Introduction to Training Jobs

At this point, we are interested in creating a **Training Job**. A training job does not only create a single model using machine learning but will define repeatable steps to help us automated the process.

A training job includes the following information:

- The location of **training data** to use in S3
- Which **compute resources** to use for model training
- The location of the **training code** in ECS
- The location of where to put **job output** artifacts in S3
- **Metadata** (Including Version # and Tags)

Ways to Launch a Training Job

- **AWS Console** - For debugging and initial set-up
- **CLI (Command Line Interface) -** For one-off jobs.
- **AWS SDK** (**Source Development Kit)** - For programmatic access

These are general guidelines, all of these methods are very flexible.

Distributed Training

Distributed training is the practice of splitting the computational load of a training job over multiple compute resources. This can occur either when the data is significantly large or when an algorithm is significantly complex. Splitting raw data intake across multiple compute resources is called **data parallel training**, and splitting algorithmic computations across multiple compute resources is called **model parallel training**.

Script Mode

Script mode allows you to create execute a custom script for your training jobs using popular machine learning frameworks like Scikit-learn. This mode allows you to specify custom libraries, run a custom script, and define a model on your own terms while still providing a stable managed container that you can rely on to execute your work.

You can also construct a custom image (copies of a computing system) that meets your specification and upload it to **ECR (Elastic Container Registry),** and invoke that **container** (a consolidated package of a computing runtime environment) through SageMaker. AWS will still handle, stably, the underlying OS and hardware.

Model Artifacts

Regardless of how you launch a training job, the result will be a **model artifact**. This model artifact will be stored in S3, and will consist both of a set of trained parameters and a set of instructions for inference.

If you already have a trained model outside of AWS and you wish to incorporate it into the SageMaker environment, you can do so through the **CreateModel API (Application Programming Interface)**. The result of calling this API is also a model artifact.

This model artifact is a required parameter for an **endpoint**, which is how AWS deploys a model to production.

New Terms

- **Training Data** - Pre-processed examples (along with labels for supervised learning) to use to train the machine learning model
- **Compute Resources** - The hosted instance types and counts to be used for the duration of the training job. Larger instance types or greater counts may increase the speed of training at an increasing marginal cost.
- **Training Code** - The Docker container which holds the environment, libraries and executable code to be deployed on the selected compute resources. This code will incorporate all

training steps, allowing only a few parameters to be injected at runtime.

- **Job Output** - This is the trained model artifact that is the product of running back-propagation over the examples in the training data. This artifact can be hosted to request inferences on new examples.
- **Distributed Training -** The practice of splitting the computational load of a training job over multiple compute resources.
- **Data Parallel Training -** Splitting raw data intake across multiple compute resources
- **Model Parallel Training -** Splitting algorithmic computations across multiple compute resources
- **Script Mode** - Executions of custom training algorithms within SageMaker.
- **Model Artifact** - A set of trained parameters and a set of instructions for inference stored in S3.
- **Image** - copies of a computing system
- **Container** - A consolidated package of a computing runtime environment
- **S3** - Simple Storage Service
- **ECR** - Elastic Container Registry
- **CLI** - Command Line Interface
- **SDK** - Source Development Kit
- **API** - Application Programming Interface

Additional Resources

- If you want to learn more about distributed training, we recommend you to read Amazon SageMaker Distributed Training Libraries.
- If you want to learn more about script mode, we recommend Bring your own model with Amazon SageMaker script mode
- If you want to learn more about ECR, we recommend What is Amazon Elastic Container Registry?