IBM Developer Model Asset eXchange

Nick Pentreath
Principal Engineer

@MLnick

#SAISDL6

About

@MLnick on Twitter & Github

Principal Engineer, IBM

CODAIT - Center for Open-Source Data & AI Technologies

Machine Learning & AI

Apache Spark committer & PMC

Author of *Machine Learning with Spark*

Various conferences & meetups

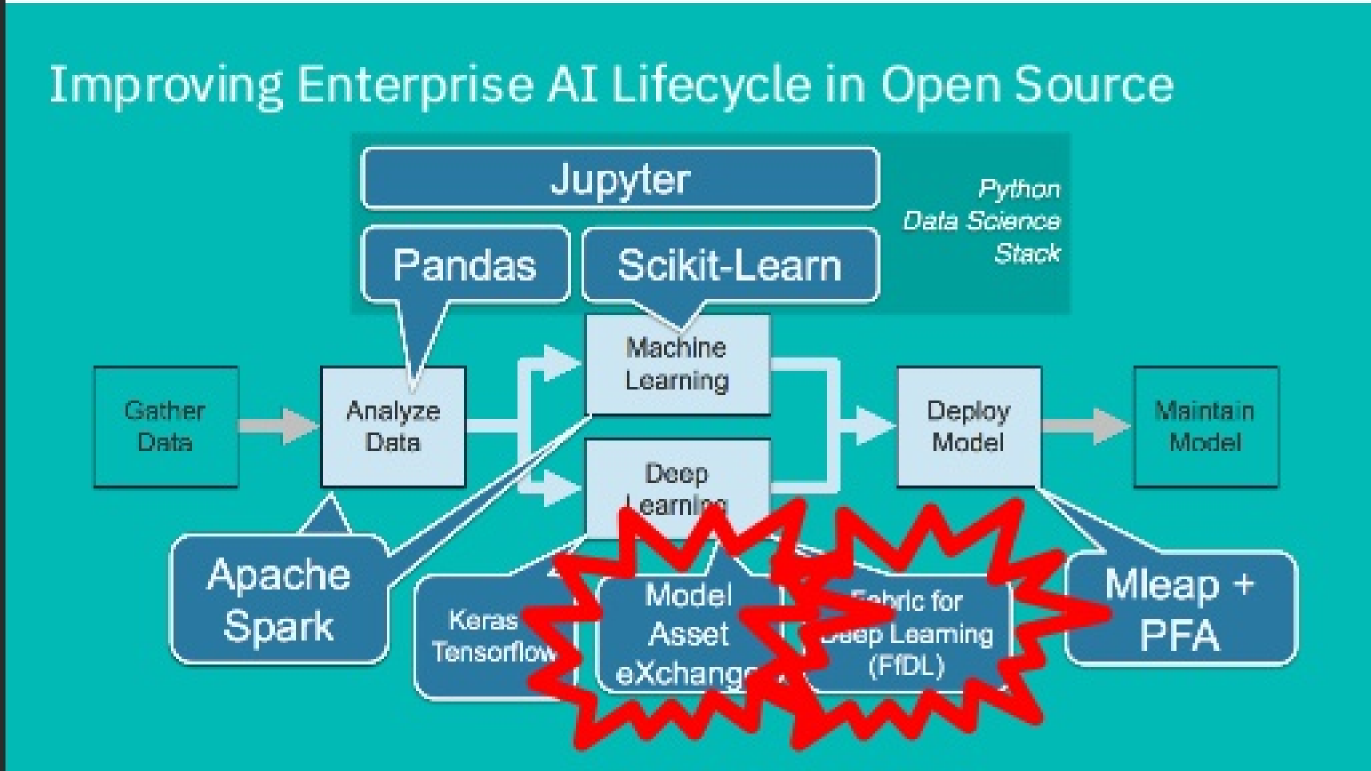
Center for Open Source Data and AI Technologies

CODAIT

codait.org

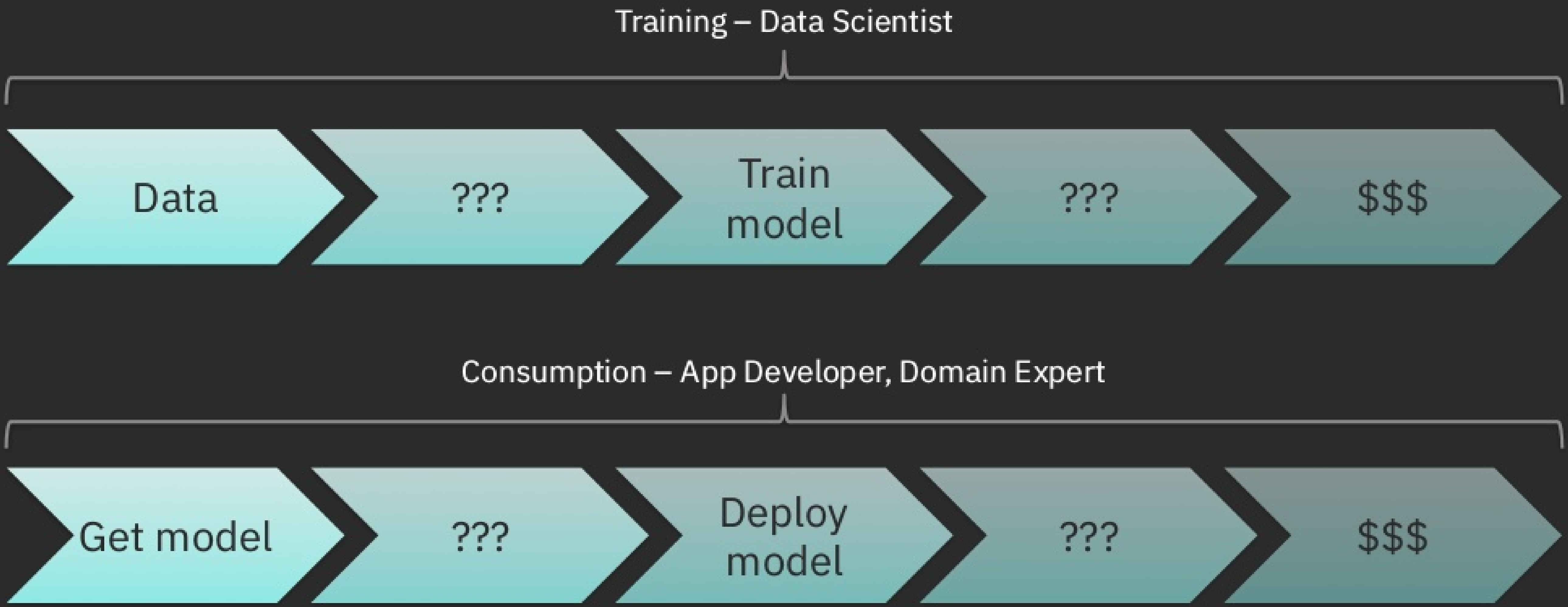
CODAIT aims to make AI solutions dramatically easier to create, deploy, and manage in the enterprise

Relaunch of the Spark Technology Center (STC) to reflect expanded mission



Improving Enterprise AI Lifecycle in Open Source

Applying Deep Learning: Perception



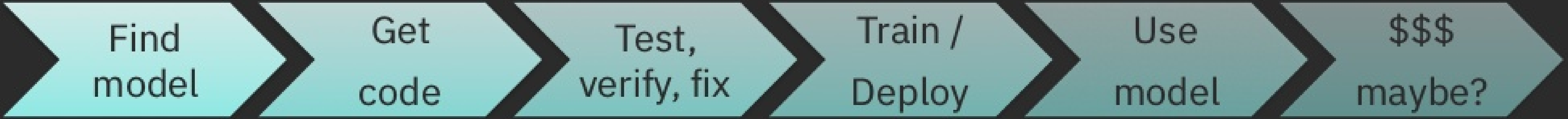
Training – Data Scientist

Data → ??? → Train model → ??? → $$$

Consumption – App Developer, Domain Expert

Get model → ??? → Deploy model → ??? → $$$

Applying Deep Learning: Reality

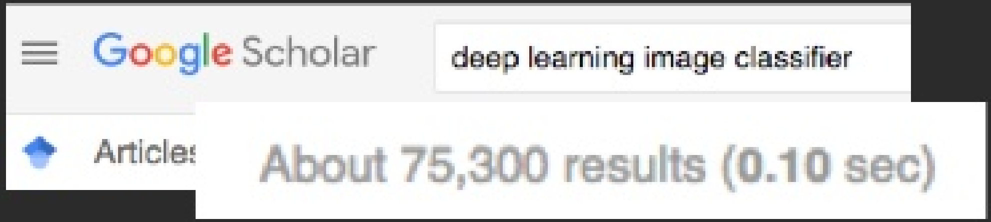


Step 1: Find a model

... that does what you need

... that is free to use

... that is performant enough



deep learning image classifier

About 75,300 results (0.10 sec)

Deep Residual Learning for Image Recognition

ImageNet Classification with Deep Convolutional Neural Networks

Very Deep Convolutional Networks for Large-Scale Image Recognition

Going deeper with convolutions

SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size

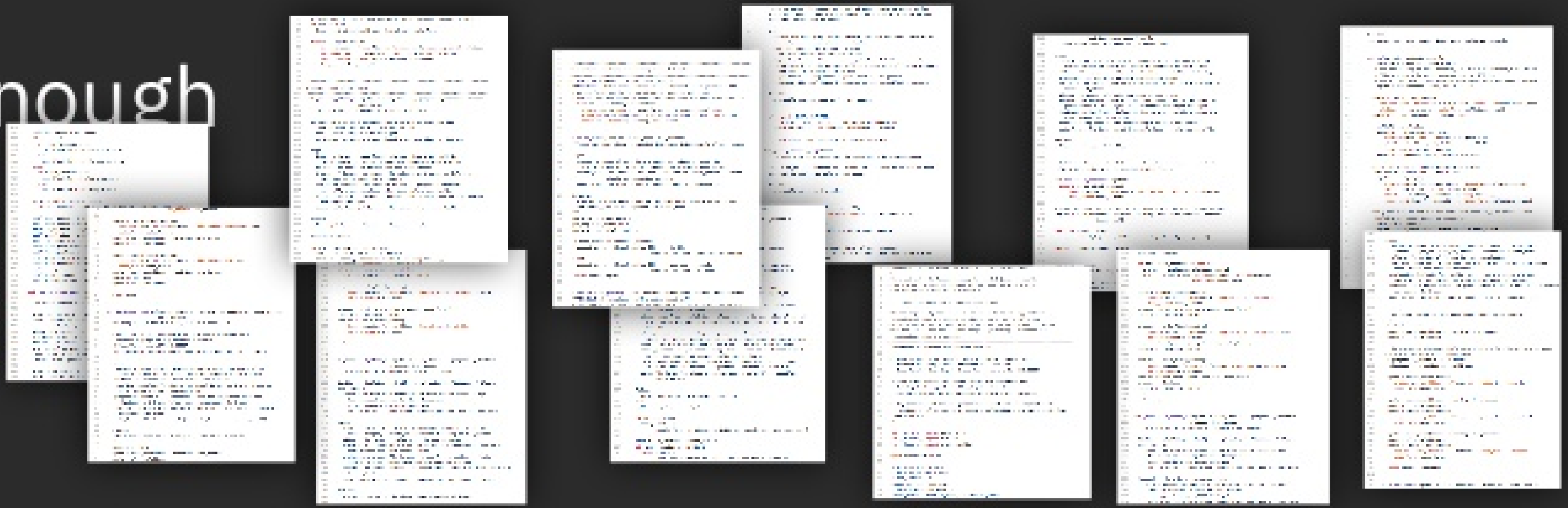
Step 2: Get the code

Is there a good implementation available?

… that does what you need

… that is free to use

… that is performant enough



TensorFlow code to build ResNet50 neural network graph

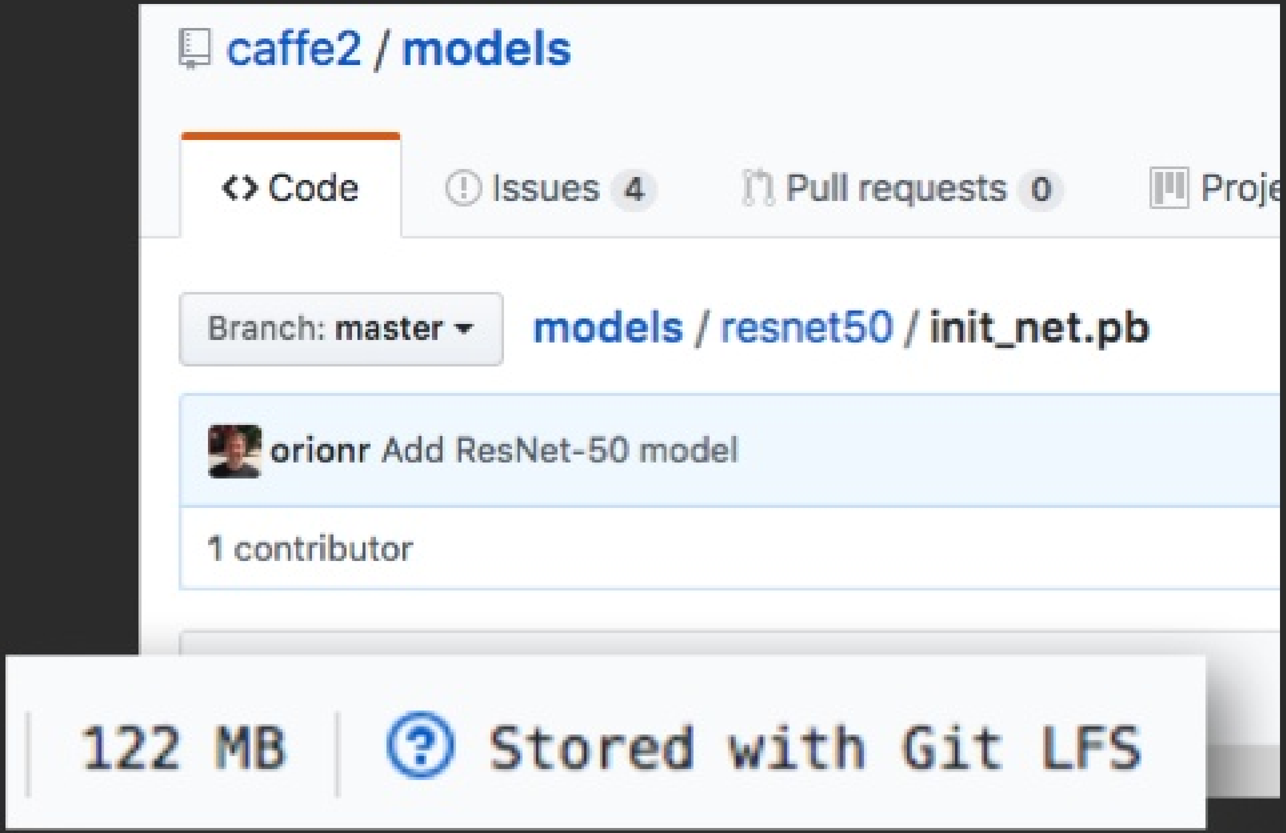
Or… Step 2: Get the pre-trained weights

Is there a good pre-trained model available?

… that does what you need

… that is free to use

… that is performant enough



caffe2 / models

Branch: master

models / resnet50 / init_net.pb

orionr Add ResNet-50 model

1 contributor

122 MB | Stored with Git LFS

Caffe2 ResNet50 model weights

Step 3: Verify the model you found

Check ...

... that it does what you need

... that it is free to use

... that it is performant enough

Step 4(a): Train the model

Step 4(a): Train the model

K
Keras
Caffe2
TensorFlow
DL4J
HOROVOD
ONNX
mxnet
Cognitive Toolkit
PaddlePaddle

** Logos trademarks of their respective projects*

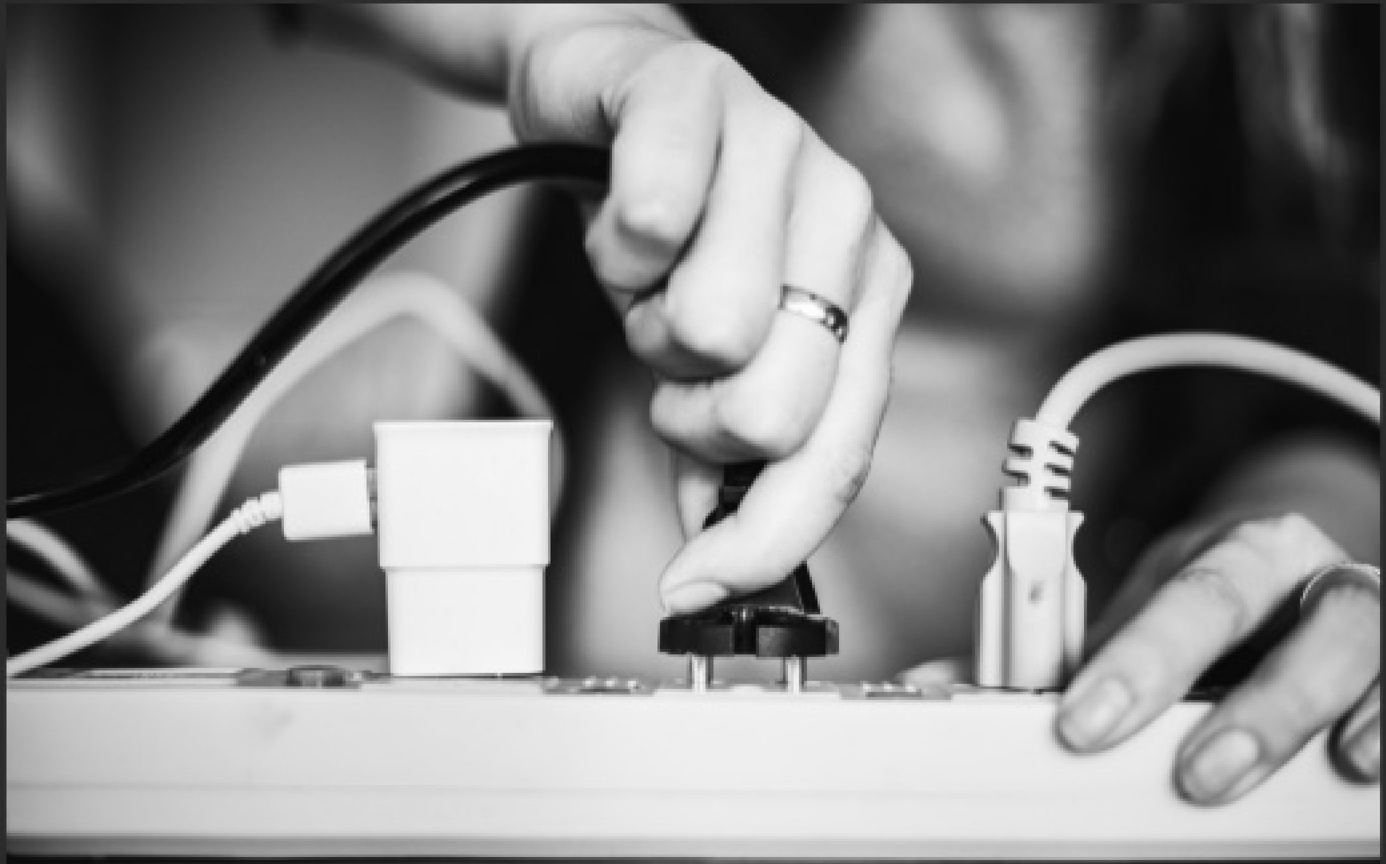
Step 4(b): Figure out how to deploy the model

... adjust inference code (or write from scratch)
... package your inference code, model code, and pre-trained weights together
... deploy your package

Step 5: Consume the model

… plug in to your application

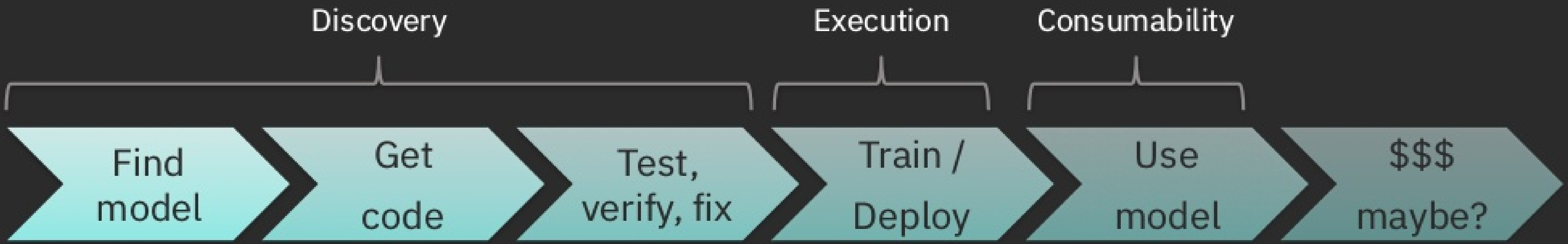
… which does not know
(or care) about tensors



Step 6: Profit

… hopefully

Applying Deep Learning: Reality



Discovery

Execution

Consumability

Find model

Get code

Test, verify, fix

Train / Deploy

Use model

$$$ maybe?

Model Zoos

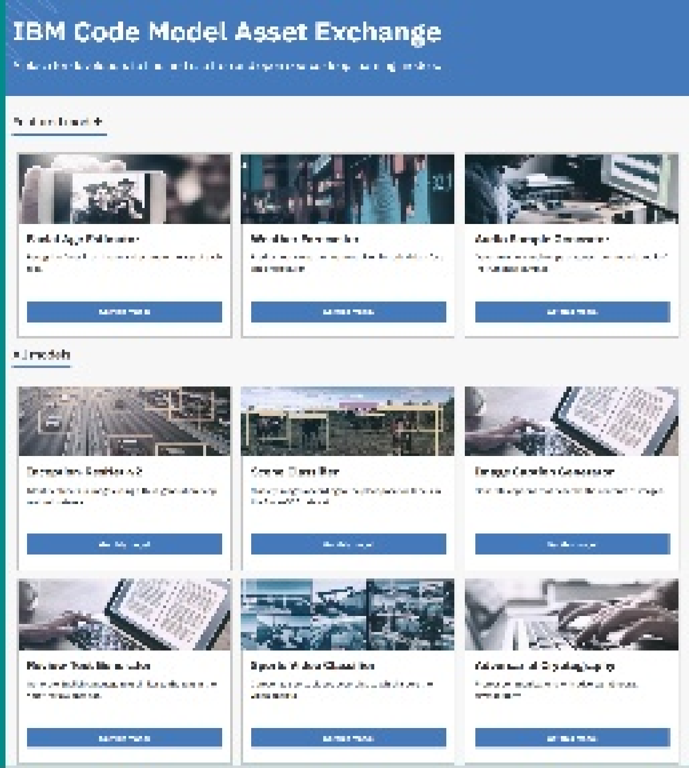
(in theory)

Model Zoos

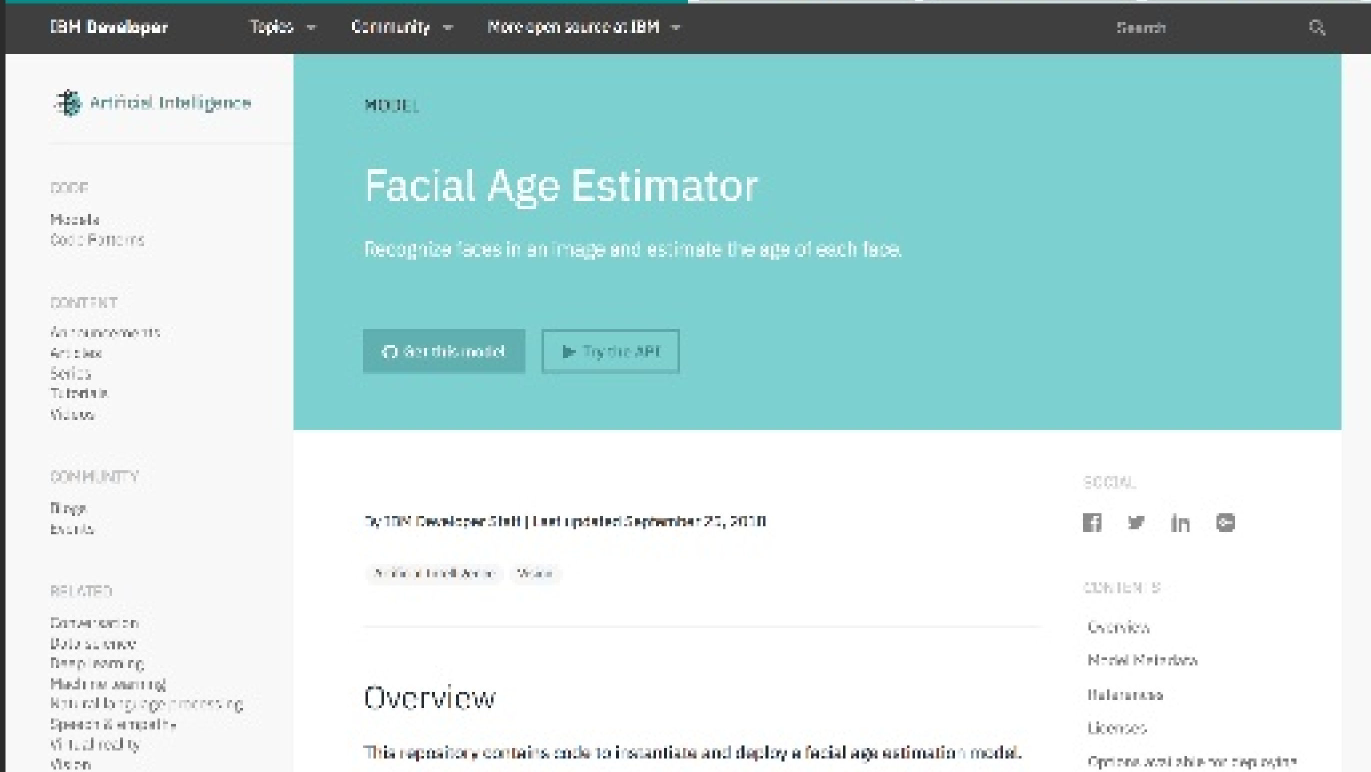
(in practice)

IBM Developer

http://ibm.biz/model-exchange



IBM Code Model Asset Exchange



Artificial Intelligence

CODE
- Models
- Code Patterns

CONTENT
- Announcements
- Articles
- Series
- Tutorials
- Videos

COMMUNITY
- Blogs
- Events

RELATED
- Conversation
- Data science
- Deep learning
- Machine learning
- Natural language processing
- Speech & empathy
- Virtual reality
- Vision

MODEL

Facial Age Estimator

Recognize faces in an image and estimate the age of each face.

By IBM Developer Staff | Last updated September 20, 2018

Artificial Intelligence · Vision

SOCIAL

CONTENTS
- Overview
- Model Metadata
- References
- Licenses
- Options available for deploying

Overview

This repository contains code to instantiate and deploy a facial age estimation model.

Fabric for Deep Learning

https://github.com/IBM/FfDL

FfDL provides a scalable, resilient, and fault tolerant deep-learning framework

- Fabric for Deep Learning or FfDL (pronounced as ‘fiddle’) is an open source project which aims at making Deep Learning easily accessible to the people it matters the most i.e. Data Scientists, and AI developers.
- FfDL provides a consistent way to deploy, train and visualize Deep Learning jobs across multiple frameworks like TensorFlow, Caffe, PyTorch, Keras etc.
- FfDL is being developed in close collaboration with IBM Research and IBM Watson. It forms the core of Watson`s Deep Learning service in open source.

FfDL

FfDL Github Page
https://github.com/IBM/FfDL

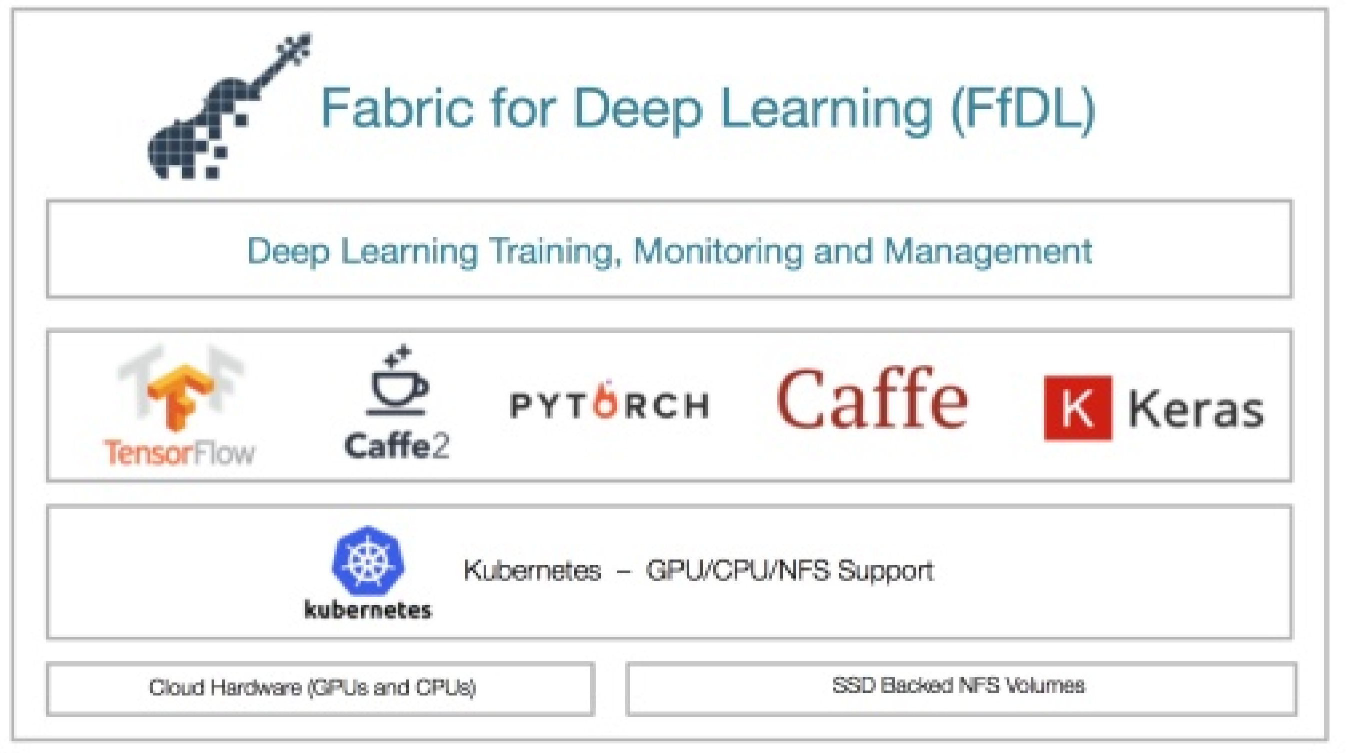
FfDL dwOpen Page
https://developer.ibm.com/code/open/projects/fabric-for-deep-learning-ffdl/

FfDL Announcement Blog
http://developer.ibm.com/code/2018/03/20/fabric-for-deep-learning

FfDL Technical Architecture Blog
http://developer.ibm.com/code/2018/03/20/democratize-ai-with-fabric-for-deep-learning

Deep Learning as a Service within Watson Studio
https://www.ibm.com/cloud/deep-learning

Research paper: “Scalable Multi-Framework Management of Deep Learning Training Jobs” http://learningsys.org/nips17/assets/papers/paper_29.pdf



Fabric for Deep Learning (FfDL)

Deep Learning Training, Monitoring and Management

TensorFlow | Caffe2 | PYTORCH | Caffe | Keras

kubernetes — Kubernetes – GPU/CPU/NFS Support

Cloud Hardware (GPUs and CPUs) | SSD Backed NFS Volumes

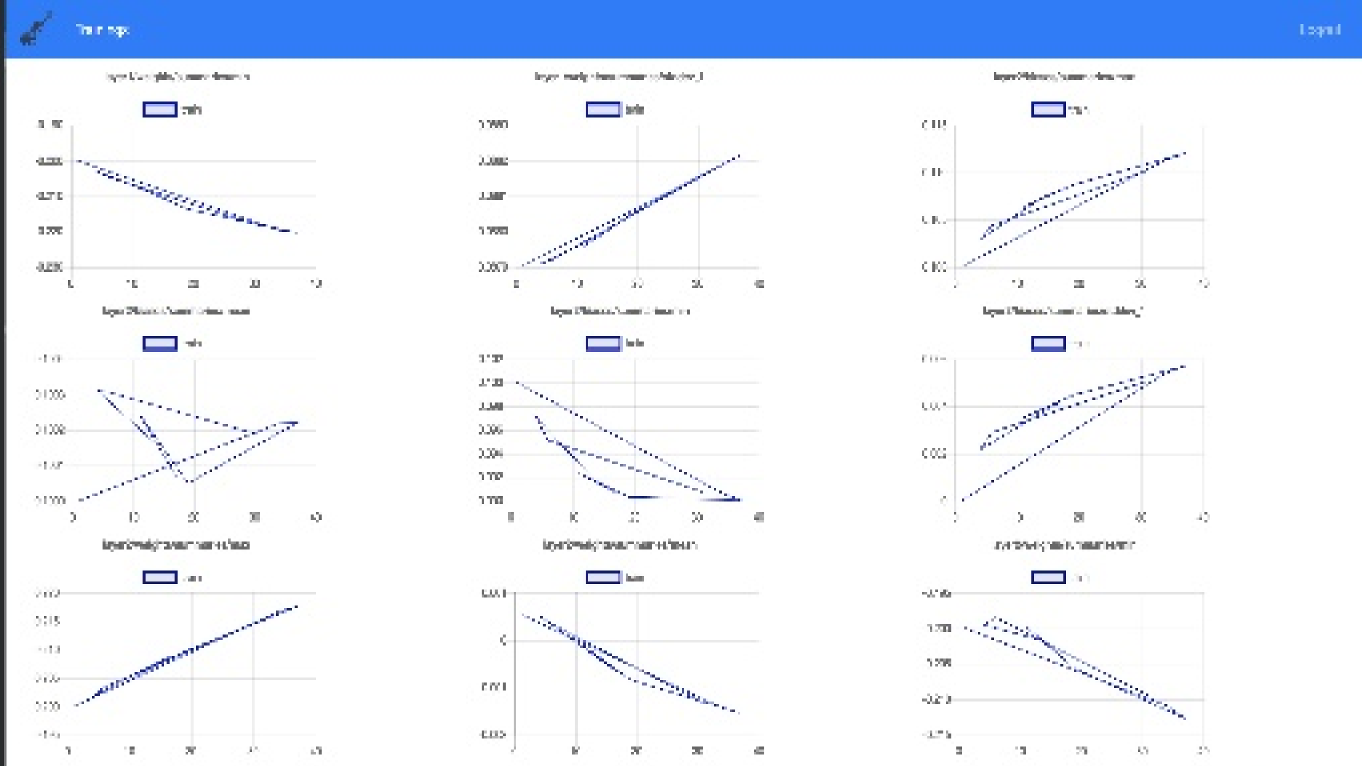
Fabric for Deep Learning

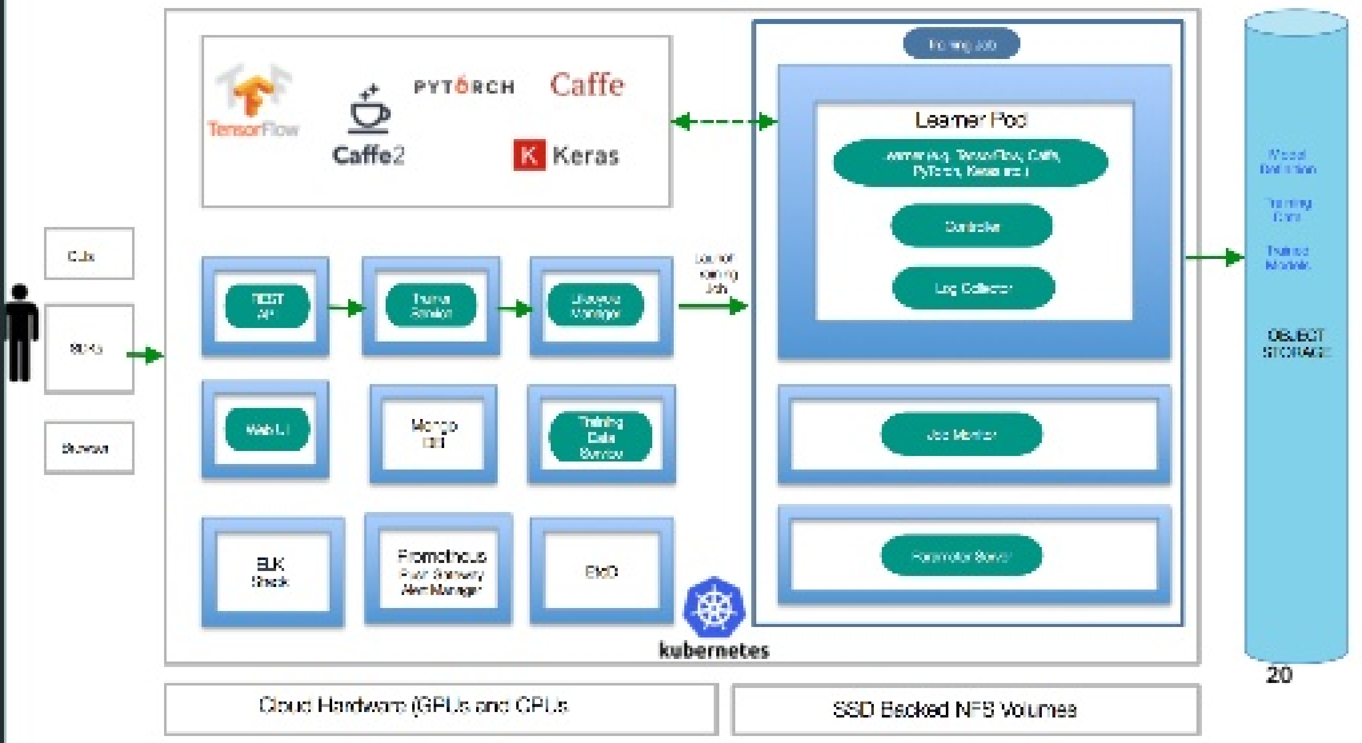
https://github.com/IBM/FfDL

FfDL is built using a microservices architecture on Kubernetes

- FfDL platform uses a microservices architecture to offer resilience, scalability, multi-tenancy, and security without modifying the deep learning frameworks, and with no or minimal changes to model code.
- FfDL control plane microservices are deployed as pods on Kubernetes to manage this cluster of GPU- and CPU-enabled machines effectively
- Tested Platforms: Minikube, IBM Cloud Public, IBM Cloud Private, GPUs using both Kubernetes feature gate Accelerators and NVidia device plugins

IBM CODE





Fabric for Deep Learning

https://github.com/IBM/FfDL

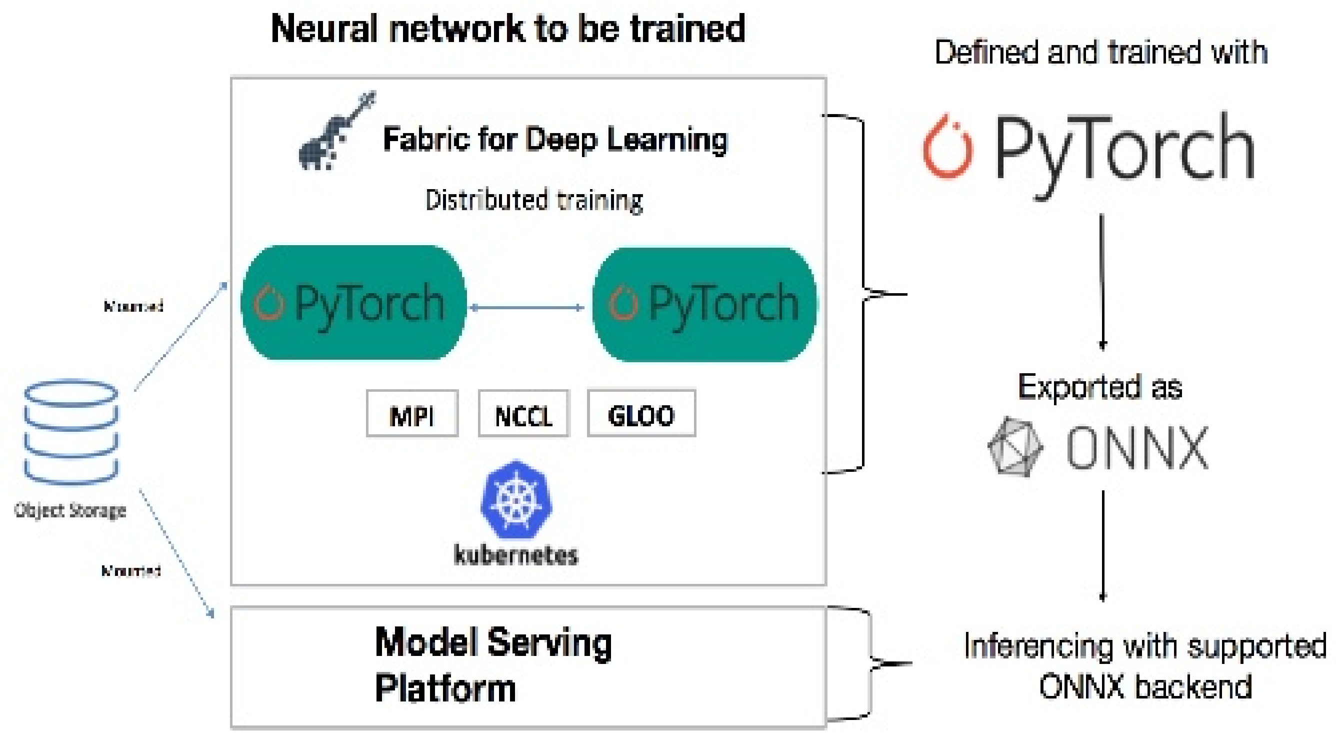
Just announced: Support for PyTorch 1.0 – including distributed training and ONNX!

Supports distributed training via Horovod

FfDL Github Page
https://github.com/IBM/FfDL

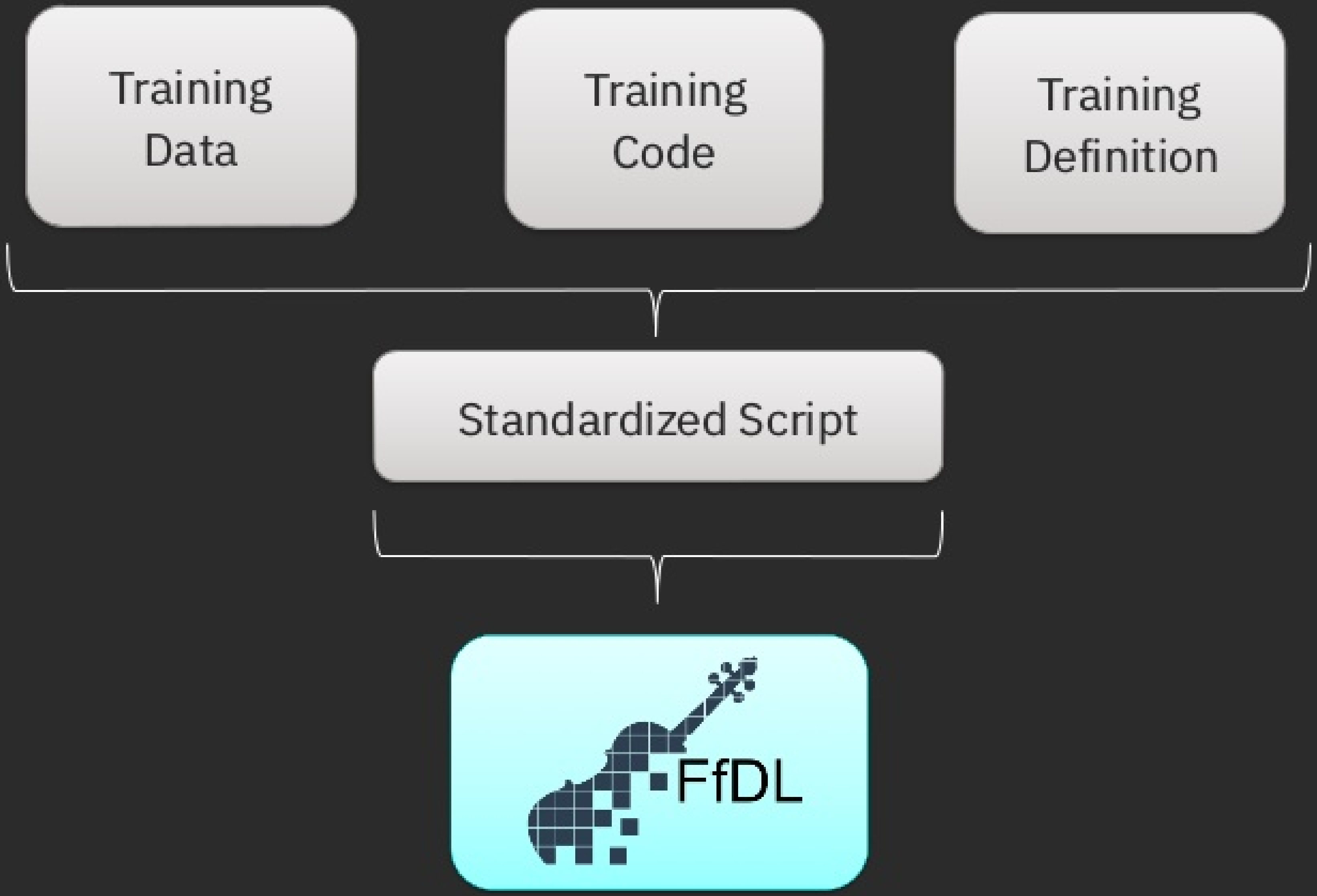
FfDL / PyTorch 1.0 Blog Post
https://developer.ibm.com/blogs/2018/10/01/announcing-pytorch-1-support-in-fabric-for-deep-learning/

FfDL / Horovod Blog Post
https://developer.ibm.com/code/2018/07/18/scalable-distributed-training-using-horovod-in-ffdl/



IBM CODE

Trainable Models



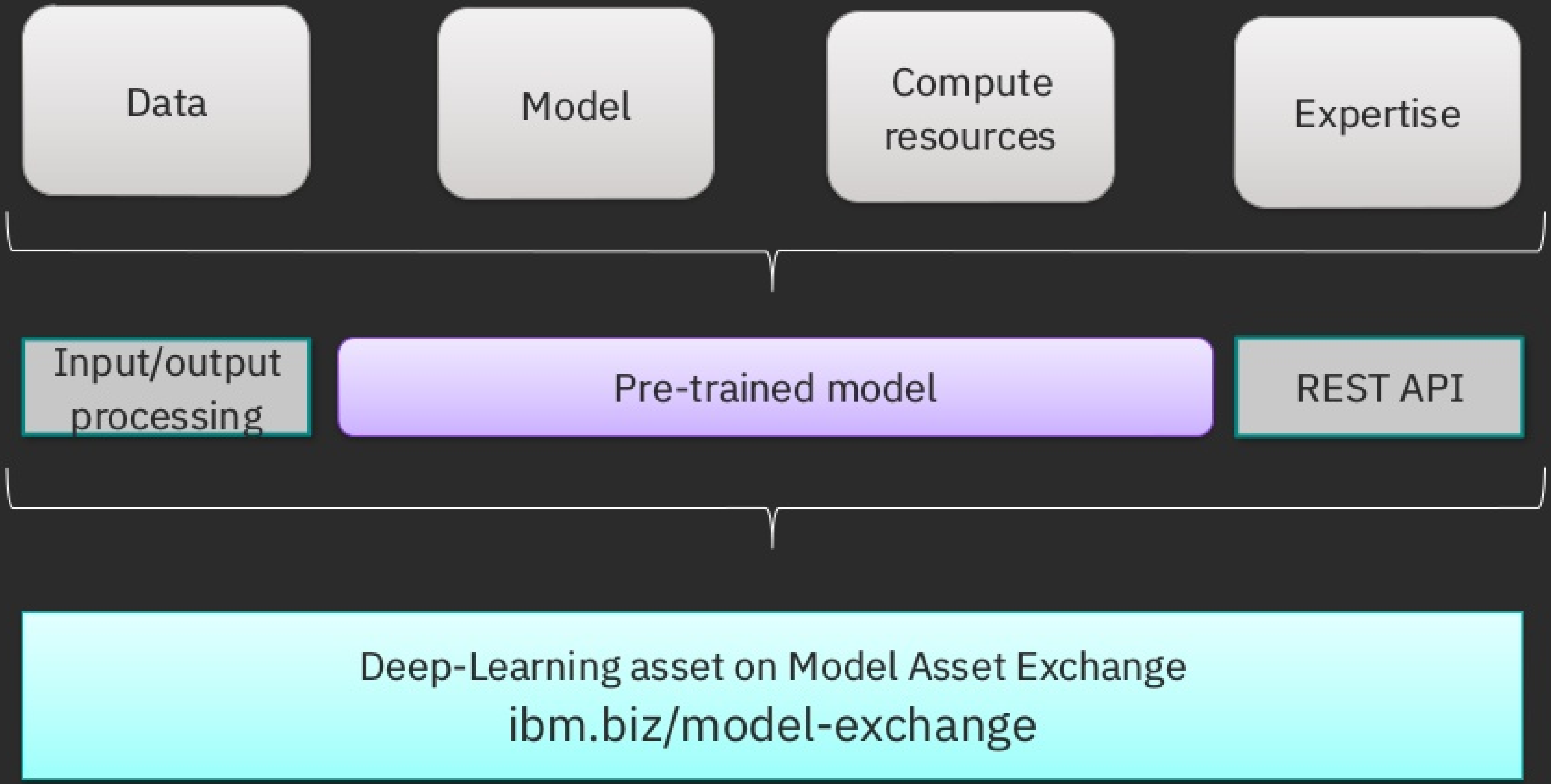
Training Data

Training Code

Training Definition

Standardized Script

Deployable Models



Data

Model

Compute resources

Expertise

Input/output processing

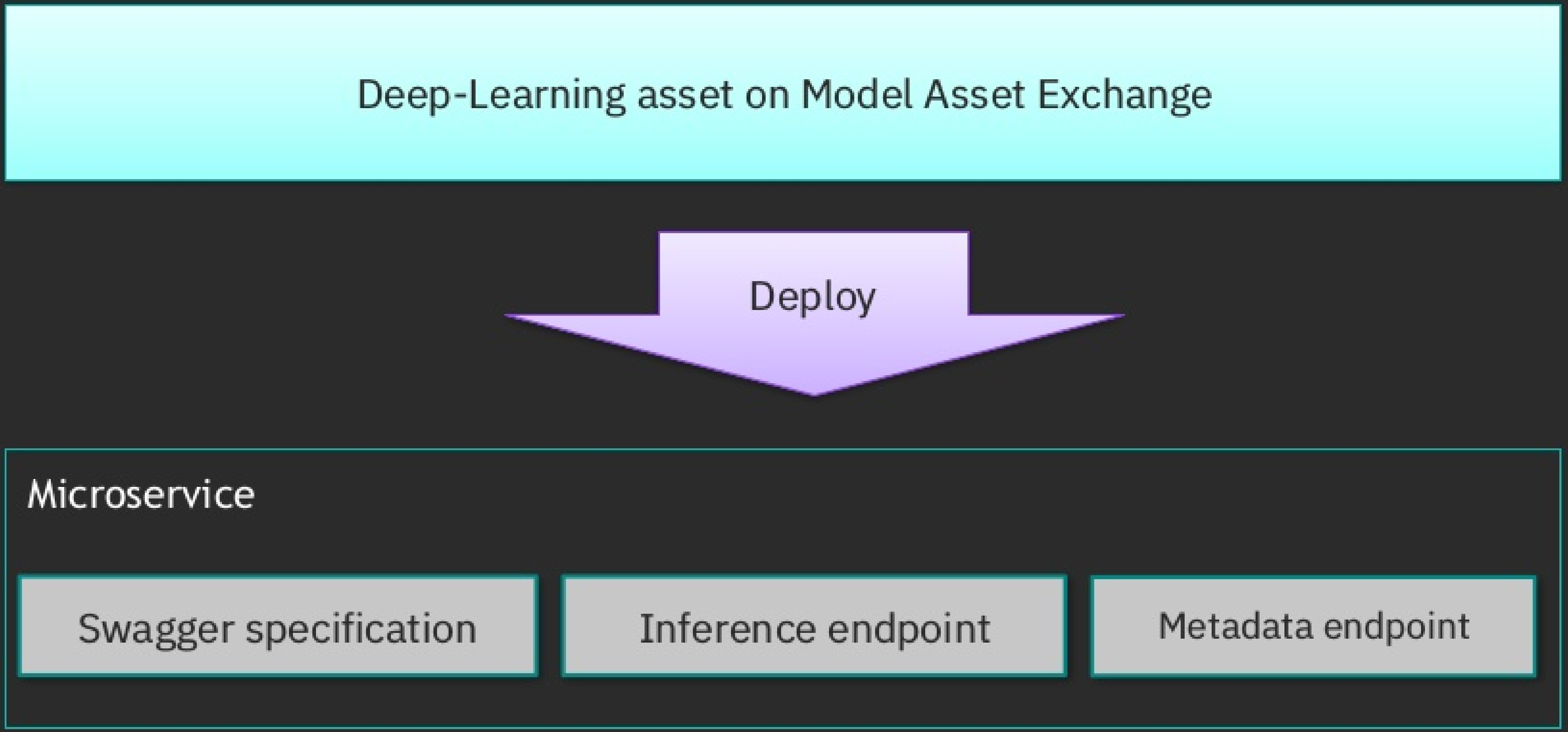
Pre-trained model

REST API

Deep-Learning asset on Model Asset Exchange

ibm.biz/model-exchange

Deployable Models



Deep-Learning asset on Model Asset Exchange

Deploy

Microservice

Swagger specification

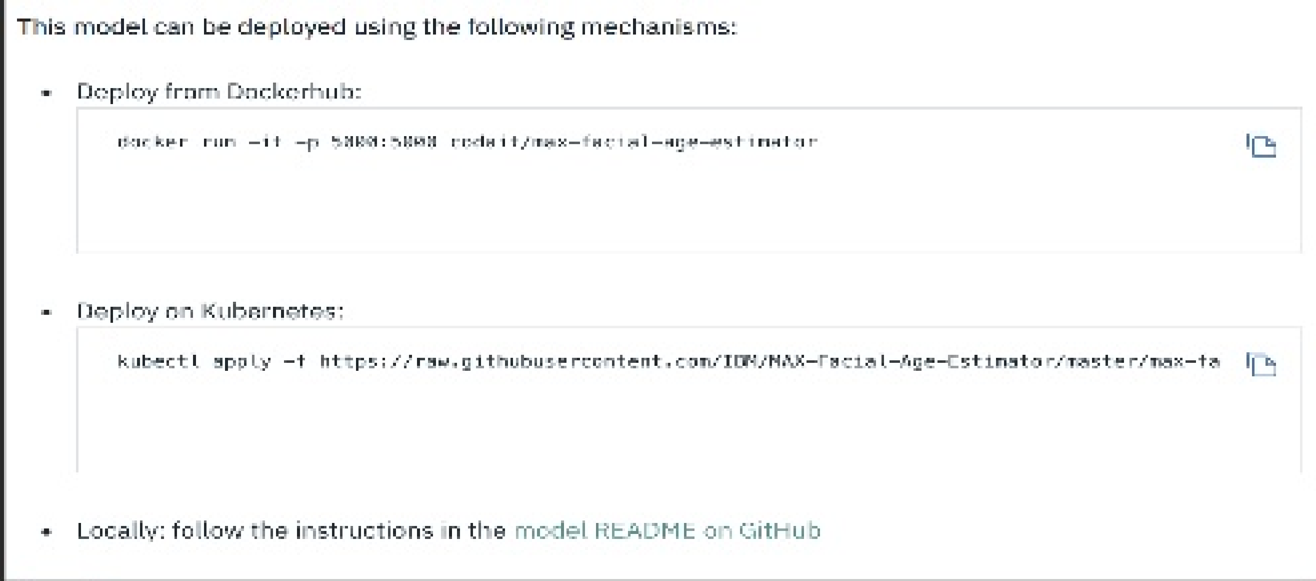
Inference endpoint

Metadata endpoint

Deployable Models

Highlights

- Image, audio, text, healthcare, time-series and more
- Pre- / post-processing & inference wrapped up in Docker container
- Generic API framework code - Flask RESTPlus
- Swagger specification for API
- One-line deployment locally and on a Kubernetes cluster
- Code Patterns demonstrating how to easily consume MAX models



This model can be deployed using the following mechanisms:

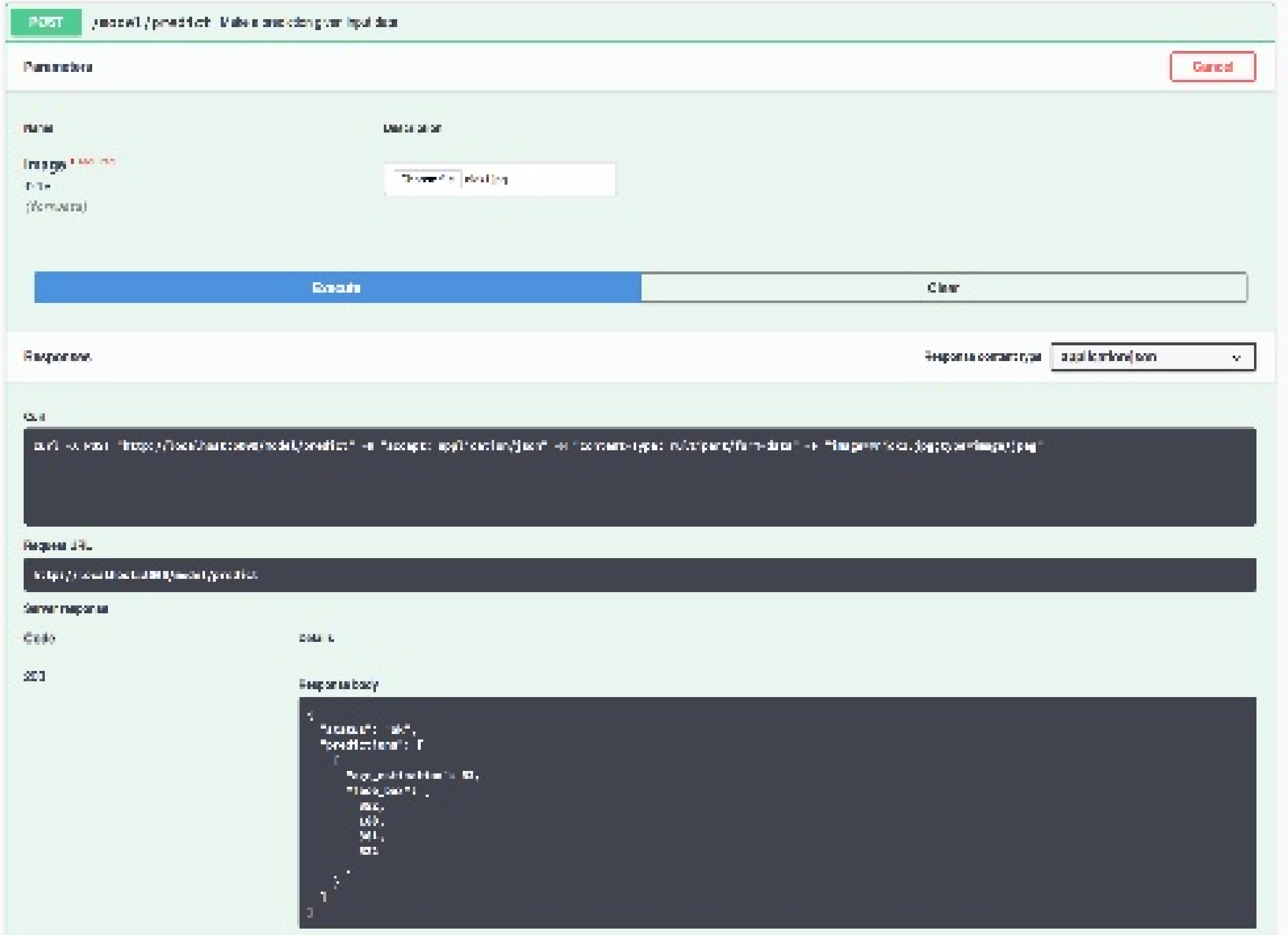
- Deploy from Dockerhub:

```
docker run -it -p 5000:5000 codait/max-facial-age-estimator
```

- Deploy on Kubernetes:

```
kubectl apply -f https://raw.githubusercontent.com/IBM/MAX-Facial-Age-Estimator/master/max-fa
```

- Locally: follow the instructions in the model README on GitHub



```
{
  "status": "ok",
  "predictions": [
    {
      [illegible]
```

Summary and Possible Future Directions

Current status

- 22 models (4 trainable)
- Image, audio, text, healthcare, time-series and more
- 3 Code Patterns demonstrating how to consume MAX models in a web app
- Code Pattern on training an audio classifier using Watson Machine Learning
- One-line deployment via Docker and on a Kubernetes cluster

Potential Future

- More deployable models – breadth and depth
- More trainable models - transfer learning in particular
- New MAX web portal launching soon
- More MAX-related content:
  - Code Patterns
  - Conference talks, meetups
  - Workshops
- Enhance production-readiness of MAX models
- Improve MAX API framework

IBM Developer

Model Asset eXchange

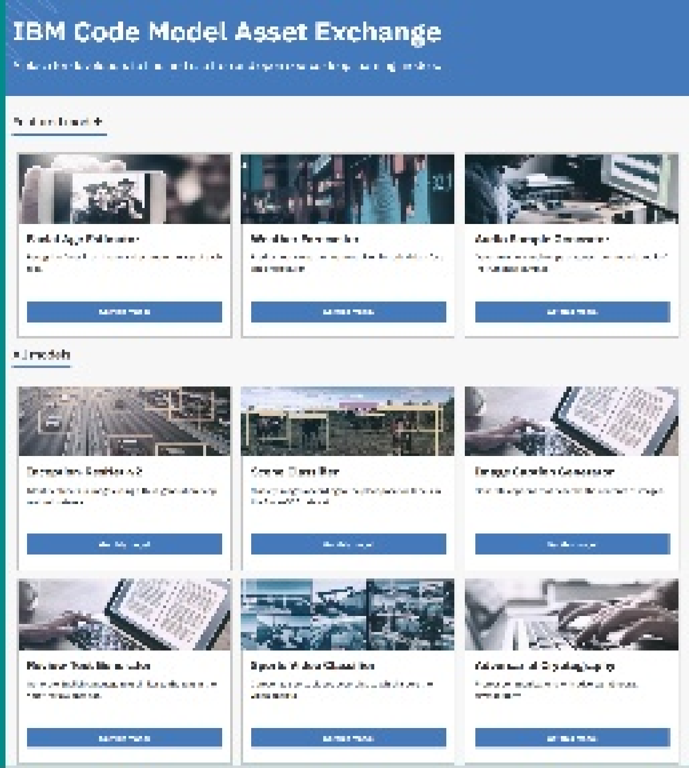
Free, open-source deep learning models.

Wide variety of domains.

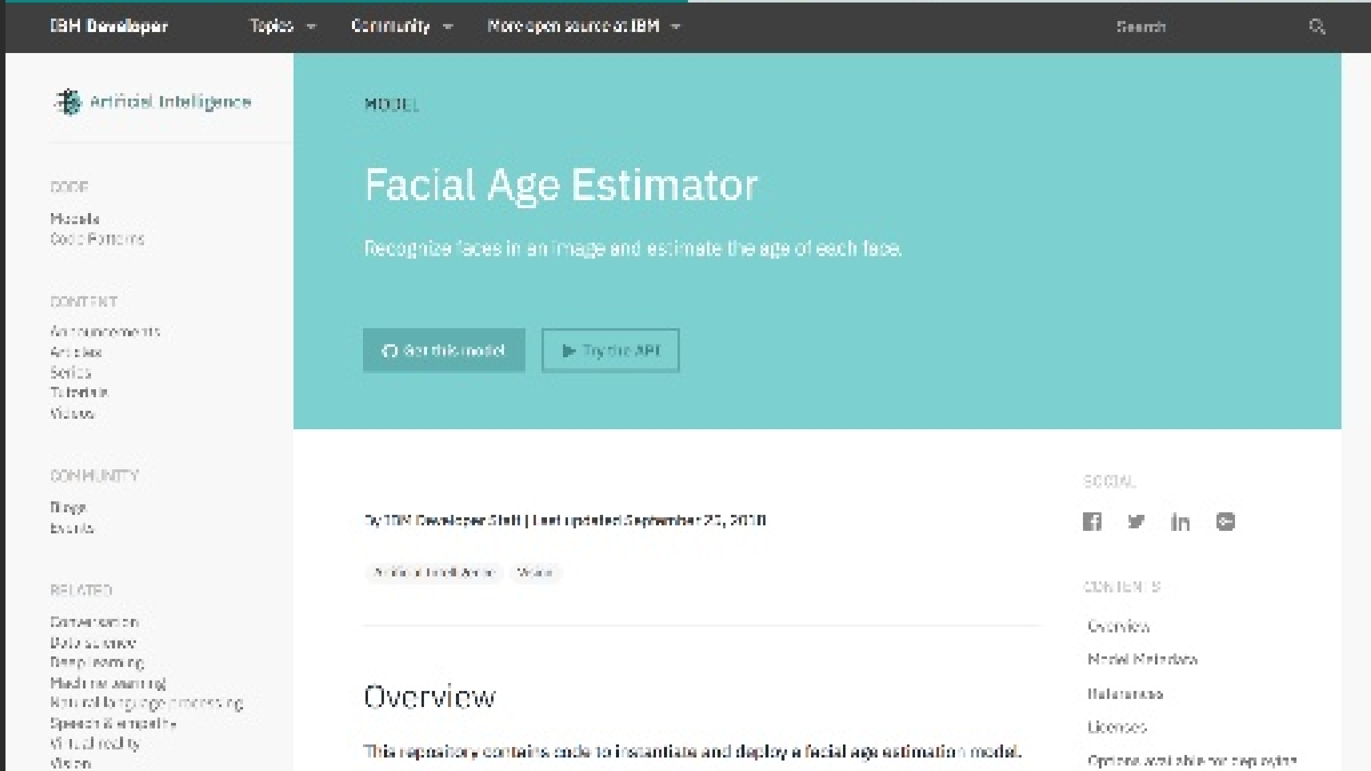
Multiple deep learning frameworks.

Vetted and tested code and IP.

http://ibm.biz/model-exchange



IBM Code Model Asset Exchange



Artificial Intelligence

CODE

Models
Code Patterns

CONTENT

Announcements
Articles
Series
Tutorials
Videos

COMMUNITY

Blogs
Events

RELATED

Conversation
Data science
Deep learning
Machine learning
Natural language processing
Speech & empathy
Virtual reality
Vision

MODEL

Facial Age Estimator

Recognize faces in an image and estimate the age of each face.

By IBM Developer Staff | Last updated September 20, 2018

Artificial Intelligence Vision

SOCIAL

CONTENTS

Overview
Model Metadata
References
Licenses
Options available for deploying

Overview

This repository contains code to instantiate and deploy a facial age estimation model.

IBM
CODE

Thank you!

codait.org

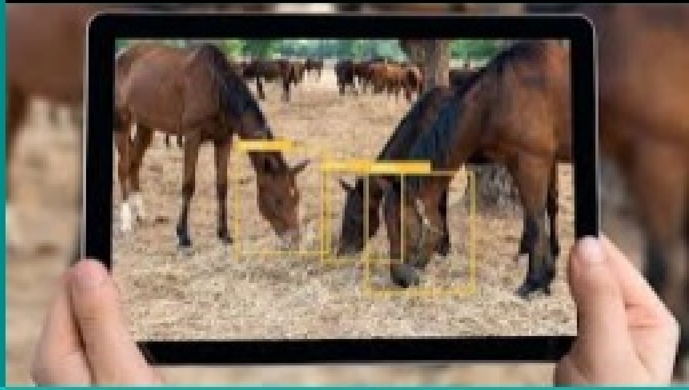
twitter.com/MLnick

github.com/MLnick

developer.ibm.com

FfDL

MAX



Sign up for IBM Cloud and try Watson Studio!

https://ibm.biz/BdYbTY

https://datascience.ibm.com/

IBM

IBM
CODE