Building Machine Learning Model on Kubeflow



Abhishek Kumar
DATA SCIENTIST | AUTHOR | SPEAKER
@meabhishekkumar



Overview



Model development process and challenges

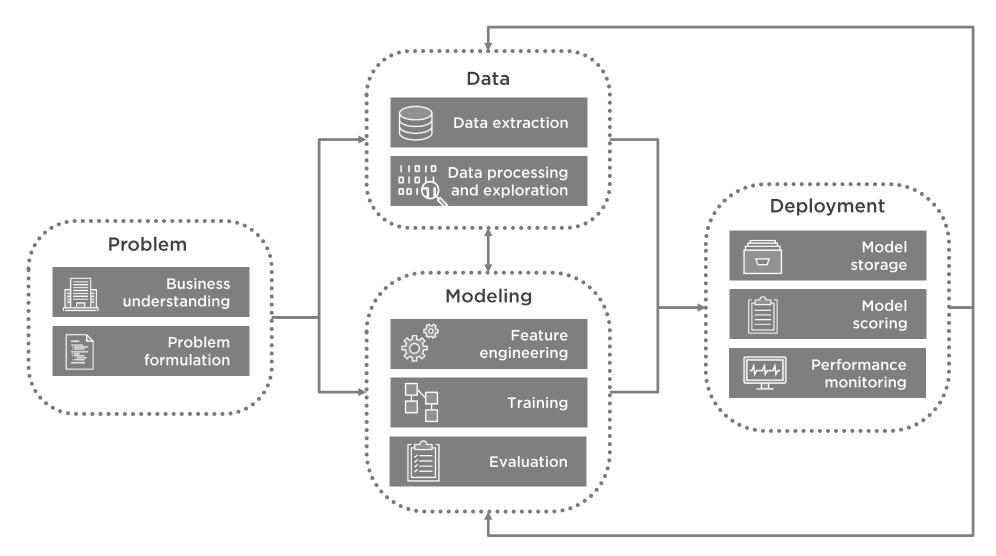
Kubeflow components for training

Demo: Building machine learning model

- Setup notebook server
- Train locally
- Fairing
- Distributed training job (GPU, multiworker)
- Hyperparameter tuning with Katib
- Export model



Machine Learning Workflow





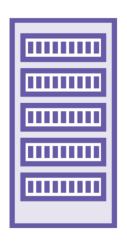


Track experiments

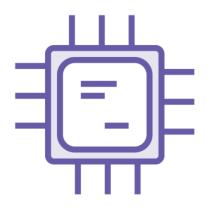
Increase productivity and ensure reproducibility



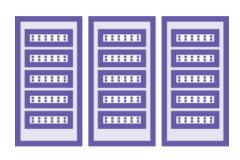
Execution







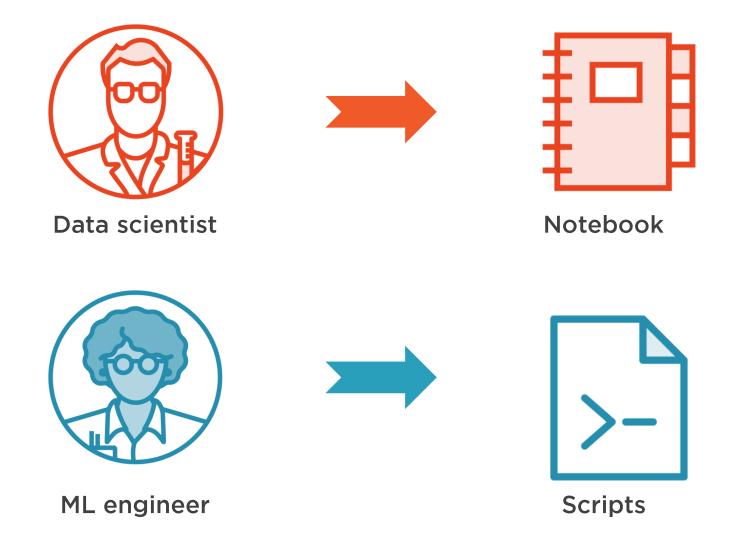
Accelerators GPU, TPU



Multi-node/Multi-worker
Distributed training



Development Environment





Kubeflow Components for Training

Notebook server

Interactive multi-user environment

Training at scale

TFJob, PyTorch, MXNet

Fairing

Training jobs from notebook

Metadata

Track model artifacts and metadata

Katib

Hyperparameter tuning



Fashion-MNIST Training Workflow

Setting up

Notebook

Pre-built and custom image

Fairing

Notebook

Kubernetes cluster

Modeling

Script

Multi node, multi worker

Modeling with metadata tracking

Notebook

CPU training, export model to GCS

Modeling

Notebook

GPU

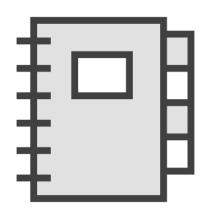
Hyperparameter tuning

Script

Katib



Kubeflow Notebook



Notebook server

Use pre-existing or custom image

Authentication and access control

Attach persistent volume for data or workspace

Configure resources (CPU, RAM)

Configure accelerators such as GPU





Setup notebook server with pre-built image





Setting up notebook server with custom image

- Build Docker image
- Push image to GCR
- Use custom image to setup notebook server



Why Custom Image?



Pre-built images are not sufficient for the use case



Custom images for different teams or use cases e.g. exploration, classical machine learning, deep learning (TensorFlow/PyTorch)



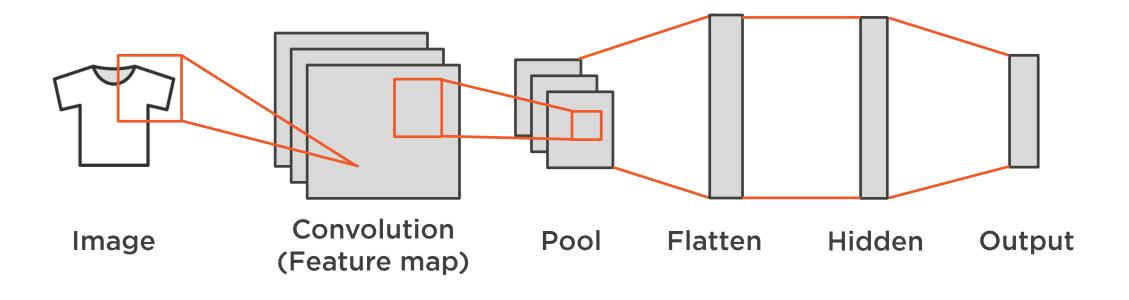
Central team managing custom images for faster onboarding of data scientists



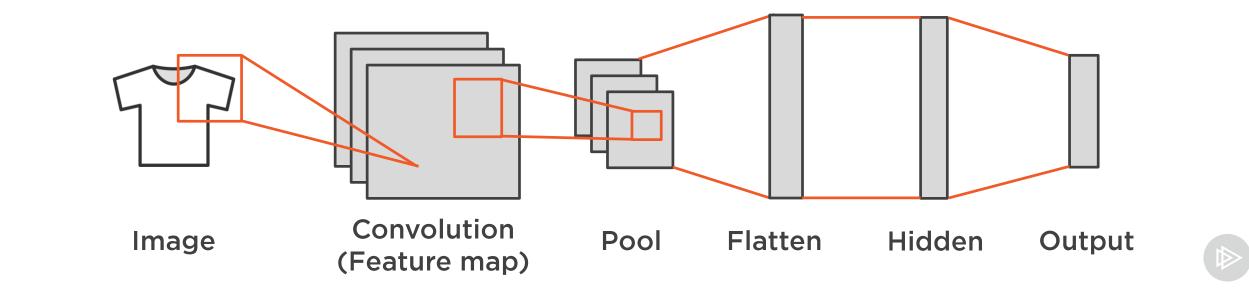
Fashion-MNIST: Convolutional Neural Networks (CNN) deep learning model

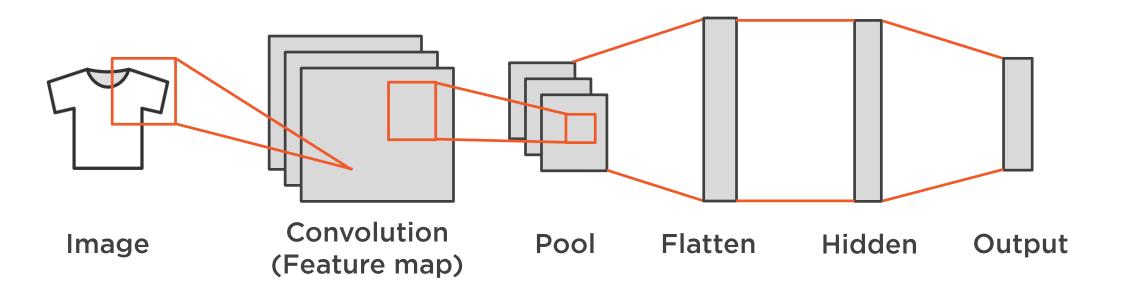


```
model = tf.keras.Sequential([
tf.keras.layers.Conv2D(filters=32, kernel_size=(3,3),
activation='relu', input_shape=(28, 28, 1), name='x'),
tf.keras.layers.MaxPooling2D(),
tf.keras.layers.Flatten(),
tf.keras.layers.Dense(64, activation='relu'),
tf.keras.layers.Dense(10, activation='softmax')
```













Training in Kubeflow notebook

- Build, train, and evaluate model
- TensorBoard
- Model export
- Upload to GCS





Metadata

Track and manage metadata for ML workflow

Backend database to store information

API to query and retrieve information

Artifact store dashboard

Track metadata about

- Model
- Metric
- Dataset





Metadata tracking

- Connect to metadata store
- Log metadata
- List metadata
- Kubeflow artifact dashboard



Kubeflow Fairing

Python package to streamline the process

- Build
- Train anywhere (local/cloud)
- Deploy

Abstraction layer

- Can run directly from notebook
- Reusable building blocks

Targeted for data scientists



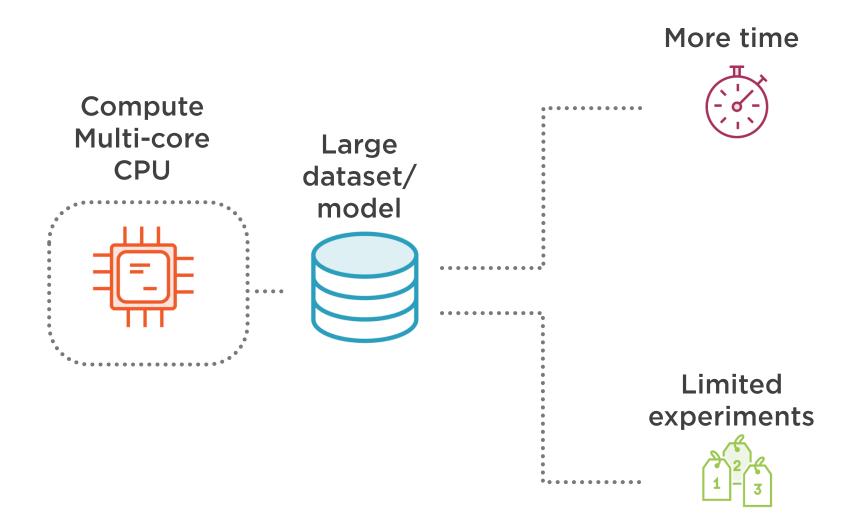


Training job with Kubeflow Fairing

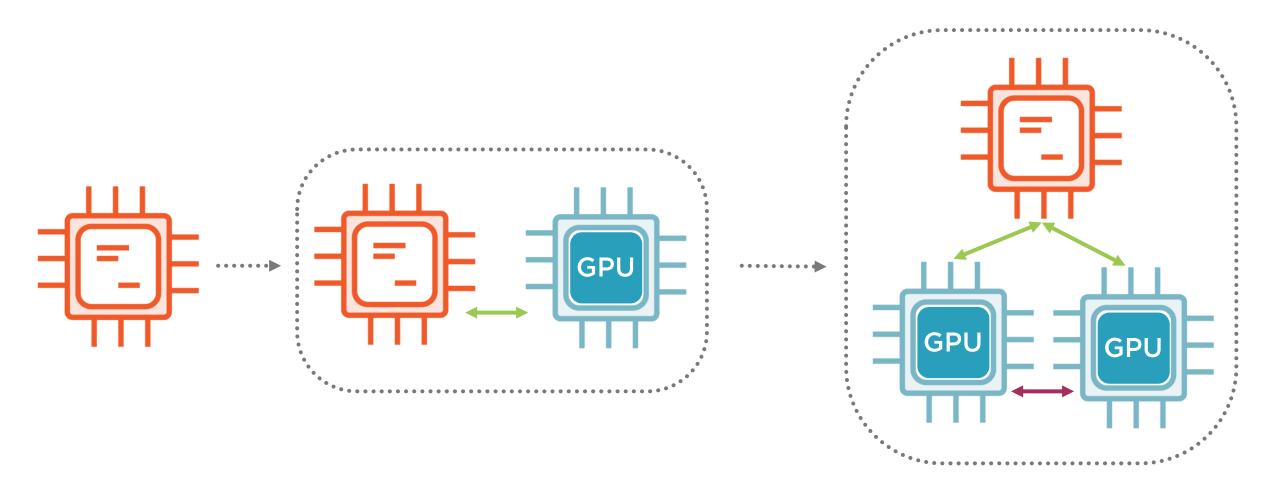
- Local
- Kubernetes cluster



Distributed Training

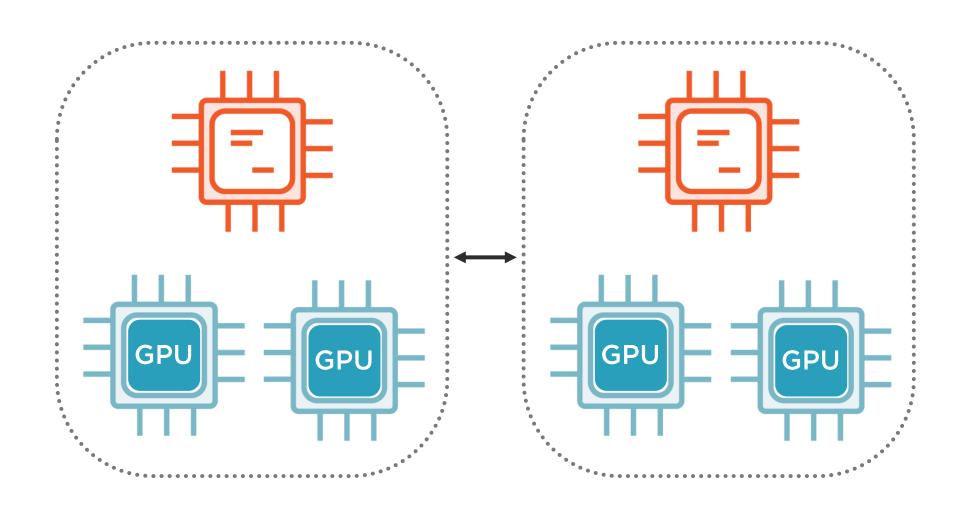


Distributed Training



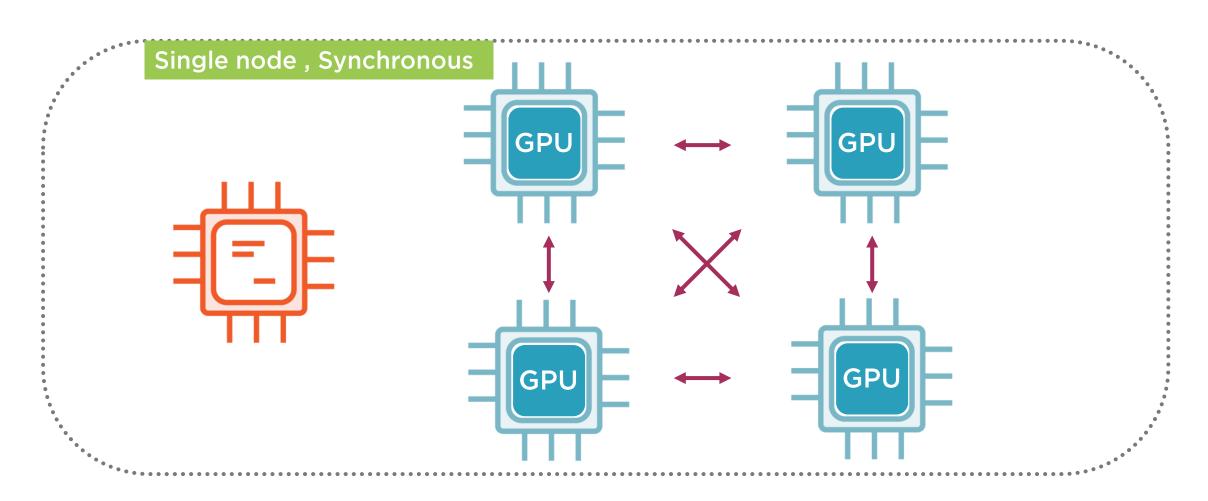


Distributed Training



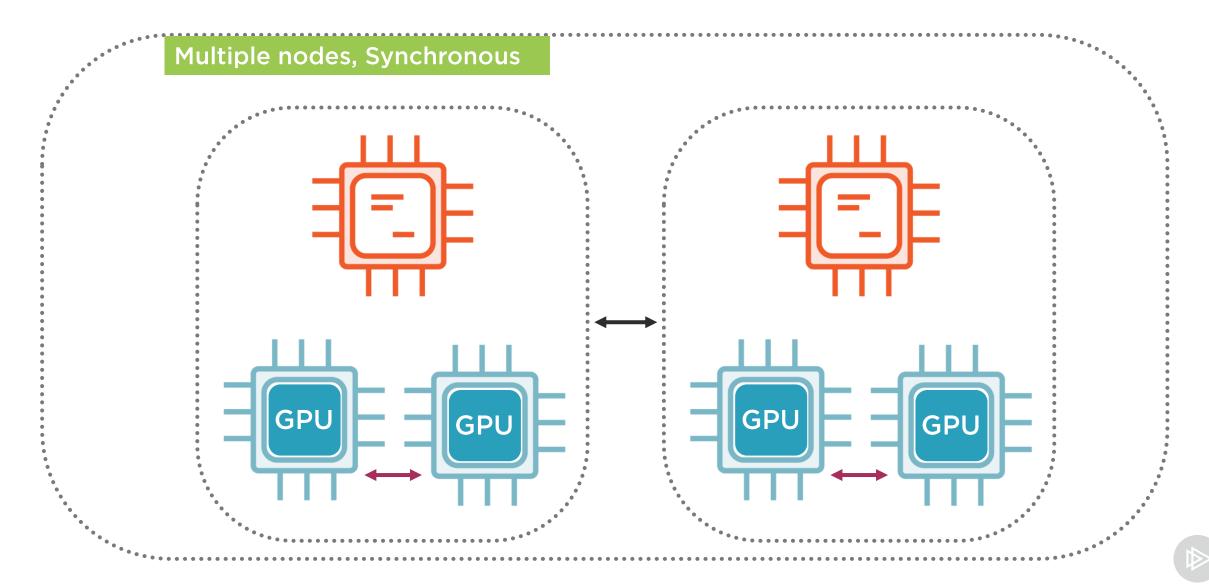


Mirrored Strategy





Multi-worker Mirrored Strategy





Training in Kubeflow notebook with GPU

- Attach GPU to notebook server
- Multi-GPU training
- Check GPU utilization





Distributed training with TFJob



Hyperparameter Tuning

Hyperparameters

- Configuration parameters such as learning rate, batch size
- Set before training process

Hyperparameter tuning

- Finding optimal value that optimize objective function such as accuracy
- Optimal values improve model performance



Inspired by Google Vizier

Framework agnostic

- TensorFlow
- PyTorch
- MxNet

Katib

Support multiple optimization algorithms

- Random search
- Grid search
- Bayesian optimization
- Hyperband



Experiment

- End-to-end process
- Objective: what to optimize, search space, search algorithm

Suggestion

- Optimization algorithm

Katib

Trial

- One iteration of optimization process with one set of hyperparameters
- Parameters, observation metrics

Job

- Evaluate a trial
- Calculate objective





Hyperparameter tuning with Katib



Summary



Model training challenges

Model training using Kubeflow

- Notebooks
- Kubernetes cluster
- GPU training
- Multi worker training

Hyperparameter tuning with Katib



Next up: Serving Machine Learning Model on Kubeflow

