Mastering Amazon SageMaker

Model build, train and tune using Amazon SageMaker

Name: Mani Khanuja

Role: Sr. AI/ML Specialist SA

Email: mankhanu@amazon.com

Agenda

- Overview of Amazon SageMaker
- Module 2 – SageMaker Building ML models
  - SageMaker HPO
  - Spot Training
  - SageMaker Debugger
- Q&A
- Survey

aws machine learning

Support the responsible use of ML throughout the model lifecycle

Build

Perform bias analysis during exploratory data analysis

Train

Conduct bias and explainability analysis after training

Deploy

Explain individual inferences from models in production

Monitor

Validate bias and relative feature importance over time

SageMaker Building ML models

aws machine learning

Build ML models

Fully managed shareable notebooks on Amazon EC2

Fully managed, sharable Jupyter notebooks
Run notebooks on elastic compute resources

Built-in algorithms
15 built-in algorithms available in prebuilt container images

Prebuilt solutions and open-source models
Over 150 popular open-source models

AutoML
Automatically create ML models with full visibility

Support for major frameworks and toolkits
Optimized for popular deep learning (DL) frameworks such as TensorFlow, PyTorch, Apache MXNet, and Hugging Face

aws machine learning

Amazon SageMaker has built-in algorithms or bring your own

Computer vision
Image classification | Object detection | Semantic segmentation

Topic modeling
LDA | NTM

Classification
Linear Learner | XGBoost | KNN

Recommendation
Factorization machines

Forecasting
DeepAR

Working with text
BlazingText | Supervised | Unsupervised

Regression
Linear Learner | XGBoost | KNN

Clustering
KMeans

Sequence translation
Seq2Seq

Anomaly detection
Random cut forests | IP Insights

Feature reduction
PCA

SageMaker Training Deep Dive

aws machine learning

Train ML models

Fast and cost-effective ML model training

Experiment management and model tuning
Save weeks of effort by automatically tracking training runs and tuning hyperparameters

Debug and profile training runs
Use real-time metrics to correct performance problems

Distributed training
Complete distributed training up to 40% faster

Training compiler
Accelerate training times by up to 50% through more efficient use of GPUs

Managed spot training
Reduce the costs of training by up to 90%

aws machine learning

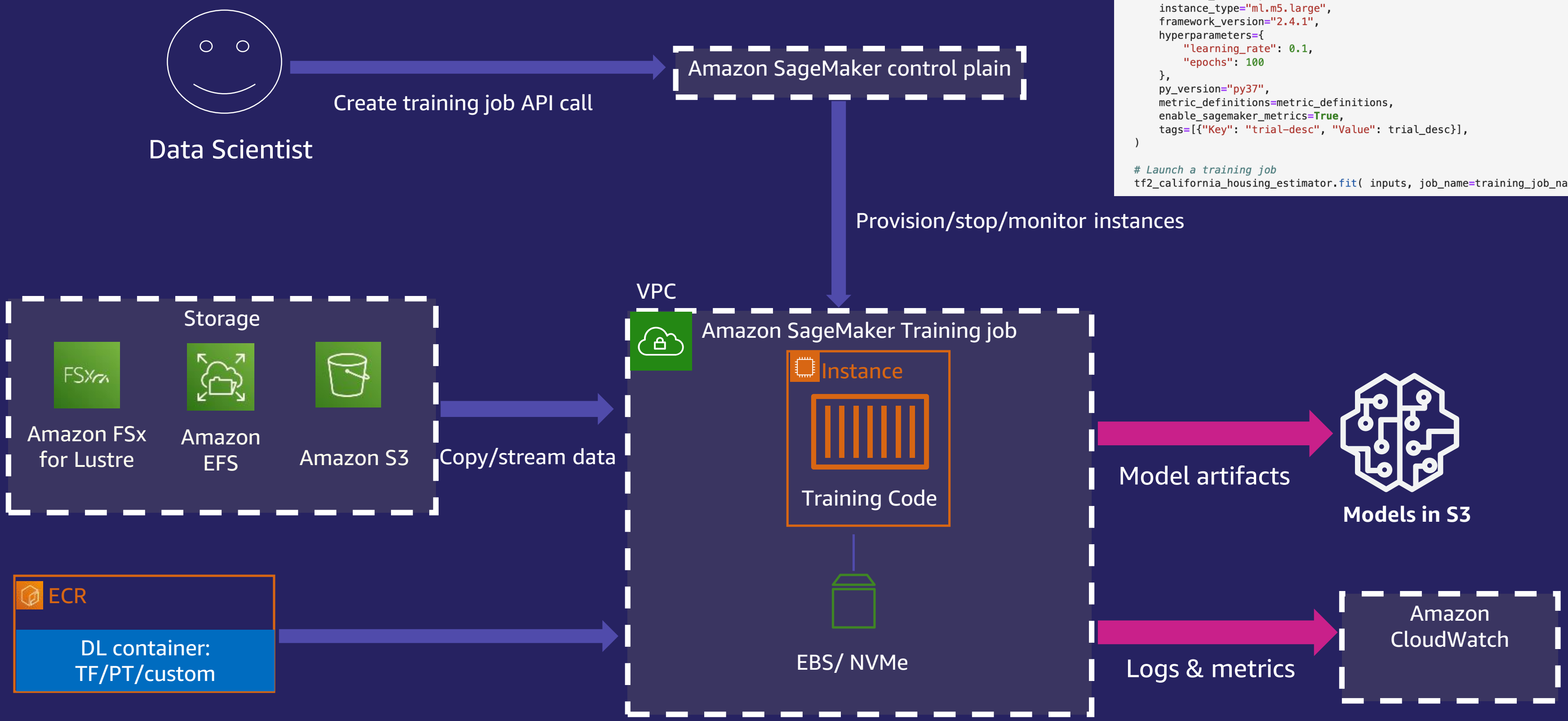
Training on Amazon SageMaker

```python
# Input data from s3
inputs = {"train": s3_inputs_train, "test": s3_inputs_test}

metric_definitions = [
    {"Name": "loss", "Regex": "loss: ([0-9\\.]+)"},
    {"Name": "accuracy", "Regex": "accuracy: ([0-9\\.]+)"},
    {"Name": "val_loss", "Regex": "val_loss: ([0-9\\.]+)"},
    {"Name": "val_accuracy", "Regex": "val_accuracy: ([0-9\\.]+)"},
]

# Create a TensorFlow Estimator
tf2_california_housing_estimator = TensorFlow(
    entry_point="california_housing_tf2.py",
    source_dir="code",
    role=sagemaker.get_execution_role(),
    instance_count=1,
    instance_type="ml.m5.large",
    framework_version="2.4.1",
    hyperparameters={
        "learning_rate": 0.1,
        "epochs": 100
    },
    py_version="py37",
    metric_definitions=metric_definitions,
    enable_sagemaker_metrics=True,
    tags=[{"Key": "trial-desc", "Value": trial_desc}],
)

# Launch a training job
tf2_california_housing_estimator.fit( inputs, job_name=training_job_name)
```



Training Estimator

```python
# Input data from s3
inputs = {"train": s3_inputs_train, "test": s3_inputs_test}

metric_definitions = [
    {"Name": "loss", "Regex": "loss: ([0-9\\.]+)"},
    {"Name": "accuracy", "Regex": "accuracy: ([0-9\\.]+)"},
    {"Name": "val_loss", "Regex": "val_loss: ([0-9\\.]+)"},
    {"Name": "val_accuracy", "Regex": "val_accuracy: ([0-9\\.]+)"},
]

# Create a TensorFlow Estimator
tf2_california_housing_estimator = TensorFlow(
    entry_point="california_housing_tf2.py",
    source_dir="code",
    role=sagemaker.get_execution_role(),
    instance_count=1,
    instance_type="ml.m5.large",
    framework_version="2.4.1",
    hyperparameters={
        "learning_rate": 0.1,
        "epochs": 100
    },
    py_version="py37",
    metric_definitions=metric_definitions,
    enable_sagemaker_metrics=True,
    tags=[{"Key": "trial-desc", "Value": trial_desc}],
)

# Launch a training job
tf2_california_housing_estimator.fit( inputs, job_name=training_job_name)
```

aws machine learning

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

EXAMPLES

Decision trees
Tree depth | Max leaf nodes | Gamma | Eta | Lambda | Alpha

Neural networks
Number of layers | Hidden layer width | Learning rate | Embedding dimensions | Dropout

aws machine learning

Amazon SageMaker Automatic Model Tuning

Hyperparameter Tuning

Setting up hyper parameter tuning job

1. Pick hyperparameters and ranges

```
hyperparameter_ranges = {'eta': ContinuousParameter(0, 1),
                         'min_child_weight': ContinuousParameter(1, 10),
                         'alpha': ContinuousParameter(0, 2),
                         'max_depth': IntegerParameter(1, 10)}
```

2. Pick objective metric

```
objective_metric_name = 'validation:auc'
```

3. Pick job parameters

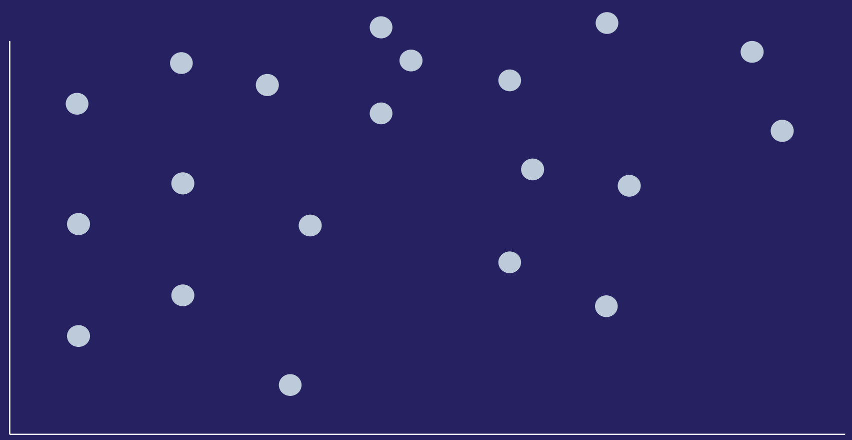
```
tuner = HyperparameterTuner(xgb,
                            objective_metric_name,
                            hyperparameter_ranges,
                            max_jobs=20,
                            max_parallel_jobs=3)
```

aws machine learning

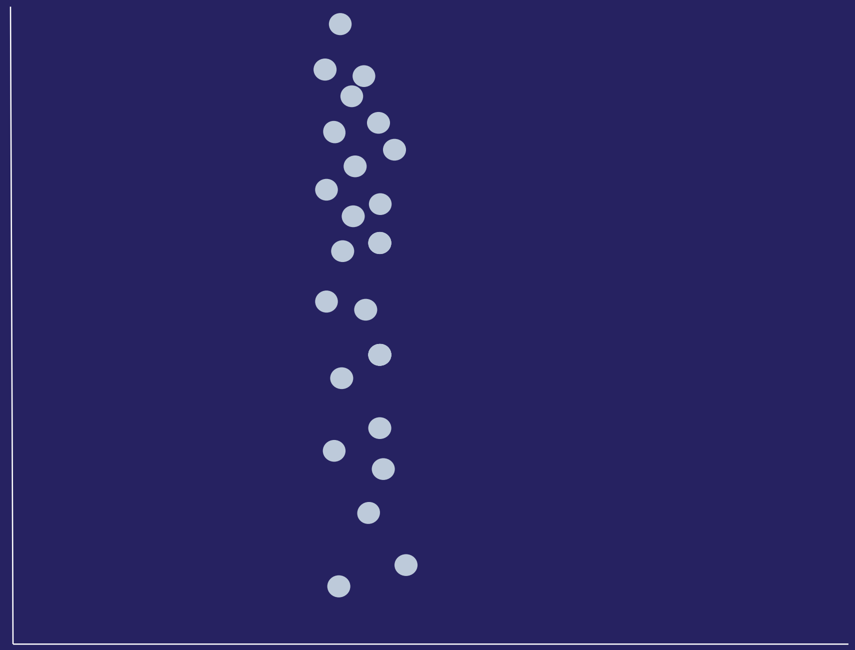
Amazon SageMaker Automatic Model Tuning

What if I need all my jobs tuned at the same time ?

Bayesian Search



Random Search



aws machine learning

Random Search

```
{
    "ParameterRanges": {...}
    "Strategy": "Random",
    "HyperParameterTuningJobObjective": {...}
}
```

```
tuner = HyperparameterTuner(
    sagemaker_estimator,
    objective_metric_name,
    hyperparameter_ranges,
    max_jobs=20,
    max_parallel_jobs=20,
    strategy="Random"
)
```

aws machine learning

Amazon SageMaker Automatic Model Tuning

Can I use hyperparameter tuning with my own model ?

1 Built-in Algorithms

2 Docker

3 Script Mode

Fully Customizable

aws machine learning

Amazon SageMaker Automatic Model Tuning

Can I use hyperparameter tuning with my own model ?

Setting the hyperparameters

In [5]:
```
hyperparameters = dict(batch_size=32, data_augmentation=True, learning_rate=.0001,
                       width_shift_range=.1, height_shift_range=.1, epochs=1)
hyperparameters
```

Out[5]:
```
{'batch_size': 32,
 'data_augmentation': True,
 'learning_rate': 0.0001,
 'width_shift_range': 0.1,
 'height_shift_range': 0.1,
 'epochs': 1}
```

Docker ✓

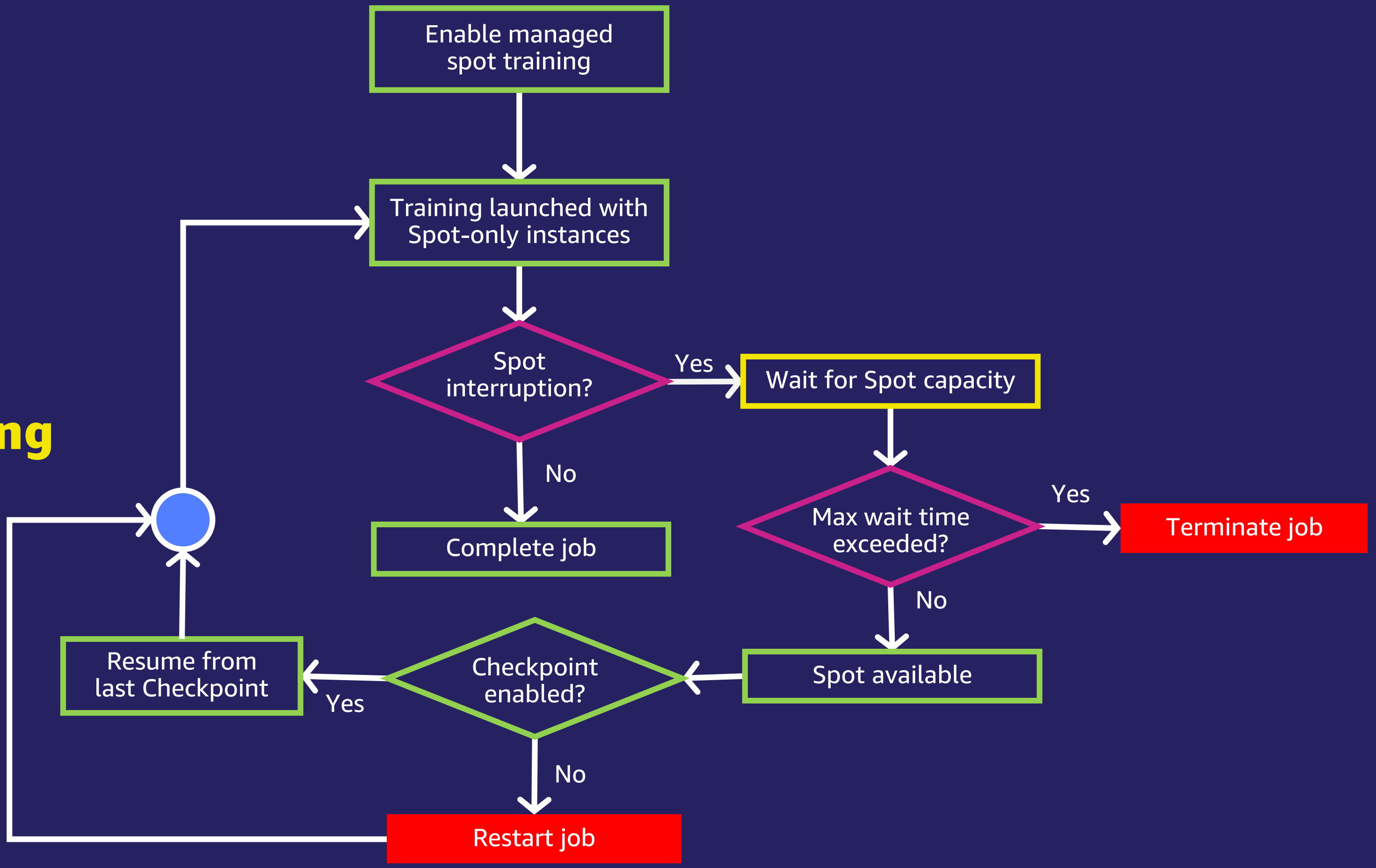
Script Mode ✓

aws machine learning

Managed Spot Training

Managed Spot Training

Save up to 90% on model training costs



Key considerations

Training with only Spot

Interrupted jobs resume if checkpointed and if Spot instances become available; Jobs restart if not checkpointed*

Works with Automated Model Tuning

Training jobs can run only with a single instance - type in a single - AZ

Does not integrate with Spot Fleet and Spot Block today

Checkpoint

Built-in algorithms checkpoint automatically

For custom models, checkpointing should be enabled

Checkpoints are saved to S3

Models that don't checkpoint are subject to *MaxWaitTime* of 60 mins

Pricing

View savings on AWS console or use DescribeTrainingJob API

Charged for the run duration before completion or termination; not charged for idle time, billing starts when instances are ready

Charged for data download time only once even if the job is interrupted multiple times

* Checkpointing is a best practice and is highly recommended

aws machine learning

Debugger

Generate ML models faster

Detect bottlenecks and issues during training in real-time and correct problems to deploy models faster, with a single, unified tool

Optimize resources with no additional code

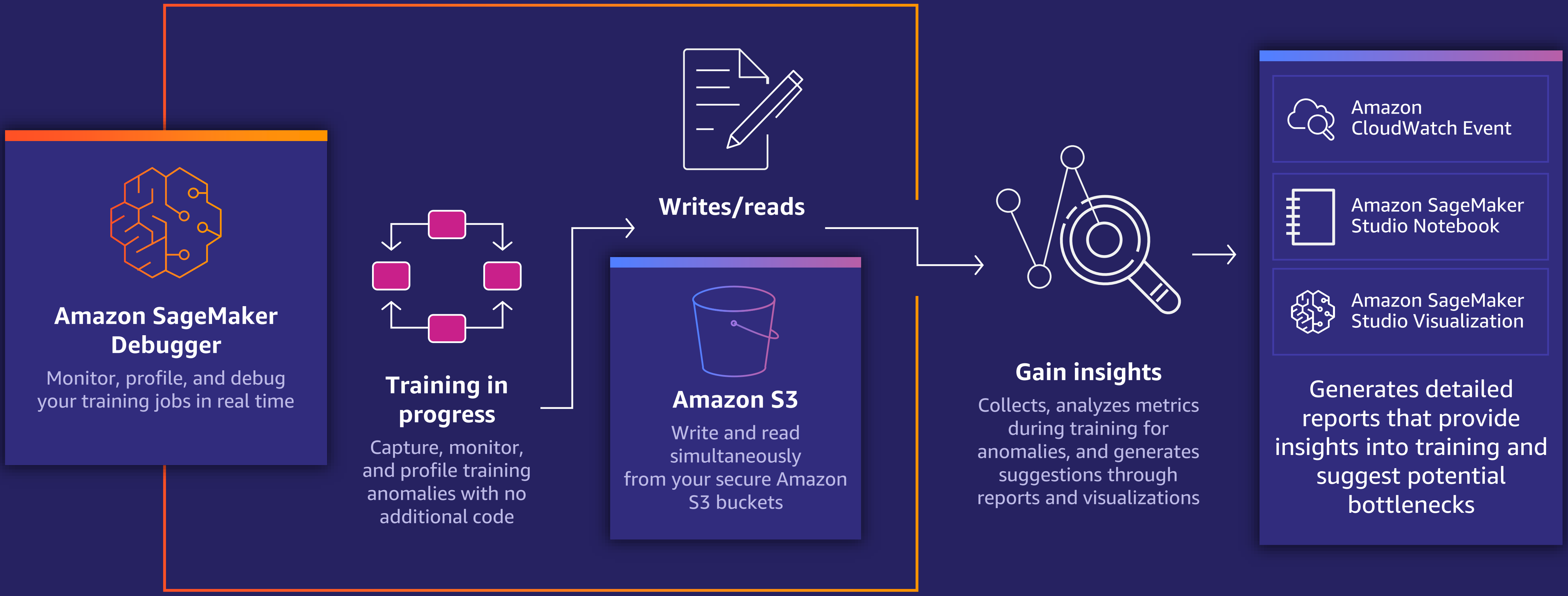
Monitor and profile system resources without code and get recommendations to optimize resources effectively

Make ML training transparent

Get complete insights into the ML training process in real-time and offline

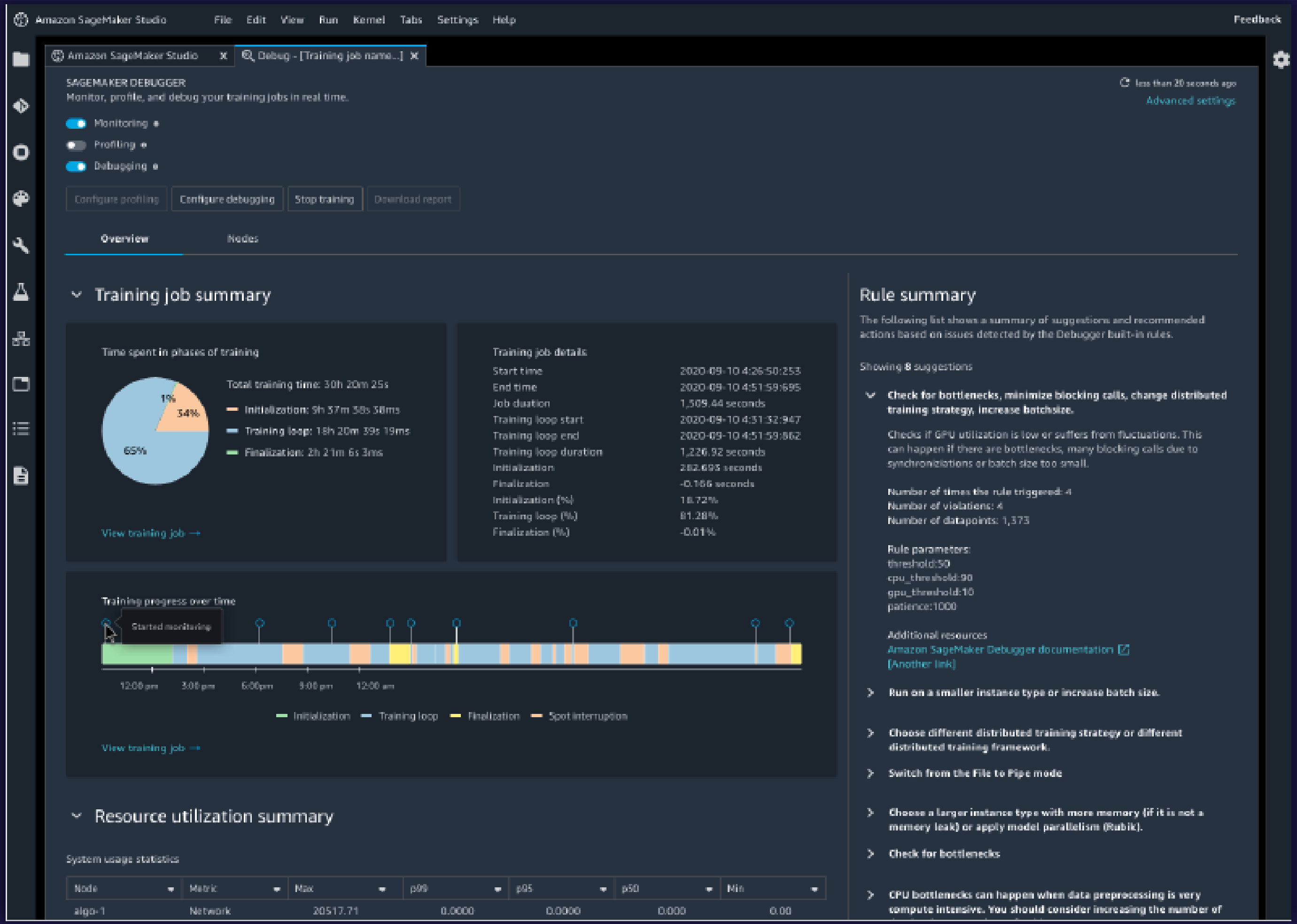
aws machine learning

Amazon SageMaker Debugger—How it works



aws machine learning

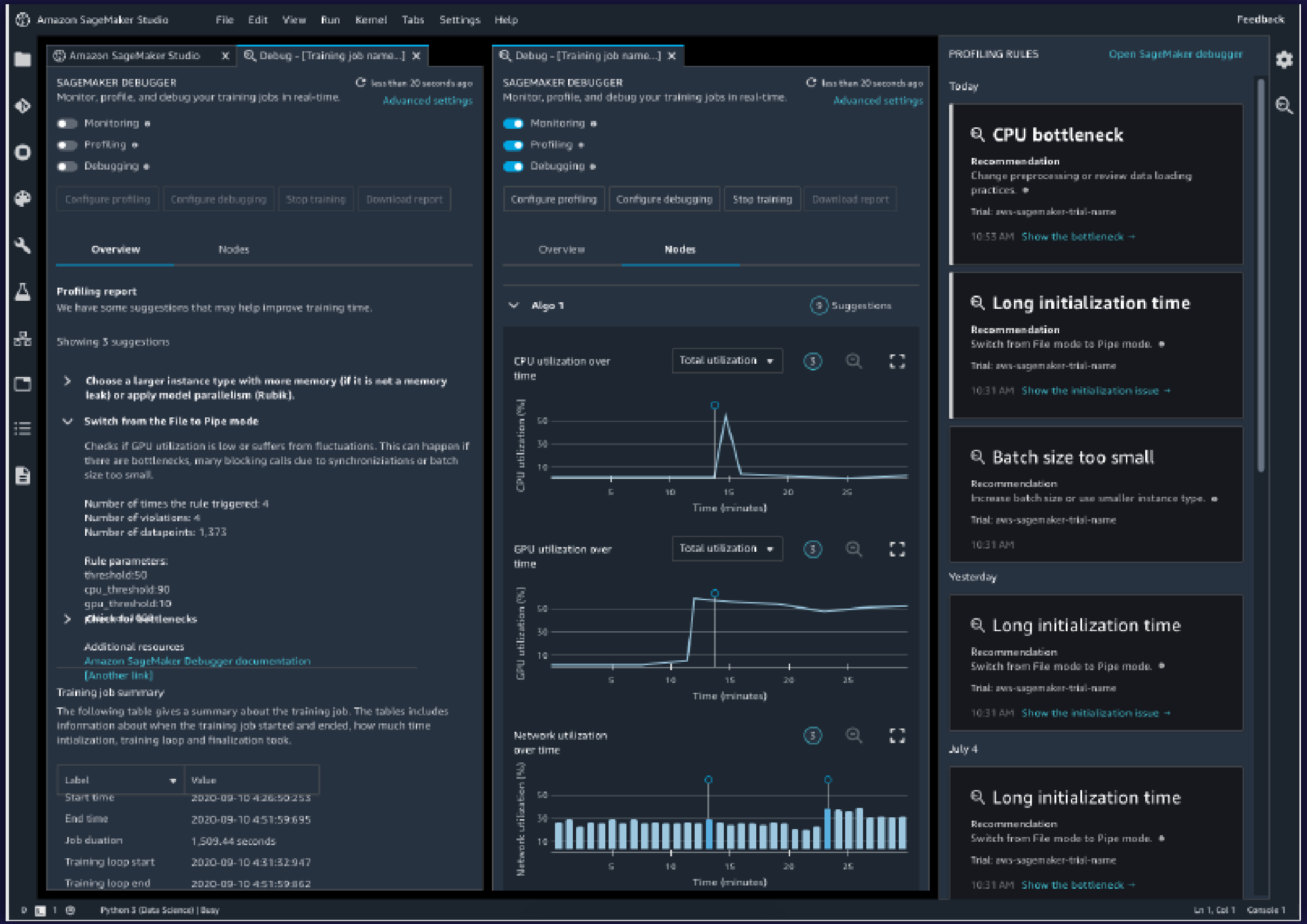
Monitor and profile system resource utilization



- Automatically monitor system resource utilization
- Profile training jobs to collect ML framework metrics
- Visualize system resource utilization for GPU, CPU, network, memory within SageMaker Studio

aws machine learning

Analyze errors and take action



- Built-in analysis in the form of rules
- Automatically analyze training data including inputs, outputs, tensors
- Detect if a model is overfitting or overtraining, or determine if gradient values are incorrect
- Specify custom actions to stop training or send alerts

aws machine learning

Broad support across algorithms and frameworks

1 **Supports** popular ML algorithms such as XGBoost and deep learning frameworks such as TensorFlow, PyTorch, Apache MXNet, and Keras, with SageMaker built-in containers

2 **Integrated** with AWS Lambda to act on results from alerts

3 **Invoke actions** to automatically stop a training job when you detect a non-converging action such as losses increasing continuously

aws machine learning

Amazon SageMaker

Next Steps

Onboarding & Processing

- https://docs.aws.amazon.com/sagemaker/latest/dg/gs-studio-onboard.html
- https://docs.aws.amazon.com/sagemaker/latest/dg/processing-job.html

Training

- https://docs.aws.amazon.com/sagemaker/latest/dg/train-model.html
- https://docs.aws.amazon.com/sagemaker/latest/dg/distributed-training.html
- https://aws.amazon.com/sagemaker/debugger

Deployment

- https://docs.aws.amazon.com/sagemaker/latest/dg/realtime-endpoints.html
- https://docs.aws.amazon.com/sagemaker/latest/dg/serverless-endpoints.html
- https://docs.aws.amazon.com/sagemaker/latest/dg/async-inference.html
- https://docs.aws.amazon.com/sagemaker/latest/dg/batch-transform.html

https://github.com/aws/amazon-sagemaker-examples

https://sagemaker.readthedocs.io/en/stable/index.html

aws machine learning

Q & A

Please Complete the session Survey

Thank you!