Machine Learning Made Easy on Kubernetes: MLOps

Brian Redmond, Azure
Brian Redmond – Who am I?

- Cloud Architect @ Microsoft (18 years)
- Azure Global Black Belt Team
- Live in Pittsburgh, PA, USA
- Avid marathon runner and outdoors enthusiast
- World traveler
Let’s start with a demo!

“Real world” application with Game of Thrones

What Game of Thrones character is this?
• Let’s use Tensorflow and build a image classifier

http://aka.ms/thrones
What is Machine Learning?
“The ability to learn without being explicitly programmed,” Arthur Samuel
Machine Learning is super powerful

OpenAI created “MuseNet” which can generate 4-minute musical compositions with 10 different instruments (2019)

Microsoft reaches 'human parity' with new speech recognition system (2016)

Delphi self driving car drove cross country with a computer handling 99% of the drive (2015)

Google DeepMind’s AlphaGo program beat a world class Go player in multiple games (2016)
But Machine Learning is hard!
Kubernetes
What’s behind the growth?
Kubernetes: the leading orchestrator shaping the future app development and management

42% portability
45% scalability
50% agility

The perceived benefits of Kubernetes

Source: Cloud Native Computing Foundation
Cloud Native Apps
Cloud Native ML?
Building a model
Platform

Data ingestion -> Data analysis -> Data transformation -> Data validation -> Data splitting

Trainer -> Building a model -> Model validation -> Training at scale

Roll-out -> Serving -> Monitoring -> Logging
Introducing Kubeflow
Make it Easy for Everyone to **Develop, Deploy and Manage** Portable, Distributed ML on Kubernetes
Cloud Training Experimentation

Kubeflow

Kubeflow

Kubeflow

Kubeflow
Cloud Native ML!
Kubeflow - What’s in the box?

Key Features

• Ingest your data with Pachyderm
• Services for spawning and managing Jupyter notebooks
• Operators for Tensorflow, PyTorch, MXNet, Chainer jobs
• Serving
  • TF Serving
  • Seldon
  • TensorRT (NVidia)
• Apache Beam (batch and stream processing)
• Katib (hyperparameter tuning)
• Kubebench (benchmarking)
• Pipelines
• Argo CD (GitOps)
Rich Container Based Pipelines

User Goal = Repeatable, multi-stage ML training

Problem
- Tools not built to be containerized/orchestrated
- Coordinating between steps often requires writing custom code
- Different tools have different infra requirements
Kubeflow Pipeline Details

• Containerized implementations of ML Tasks
  • Escapsulates all the dependencies of a step with no conflicts
  • Step can be singular or distributed
  • Can also involve external services

• Specified via Python SDK

• Inputs/outputs/parameters can be chained together
Ingestion
TF.Transform

Ingestion
TF.Transform

Training
TF.Job

Serving
TF.Serving

ingestStep = dsl.ContainerOp(image=tft_image, <params>,
file_outputs={'bucket': '/output.txt'})

trainStep = dsl.ContainerOp(image=tfjob_image, <params>,
arguments=[ingestStep.outputs['bucket']])

servingStep = dsl.ContainerOp(image=tfs_image, <params>,
arguments=[convertStep.outputs['bucket']])
What Game of Thrones character is this?

- Let’s use Tensorflow!
- Build a model based on Inception image classification
- Train using TFJob in Kubernetes
- Tensorflow Serving
- Hyperparameter Optimization
- Pipeline workflow
- Jupyter Notebook

http://aka.ms/thrones
Questions?

Find me at @chzbrgr71

Source for demos: https://github.com/chzbrgr71/got-image-classification