without
writing code



Higher quality ML
models

Automatically debug
errors, monitor
models, & maintain
high quality



Increased
productivity

Code, build, train,
deploy, & monitor in
a unified visual
interface



Amazon SageMaker Autopilot

Automatic model creation with full visibility & control



Quick to start

Provide your data in a tabular form & specify target prediction



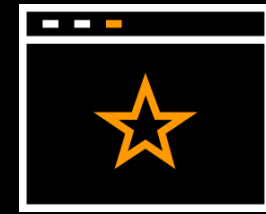
Automatic model creation

Get ML models with feature engineering & model tuning automatically done



Visibility & control

Get notebooks for your models with source code



Recommendations & Optimization

Get a leaderboard & continue to improve your model

awsFileEditViewRunKernelGitTabsSettingsHelp

+
/ automl-preview /

NameLast Modified

bank-additional12 hours ago

model12 hours ago

sagemaker_auto...3 hours ago

sagemaker_automl_direct_miX

my-sagemaker-autopilotX

Deploy modelX

EXPERIMENT: MY-SAGEMAKER-AUTOPILOT

Open candidate generation notebook

Open data exploration notebook

TrialsJob profile

TRIALS

0 row selected

Deploy model

Trial name	Status	Start time	End time	▼	Objective
my-sagemaker-tuning-job-...	Completed	9 hours ago			0.9206119775772095
my-sagemaker-tuning-job-...	Completed	9 hours ago			0.9202479720115662
my-sagemaker-tuning-job-...	Completed	7 hours ago			0.9200050234794617
my-sagemaker-tuning-job-...	Completed	7 hours ago			0.9195190072059631
my-sagemaker-tuning-job-...	Completed	9 hours ago			0.9191550016403198
my-sagemaker-tuning-job-...	Completed	7 hours ago			0.9190340042114258
my-sagemaker-tuning-job-...	Completed	8 hours ago			0.9189119935035706
my-sagemaker-tuning-job-...	Completed	8 hours ago			0.9186699986457825
my-sagemaker-tuning-job-...	Completed	8 hours ago			0.9186699986457825

0\$1

my-sagemaker-autopilot

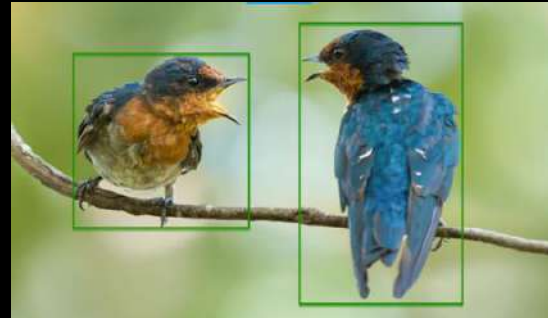
© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws

Amazon SageMaker Ground Truth



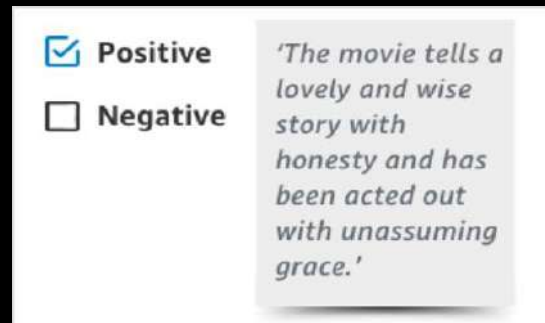
Image classification



Bounding boxes



Semantic segmentation



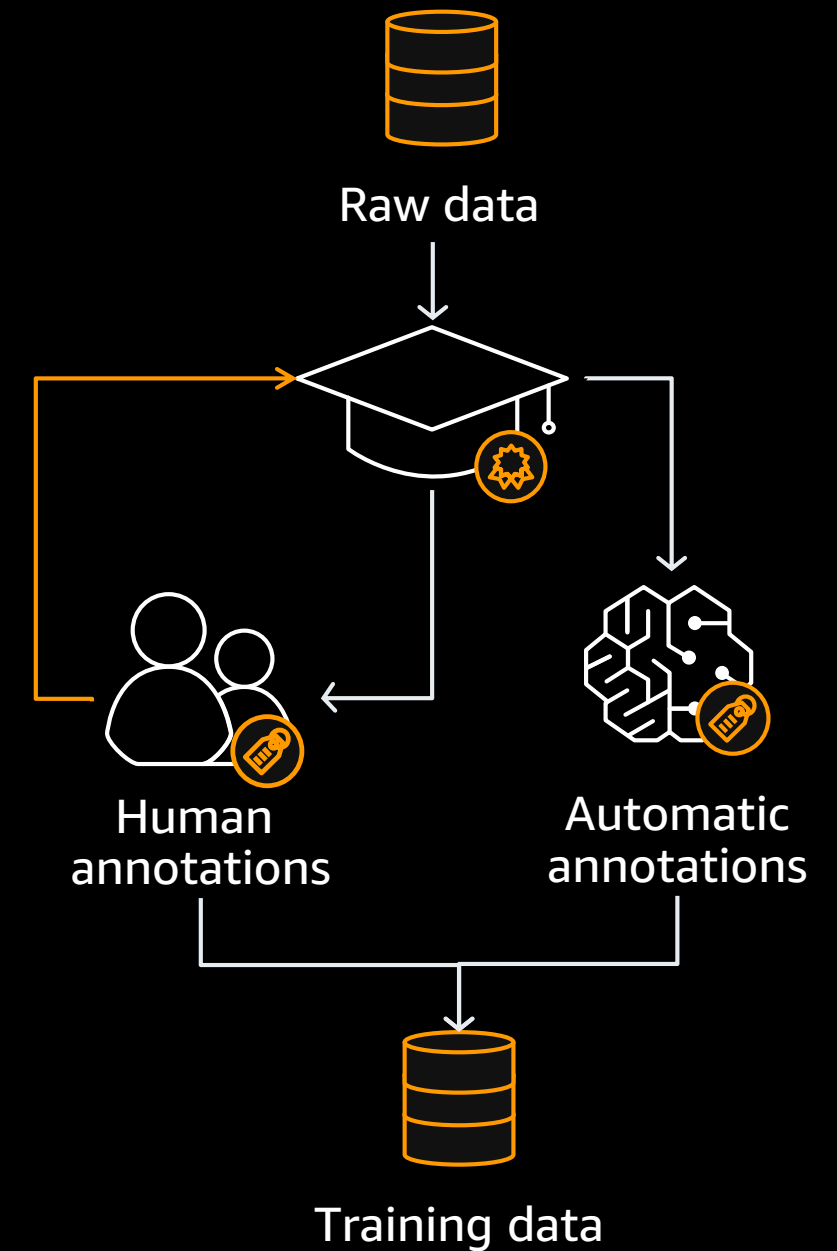
Text classification



Named entity recognition



Custom



Amazon SageMaker Built-In Algorithms

Classification

- Linear Learner
- XGBoost
- KNN

Computer Vision

- Image Classification
- Object Detection
- Semantic Segmentation

Topic Modeling

- LDA
- NTM

Forecasting

- DeepAR

Working with Text

- BlazingText
- Supervised
- Unsupervised

Recommendation

- Factorization Machines

Clustering

- KMeans

Regression

- Linear Learner
- XGBoost

Anomaly Detection

- Random Cut Forests
- IP Insights

Feature Reduction

- PCA
- Object2Vec

Sequence Translation

- Seq2Seq

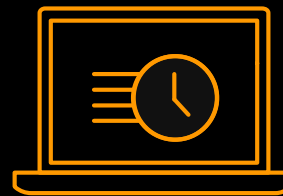


AWS Marketplace for Machine Learning

You can shop for algorithms, models, and data in AWS Marketplace



Browse or search
AWS Marketplace



Subscribe in a
single click



Available in
Amazon SageMaker

Natural language processing

Ranking

Text OCR

Computer vision

Named entity recognition

Video classification

Speech recognition

Text-to-speech

3D images

Text classification

Speaker identification

Anomaly detection

Text generation

Object detection

Regression

Text clustering

Handwriting recognition

Grammar and parsing

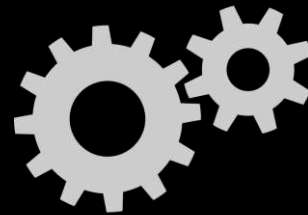
Amazon SageMaker Automatic Model Tuning

Automatically tune hyperparameters in your algorithms



Tuning at scale

Adjust thousands of different combinations of algorithm parameters



Automated

Uses ML to find the best parameters



Faster

Eliminate days or weeks of tedious manual work

Decision Trees

Tree depth
Max leaf nodes
Gamma
Lambda
Alpha

Examples

Neural Networks

Number of layers
Hidden layer width
Learning rate
Embedding dimensions
Dropout

Amazon SageMaker Experiments

Organize, track, and compare training experiments



Tracking at scale

Track parameters
& metrics across
experiments &
users



Custom organization

Organize
experiments by
teams, goals, &
hypotheses



Visualization

Easily visualize
experiments and
compare



Metrics and logging

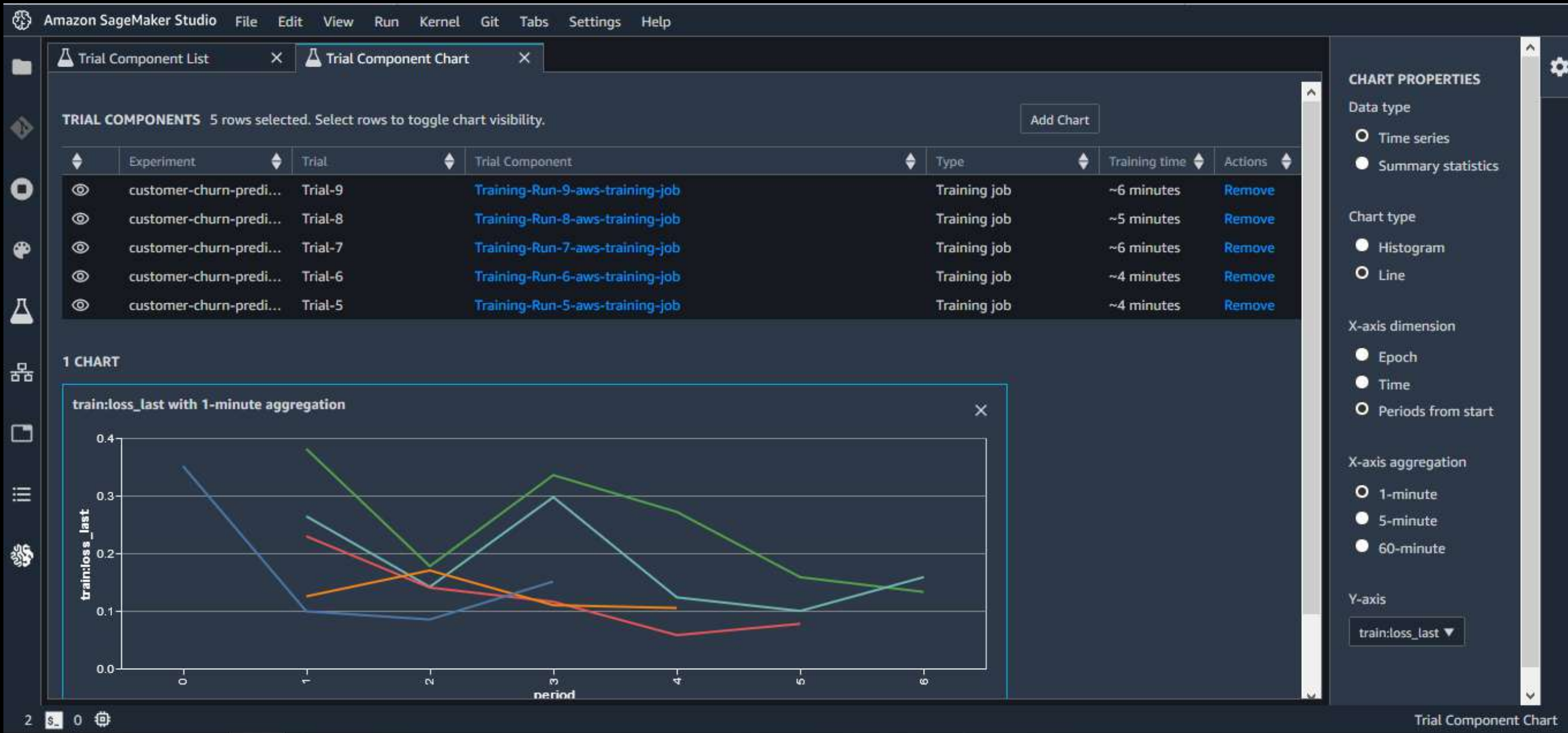
Log custom
metrics using the
Python SDK &
APIs



Fast Iteration

Quickly go back
& forth
& maintain
high-quality





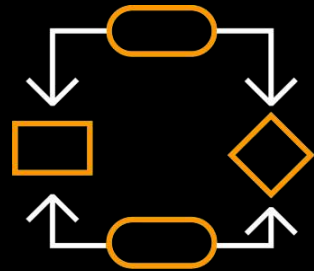
Amazon SageMaker Debugger

Analysis and debugging, explainability, and alert generation



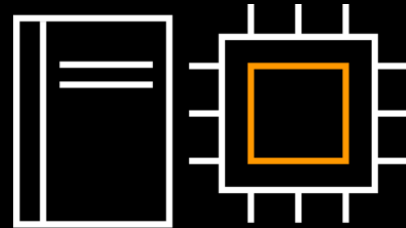
Relevant data
capture

Data is
automatically
captured for
analysis



Data analysis &
debugging

Analyze & debug
data with
no code changes



Automatic error
detection

Errors are
automatically
detected based
on rules



Improved productivity
with alerts

Take corrective
action based
on alerts



Visual analysis
and debugging

Visually analyze
& debug from
SageMaker
Studio

Amazon SageMaker Studio

File Edit View Run Kernel Git Tabs Settings Help

SMDEbugger-CloudWatch-Lo

+

✂

📄

📄

▶

■

↺

🔍

git

conda_tensorflow_p36

Using SageMaker Rules

In this example we'll demonstrate how to use SageMaker rules to be evaluated against your training. You can find the list of SageMaker rules and the configurations best suited for using them here.

We specify a few rules that check for overfitting, decrease in loss across epochs and for saturated activations.

```
[8]: estimator = TensorFlow(
    role=sagemaker.get_execution_role(),
    base_job_name='mnist-tensorflow-example',
    train_instance_count=1,
    train_instance_type='ml.p3.2xlarge',
    image_name=cpu_training_image,
    entry_point=entrypoint_script,
    framework_version='1.15',
    py_version='py3',
    train_max_run=3600,
    script_mode=True,
    sagemaker_session=sess,
    ## New parameter
    rules = [ Rule.sagemaker(rule_configs.vanishing_gradient()),
              Rule.custom(name='Overfitting', # used to identify the rule
                           image_uri='759209512951.dkr.ecr.us-west-2.amazonaws.com',
                           instance_type='ml.c4.xlarge', # instance type to run the
                           source='my_custom_rule.py', # path to the rule source file
                           rule_to_invoke='CustomGradientRule', # name of the class
                           volume_size_in_gb=400, # EBS volume size required to be
                           collections_to_save=[CollectionConfig(name='losses')],
                           rule_parameters={
                               "threshold": "20.0" # this will be used to initialize
                           }) ],
    hyperparameters = {'num_epochs': 100 }
)
```

Note that Sagemaker-Debugger is only supported for py_version='py3' currently.

Let's start the training by calling `fit()` on the MXNet estimator

```
[9]: # After calling fit, SageMaker will spin off 1 training job and 1 rule job for you
      # The rule evaluation status(es) will be visible in the training logs
      # at regular intervals

      estimator.fit(wait=False)
```

Result

Describe Trial Component

Experiment: Unassigned

Trial: Unassigned

Trial stages

Charts

Metrics

Parameters

Artifacts

AWS Settings

Debugger

mnist-tensorflow-example-2019-12-02-09-52-13-126-aws-training-job

Created

15 minutes ago

Status

Issues Found

Last modified

4 minutes ago

Rule name

VanishingGradient

Job ARN

arn:aws:sagemaker:us-west-2:3...

Issues Found

4 minutes ago

Overfitting

arn:aws:sagemaker:us-west-2:3...

Trial Component Chart

TRIAL COMPONENTS

1 rows selected. Select rows to toggle chart visibility.

Add Chart

Experiment	Trial	Trial Component	Type	Train
N/A	N/A	mnist-tensorflow-example-2019-12-02-09-52-13-126-aws-trainin...	Training job	~10

2 CHARTS

sparse_softmax_cross_entropy_loss/value:0_avg with 1-minute aggregation

sparse_softmax_cross_entropy_loss/value:0_avg

40

30

20

10

0

5

10

11

period

trialComponentName

mnist-tensorflow-example-2019-1...

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

Amazon SageMaker Model Monitor

Continuous monitoring of models in production



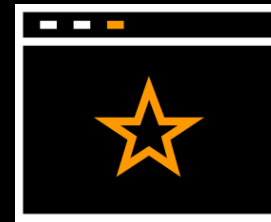
Automatic data collection

Data is automatically collected from your endpoints



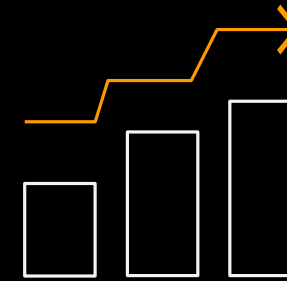
Continuous Monitoring

Define a monitoring schedule and detect changes in quality against a pre-defined baseline



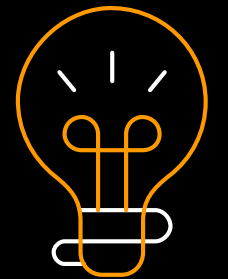
Flexibility with rules

Use built-in rules to detect data drift or write your own rules for custom analysis



Visual data analysis

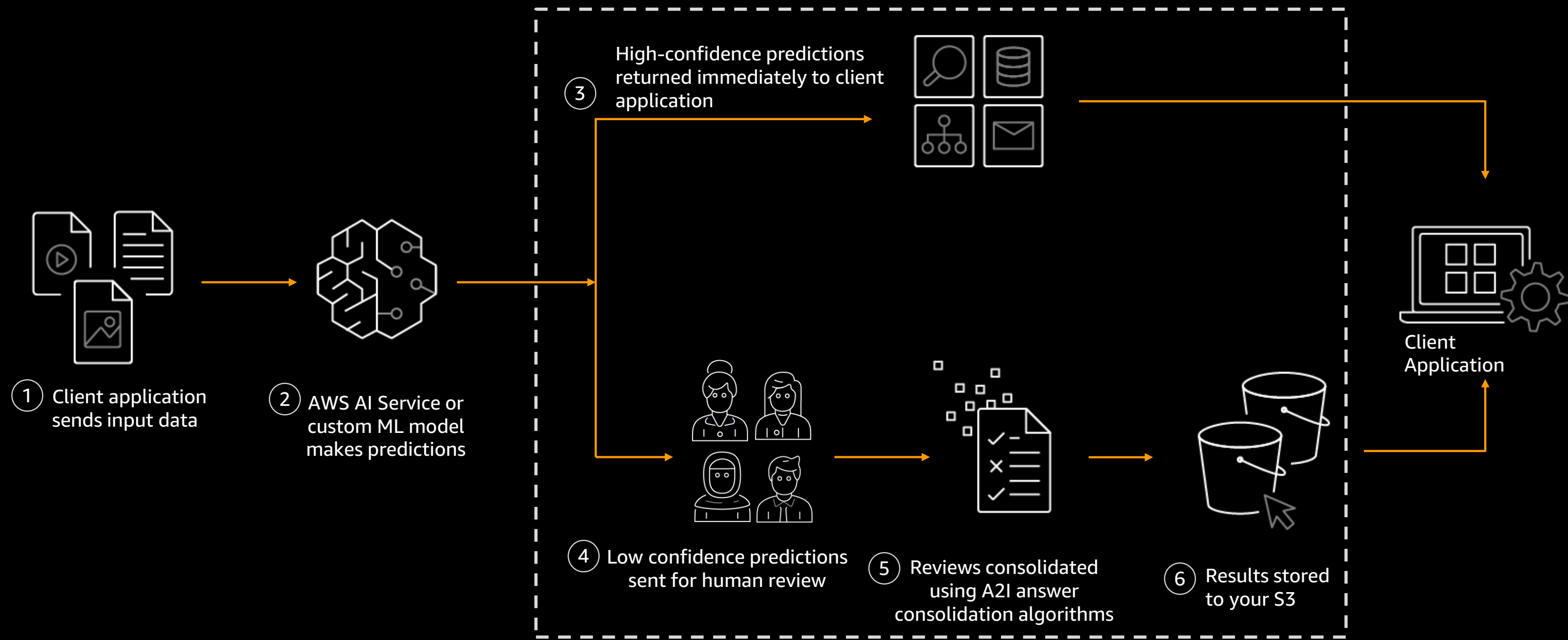
See monitoring results, data statistics, and violation reports in SageMaker Studio



CloudWatch Integration

Automate corrective actions based on Amazon CloudWatch alerts

How Amazon A2I works



Instructions

View full instructions

View tool guide

Click on a key-value block or input box to highlight the corresponding key-value pair in the document.

If it is a key-value pair, review the content for the key or value. If the content is incorrect, correct it.

If it's not a key-value relationship, choose **No**.

Jane Doe123 Any Street, Any Town, USA

Key-value pair

Yes

No

Jane Doe

Key not found

123 Any Street,

Value is blank

If you can't find the key in the document, choose **Key not found**.

Key-value pair

Yes

No

Mail address

Key not found

Value is blank

If the content of a field is empty, choose **Value is blank**.

Cell number

Review the key-value pairs listed on the right and correct them if they don't match the following document.

Employment Application

Application Information

Full Name:

Jane Doe

Phone number:

550-0100

Home address:

123 Any Street, Any Town, USA

Mail address:

same as home address

Zoom in

Zoom out

Move

Fit image

Key-value pairs to review

Key-value pair

Yes

No

Full name:

Jaue

Key not found

Jaue Done

Value is blank

Key-value pair

Yes

No

Phone number:

550-0100

Value is blank

No adjustment needed

Submit

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

aws

ML for database developers and BI analysts



Amazon Aurora



Amazon SageMaker

Application developers can access ML algorithms via the familiar **SQL language**



Amazon Aurora



Amazon Comprehend

Use out of the box, high-throughput integrations with Amazon SageMaker and Amazon Comprehend



Amazon Athena



Amazon SageMaker

Real-time predictions on transactional data without unnecessary integrations



Amazon QuickSight



Amazon SageMaker

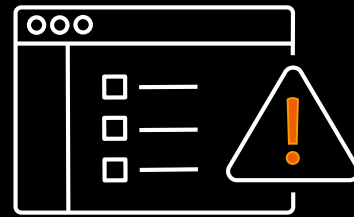
From SQL to ML-driven insights

Find suspected fraudulent transactions



```
CREATE TRIGGER insert_check
BEFORE INSERT ON approved_sales
FOR EACH ROW
BEGIN
  IF
    is_transaction_fraudulent(column1,
    column2, column3 ...) = 'True' THEN
    rollback; END IF;
END;
```

Flag comments with negative sentiment



```
SELECT FROM product_reviews
WHERE
  aws_comprehend.detect_sentiment
(review_text, 'EN') = 'NEGATIVE'
```

Sort customers by predicted future spend



```
SELECT FROM customers
ORDER BY predicted_future_spend
(column1, column2, ...)
```

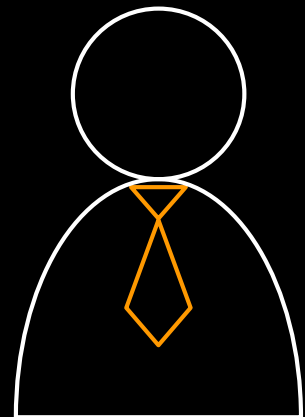
Frameworks

AWS is framework agnostic

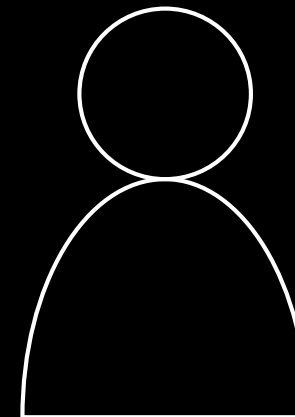
Choose from popular frameworks



Run them fully managed



Or run them yourself



The best place to run TensorFlow

Stock
TensorFlow

65%

scaling efficiency
with 256 GPUs

AWS-Optimized
TensorFlow

90%

scaling efficiency
with 256 GPUs

Benchmark:
Train Resnet50

30m

training time

14m

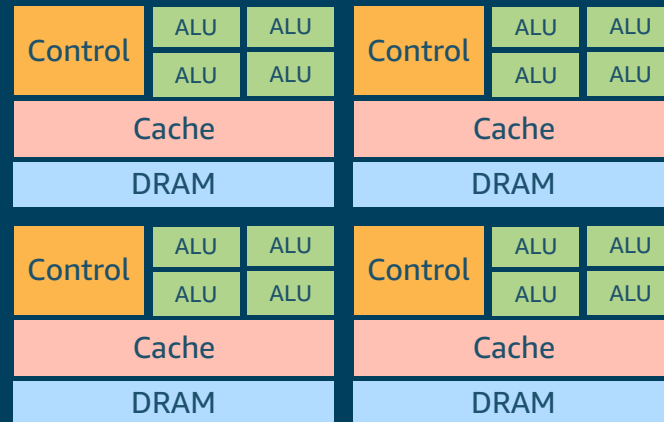
training time

**Fastest time
for TensorFlow**

Available with
Amazon SageMaker
and the AWS Deep
Learning AMIs

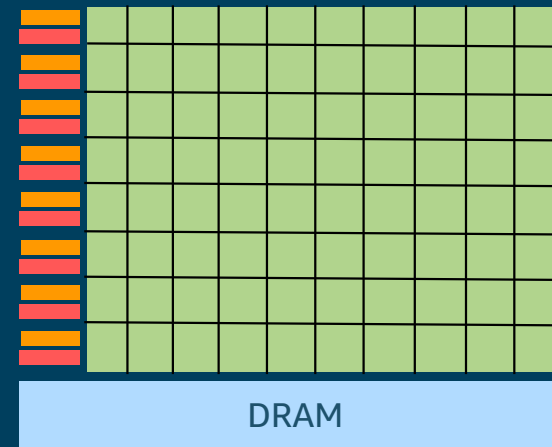
Infrastructure

CPU



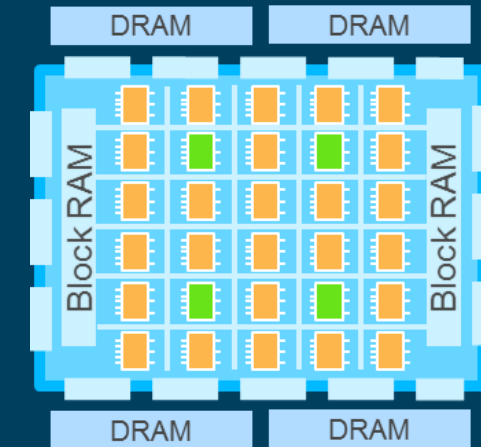
- 10s–100s of processing cores
- Optimized for general-purpose computing

GPU



- 1,000s of processing cores
- Highly effective at parallel execution

FPGA



- Millions of programmable digital logic cells
- Hardware timed parallel execution

Training of machine learning models, can take advantage of parallel compute architecture of GPUs and FPGAs

GPUs for machine learning training

Routines for training ML models fundamentally map to **matrix multiplications**, this coupled with **extremely high memory bandwidth** makes GPUs ideal for training

Matrix A		Matrix B		Matrix C																											
<table><tr><td>a_{11}</td><td>a_{12}</td><td>a_{13}</td></tr><tr><td>a_{21}</td><td>a_{22}</td><td>a_{23}</td></tr><tr><td>a_{31}</td><td>a_{32}</td><td>a_{33}</td></tr></table>	a_{11}	a_{12}	a_{13}	a_{21}	a_{22}	a_{23}	a_{31}	a_{32}	a_{33}	\times	<table><tr><td>b_{11}</td><td>b_{12}</td><td>b_{13}</td></tr><tr><td>b_{21}</td><td>b_{22}</td><td>b_{23}</td></tr><tr><td>b_{31}</td><td>b_{32}</td><td>b_{33}</td></tr></table>	b_{11}	b_{12}	b_{13}	b_{21}	b_{22}	b_{23}	b_{31}	b_{32}	b_{33}	$=$	<table><tr><td>$a_{11}.b_{11} + a_{12}.b_{21} + a_{13}.b_{31}$</td><td>$a_{11}.b_{12} + a_{12}.b_{22} + a_{13}.b_{32}$</td><td>$a_{11}.b_{13} + a_{12}.b_{23} + a_{13}.b_{33}$</td></tr><tr><td>$a_{21}.b_{11} + a_{22}.b_{21} + a_{23}.b_{31}$</td><td>$a_{21}.b_{12} + a_{22}.b_{22} + a_{23}.b_{32}$</td><td>$a_{21}.b_{13} + a_{22}.b_{23} + a_{23}.b_{33}$</td></tr><tr><td>$a_{31}.b_{11} + a_{32}.b_{21} + a_{33}.b_{31}$</td><td>$a_{31}.b_{12} + a_{32}.b_{22} + a_{33}.b_{32}$</td><td>$a_{31}.b_{13} + a_{32}.b_{23} + a_{33}.b_{33}$</td></tr></table>	$a_{11}.b_{11} + a_{12}.b_{21} + a_{13}.b_{31}$	$a_{11}.b_{12} + a_{12}.b_{22} + a_{13}.b_{32}$	$a_{11}.b_{13} + a_{12}.b_{23} + a_{13}.b_{33}$	$a_{21}.b_{11} + a_{22}.b_{21} + a_{23}.b_{31}$	$a_{21}.b_{12} + a_{22}.b_{22} + a_{23}.b_{32}$	$a_{21}.b_{13} + a_{22}.b_{23} + a_{23}.b_{33}$	$a_{31}.b_{11} + a_{32}.b_{21} + a_{33}.b_{31}$	$a_{31}.b_{12} + a_{32}.b_{22} + a_{33}.b_{32}$	$a_{31}.b_{13} + a_{32}.b_{23} + a_{33}.b_{33}$
a_{11}	a_{12}	a_{13}																													
a_{21}	a_{22}	a_{23}																													
a_{31}	a_{32}	a_{33}																													
b_{11}	b_{12}	b_{13}																													
b_{21}	b_{22}	b_{23}																													
b_{31}	b_{32}	b_{33}																													
$a_{11}.b_{11} + a_{12}.b_{21} + a_{13}.b_{31}$	$a_{11}.b_{12} + a_{12}.b_{22} + a_{13}.b_{32}$	$a_{11}.b_{13} + a_{12}.b_{23} + a_{13}.b_{33}$																													
$a_{21}.b_{11} + a_{22}.b_{21} + a_{23}.b_{31}$	$a_{21}.b_{12} + a_{22}.b_{22} + a_{23}.b_{32}$	$a_{21}.b_{13} + a_{22}.b_{23} + a_{23}.b_{33}$																													
$a_{31}.b_{11} + a_{32}.b_{21} + a_{33}.b_{31}$	$a_{31}.b_{12} + a_{32}.b_{22} + a_{33}.b_{32}$	$a_{31}.b_{13} + a_{32}.b_{23} + a_{33}.b_{33}$																													

These operations can be parallelized across **1,000s** of core available in a typical **GPU**

P3 Instances: GPU Compute

Up to 8 NVIDIA Tesla V100 GPUs

1 PetaFLOPs of computational performance

300 GB/s GPU-to-GPU communication (NVLink)

P3dn - Most powerful GPU instance in the cloud

Efficiently scale ML training and HPC simulations across multiple instances with 100Gbps of networking throughput

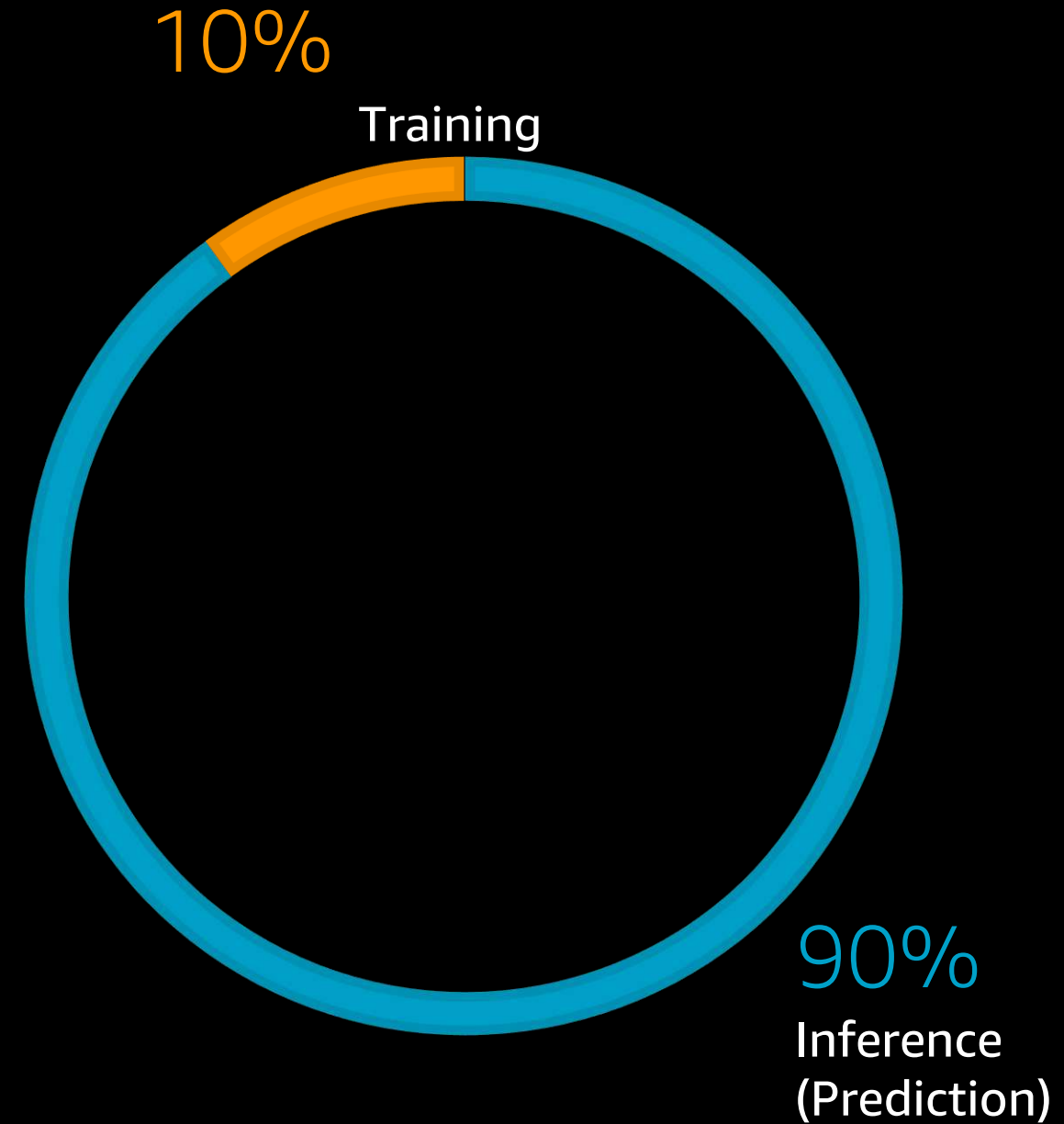
Fast access to training or simulation data via Amazon S3, network attached file systems or local instance storage

Train larger ML models or process more data via latest NVIDIA V100 GPU with 32GB of GPU memory

P3.2xlarge	P3.8xlarge	P3.16xlarge
1 V100 GPU	4 V100 GPU	8 V100 GPU
8 vCPU	32 vCPU	64 vCPU
61 GB Mem	244 GB Mem	488 GB Mem

p3dn.24xlarge
8 V100 GPU
96 vCPU
768 GB Mem
2 TB NVME SSD
100 Gbps Throughput

Predictions drive
complexity and cost
in production



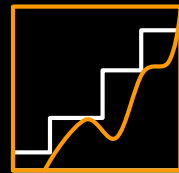
Amazon Elastic Inference

Reduce deep learning inference costs up to 75%



Lower
inference costs

Integrated with
Amazon EC2,
Amazon SageMaker
and Amazon ECS



Match capacity
to demand

Support for TensorFlow,
Apache MXNet



Available between
1 to 32 TFLOPS
per accelerator

Single and mixed-
precision operations

ML inference deployment options on Amazon EC2

Custom chip EC2 Inf1 instances

- Applications that leverage common ML frameworks
- Powered by AWS Inferentia
- Best price/performance for ML inferencing in the cloud
Up to 40% lower cost per inference and up to 3x higher throughput than G4 instances

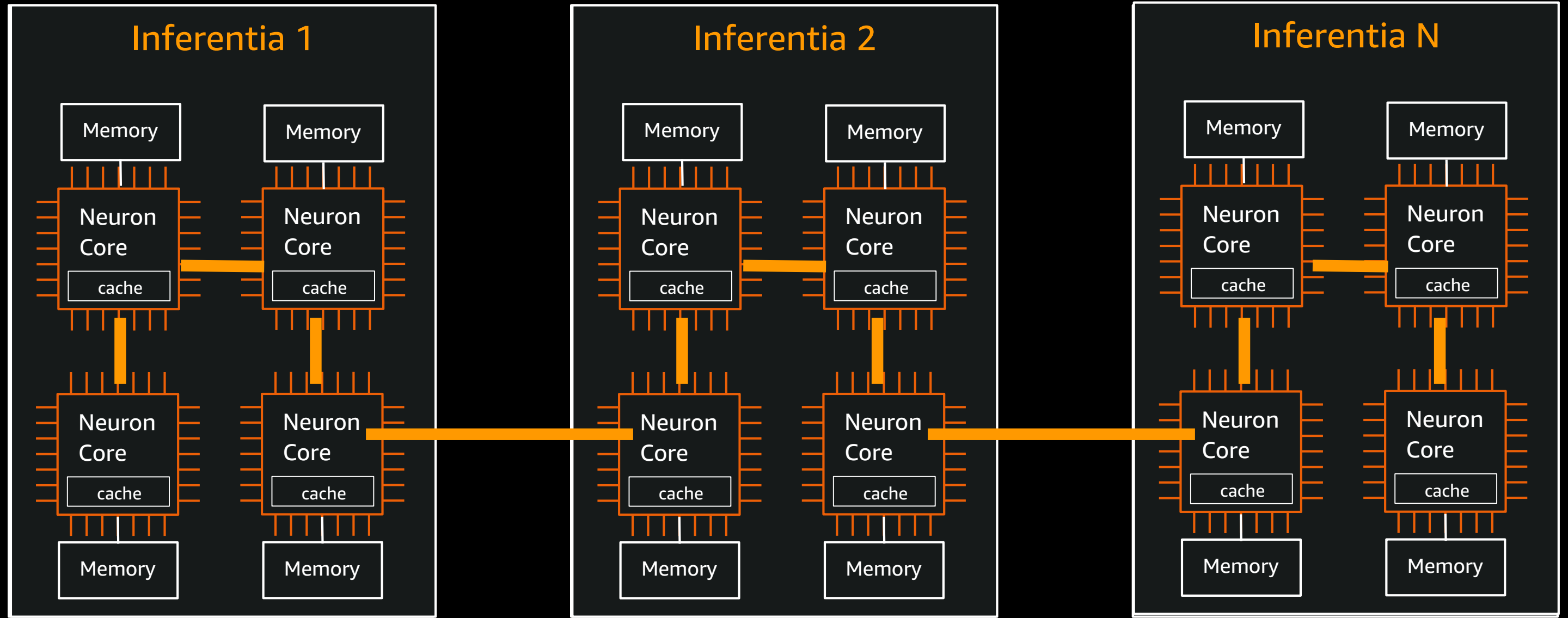
GPU based EC2 G4 instances

- Applications that require access to CUDA, CuDNN or TensorRT libraries
- Amazon EC2 G4 instances based on NVIDIA T4 GPUs

CPU based EC2 C5 instances

- Small models and low sensitivity to performance
- Intel Skylake CPUs
Support for AVX-512/VNNI instruction set

Scaling



Introducing AWS Neuron

Software suite enabling high-performance deep learning inference on Inferentia

Compiler

Run time

Profiling and debug tools

Supports all major framework

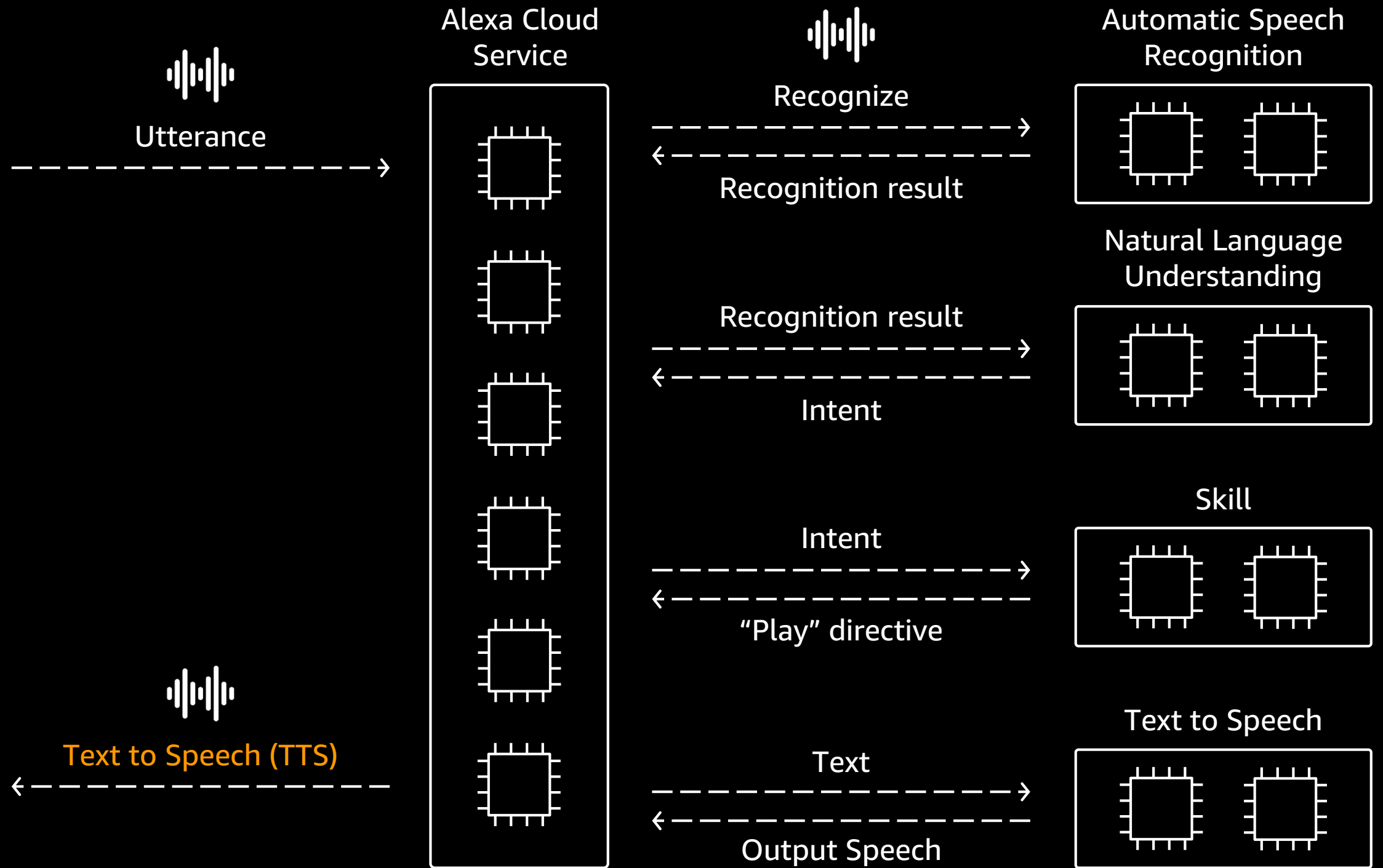
 TensorFlow

 mxnet

 PyTorch

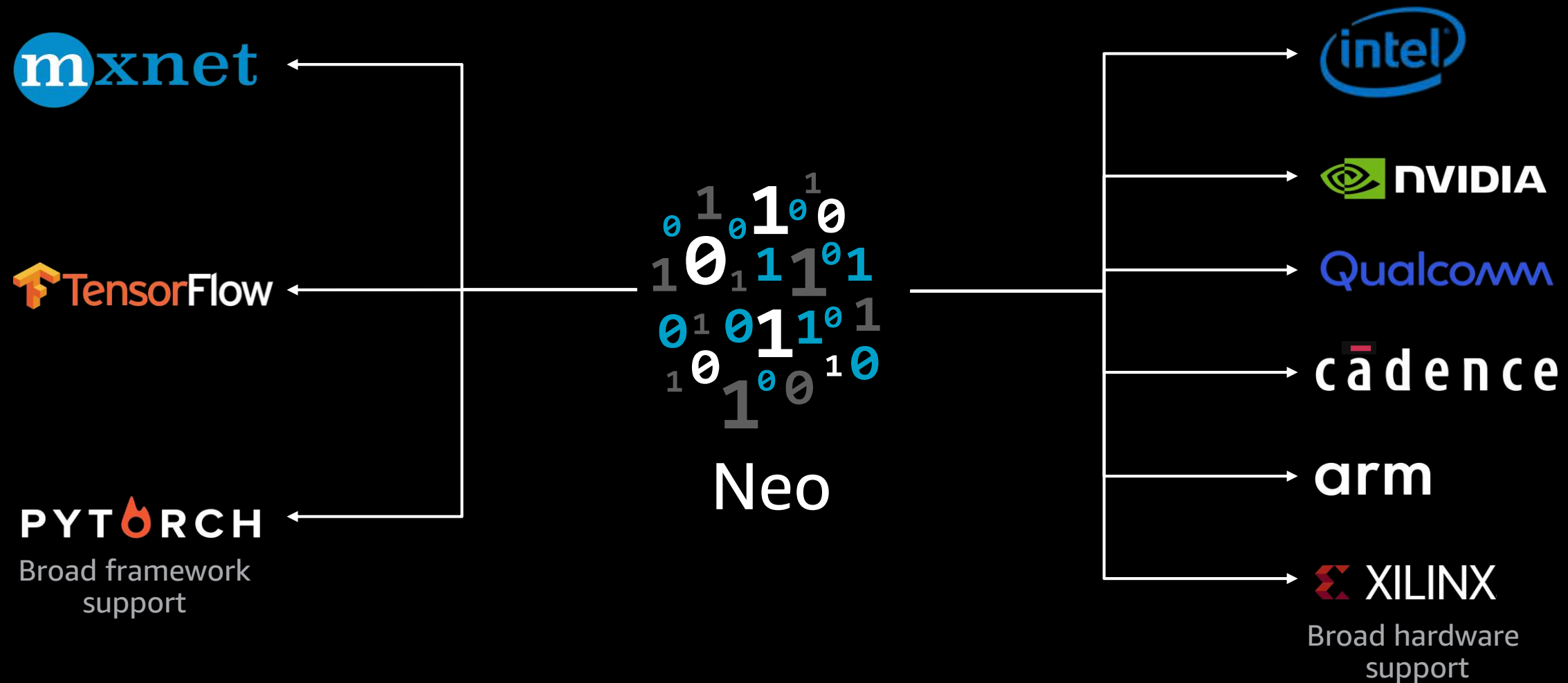


github.com/aws/aws-neuron-sdk



Amazon SageMaker Neo

Train once and run anywhere with 2x performance



Learn ML

Get hands-on experience with AI/ML



AWS DeepLens

Computer
Vision



AWS DeepRacer

Reinforcement
Learning (RL)



AWS DeepComposer

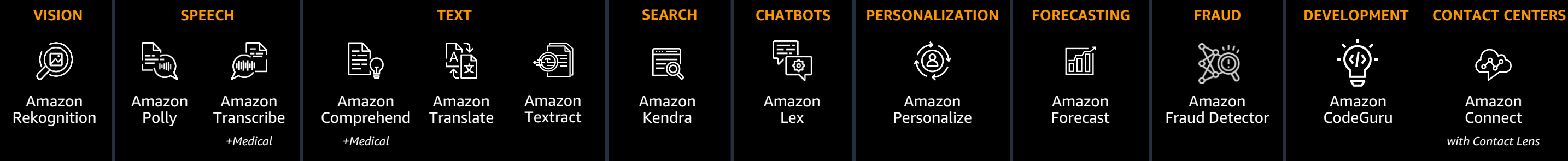
Generative Adversarial
Networks (GAN)

In Closing

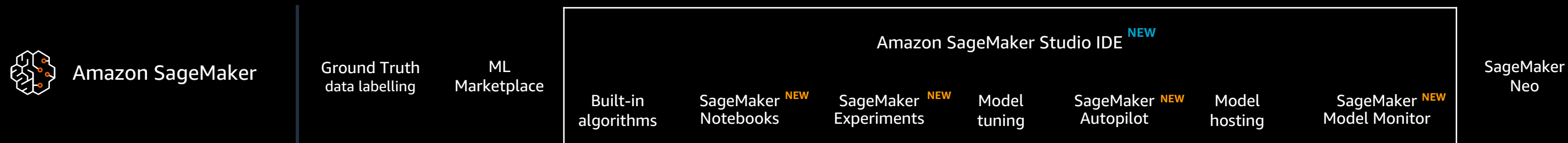
The AWS ML Stack

Broadest and most complete set of Machine Learning capabilities

AI SERVICES



ML SERVICES



ML FRAMEWORKS & INFRASTRUCTURE



KEYNOTE & CLOSING



6 TRACKS

Innovation at Amazon

AI/ML Fundamentals

AI Services and Applications

Accelerate your ML Journey

Build, Train and Deploy ML Models

AI/ML Services and Devices



5 LOCALISED TRACKS

Korean Tracks

Bahasa Indonesian Track

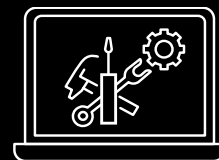
Mandarin Track

Spanish Track

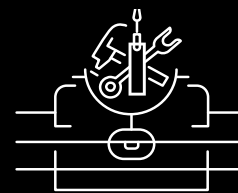
Portuguese Track



ACTIVITIES



AI/ML
SHOWCASE



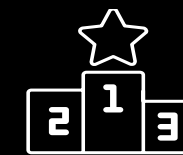
BUILDERS'
ZONE



AWS DEEPRACER
ZONE



HANDS-ON
LABS



INNOVATE
CHALLENGE

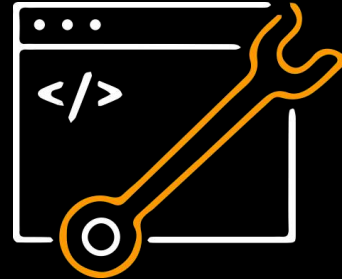


ASK THE
EXPERTS

ML Learning Paths



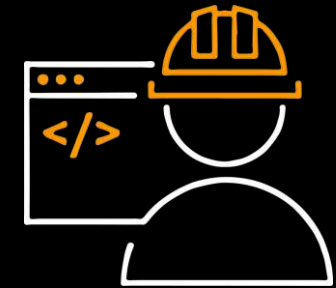
Business decision
makers



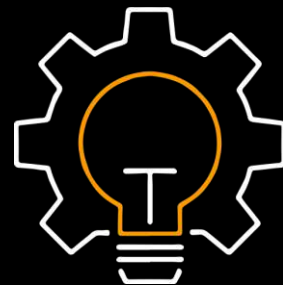
Developers



Data scientists



Data platform
engineers



AWS certified
machine learning
– specialty exam

Additional learning path for exam readiness available
Learn more at <https://aws.training/machinelearning>

Thank You for Attending AWS Innovate

We hope you found it interesting! A kind reminder to **complete the survey**.
Let us know what you thought of today's event and how we can improve
the event experience for you in the future.



aws-apac-marketing@amazon.com



twitter.com/AWSCloud



facebook.com/AmazonWebServices



youtube.com/user/AmazonWebServices



slideshare.net/AmazonWebServices



twitch.tv/aws

Thank you!