

Mastering Transformers: From Building Blocks to Real-World Applications

Machine Learning Lifecycle

Prof. Alptekin Temizel

Graduate School of Informatics, Middle East Technical University

11 Sept 2023



Real-World ML Systems

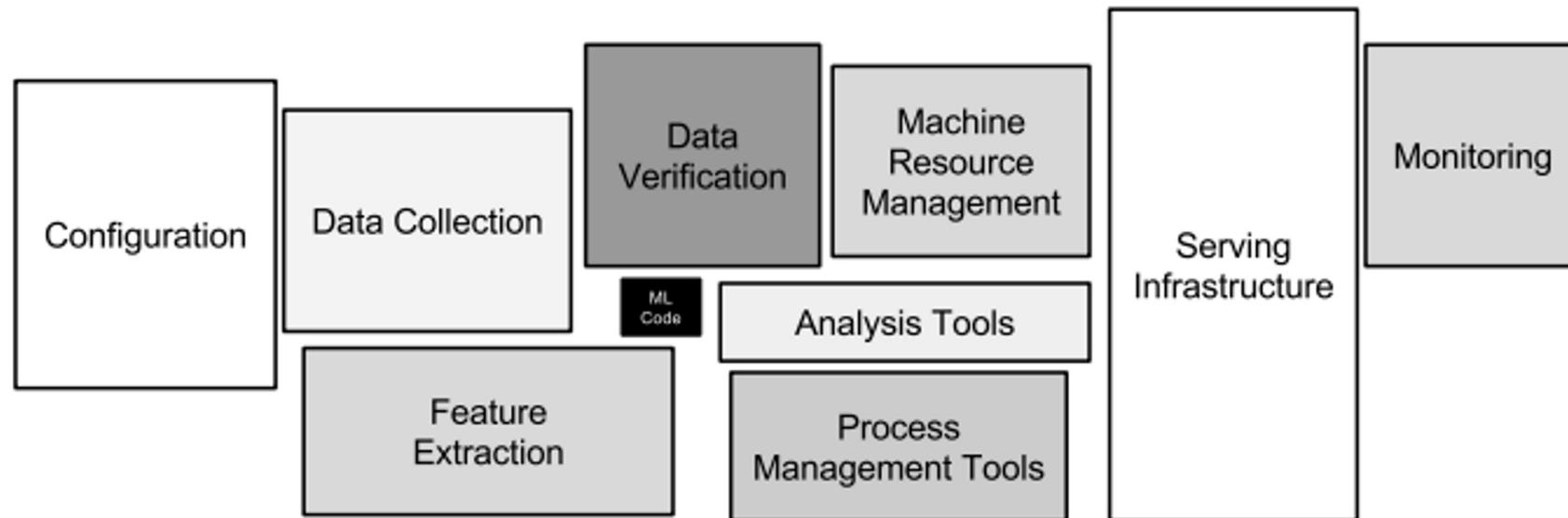
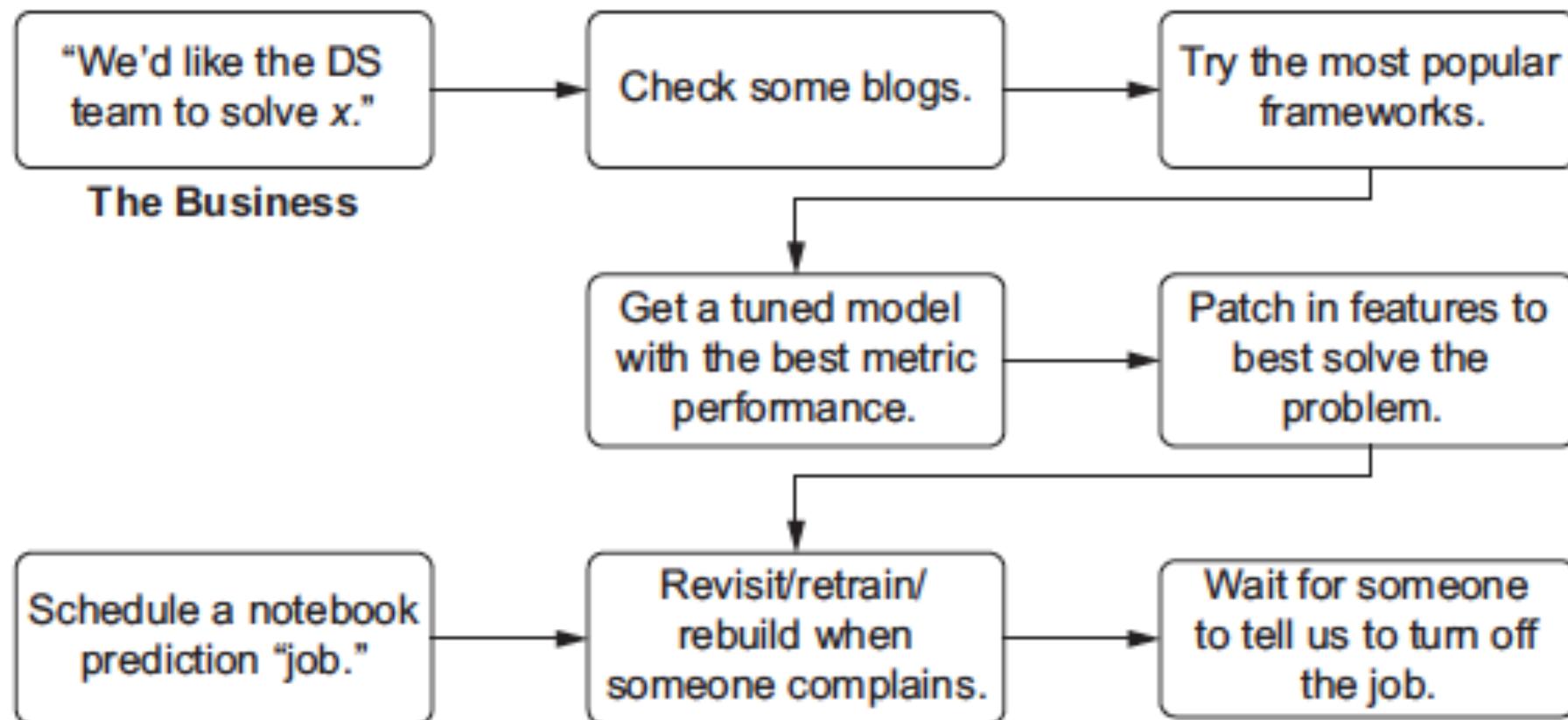


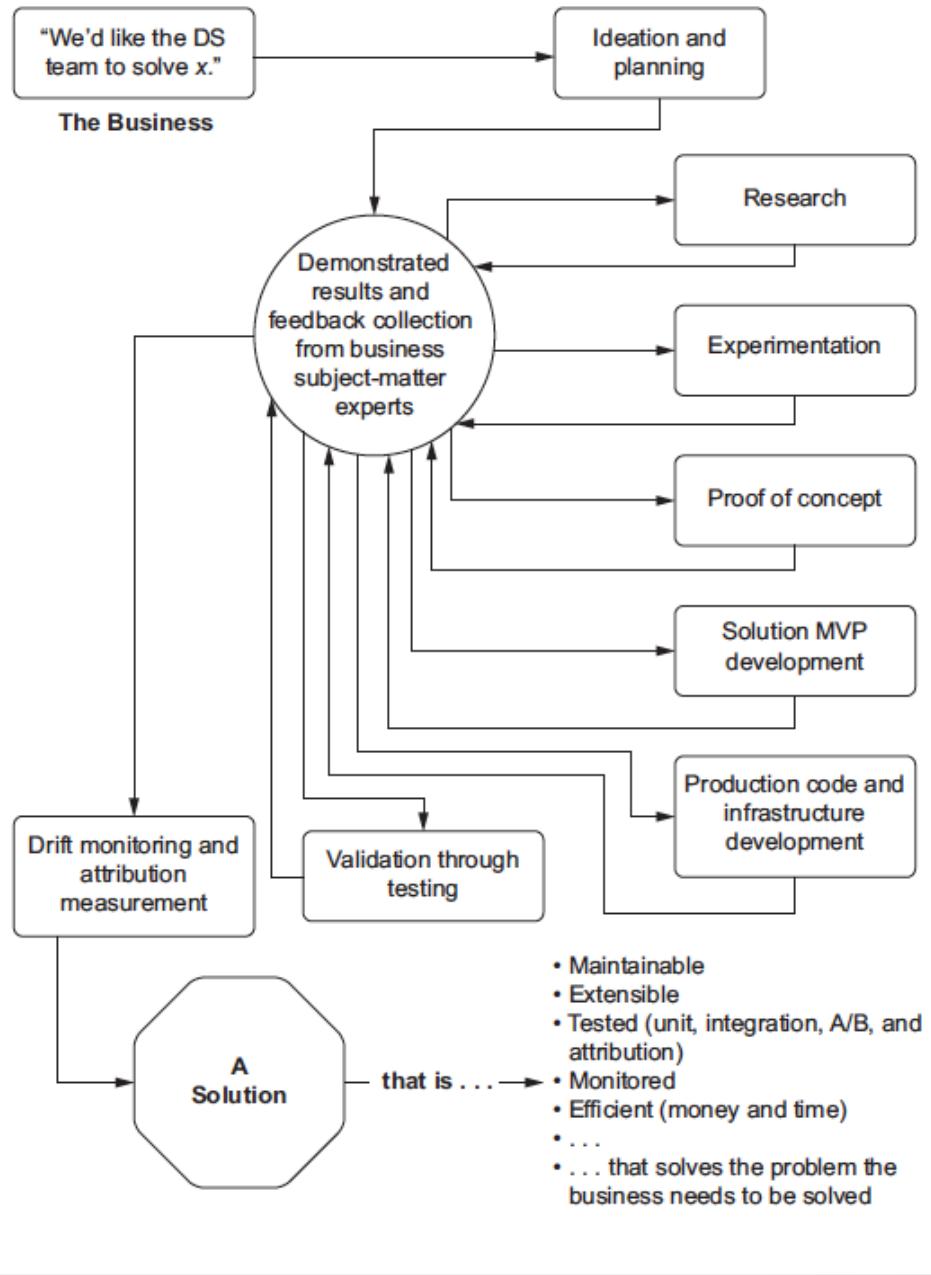
Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

Real-World ML Systems

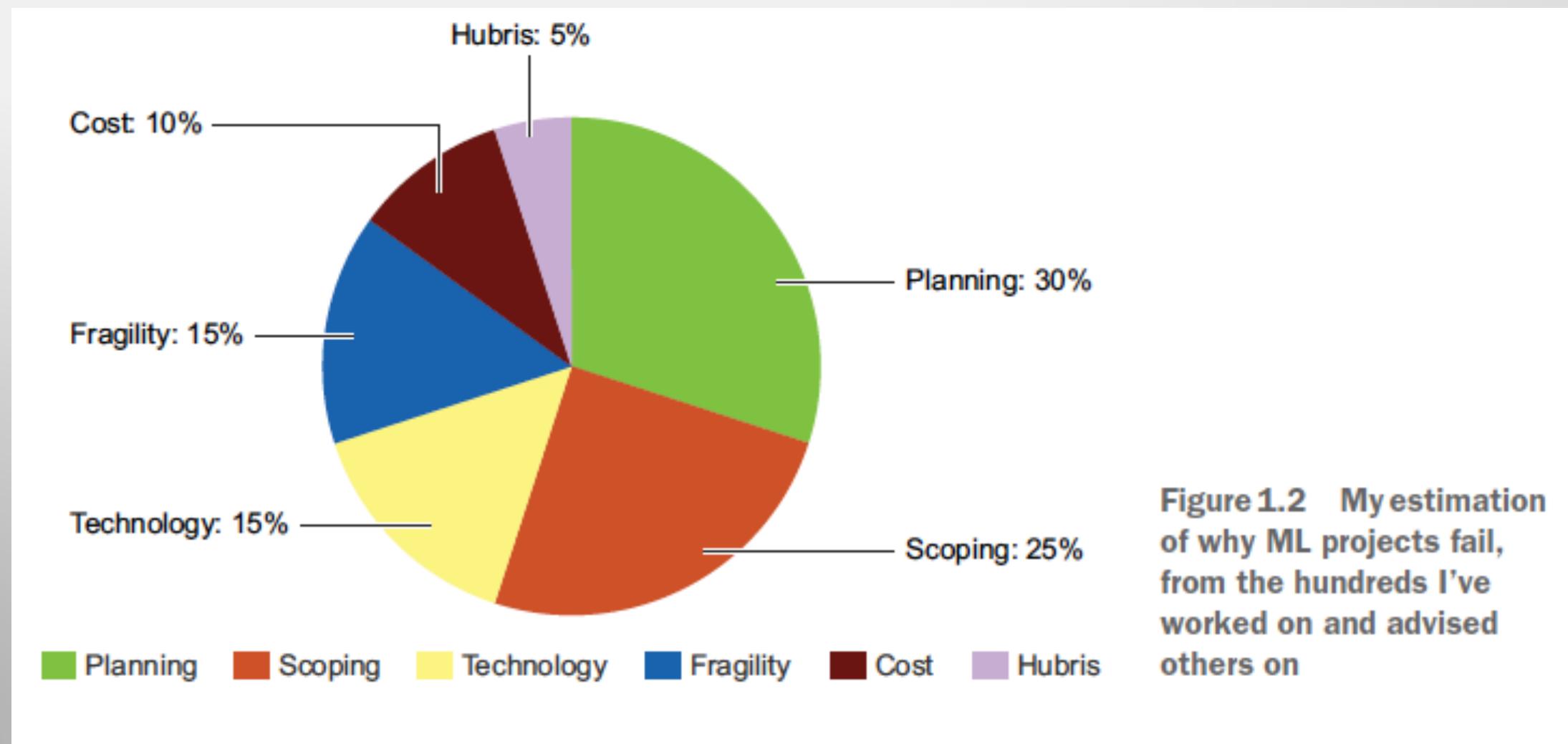
How a lot of failed ML projects are built



Real-World ML Systems



Real-World ML Systems



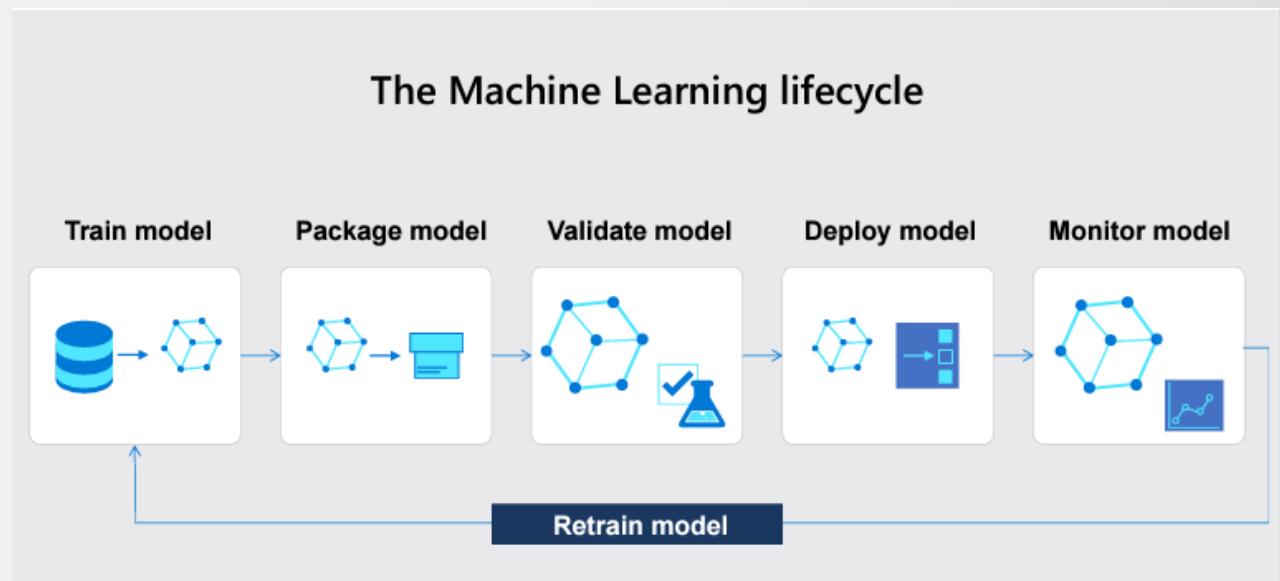
Machine Learning Lifecycle

First:

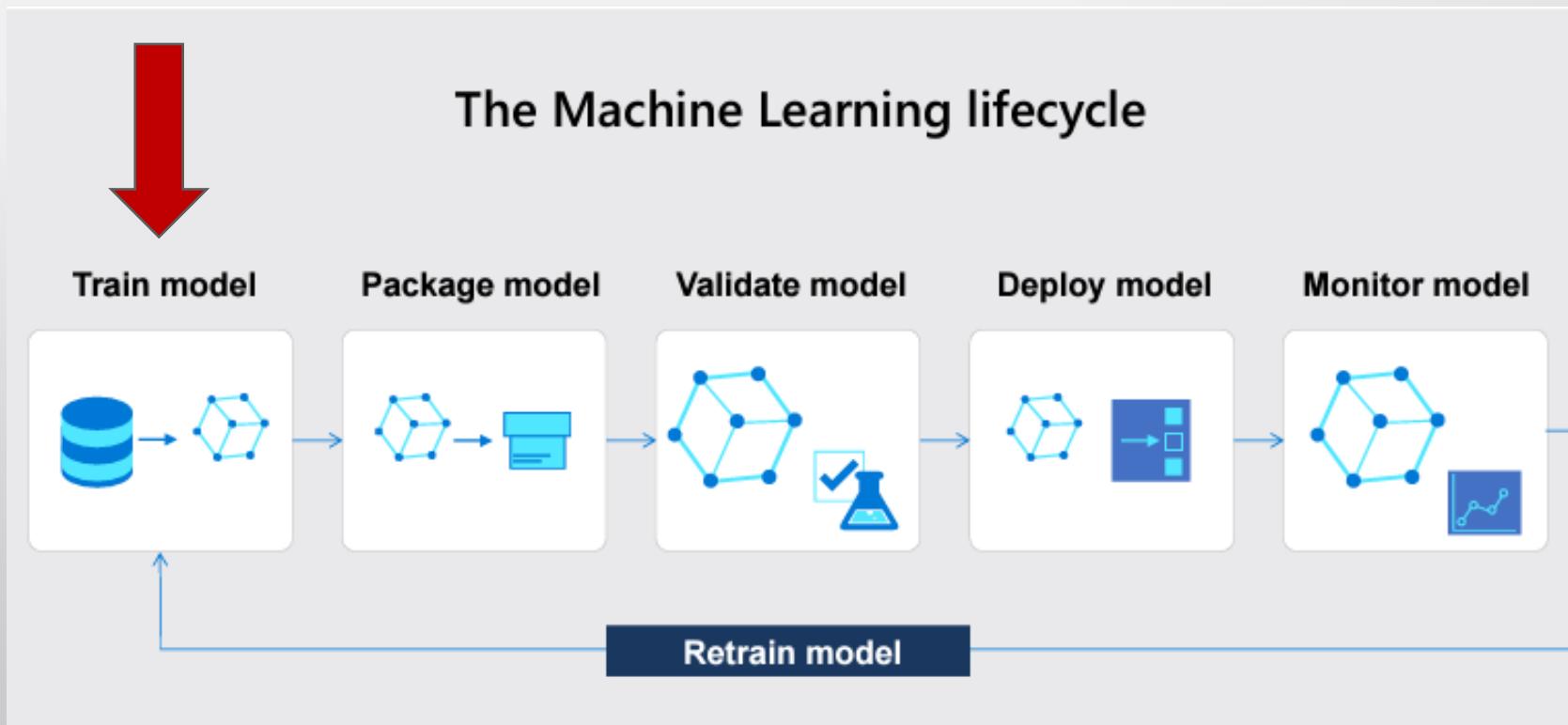
- Requirement Analysis and Definition of Targets
- Data Management
 - Data Acquisition
 - Pre-Processing
 - Data Analysis

Then:

- Model Training and Optimization
- Packaging
- Validation
- Deployment
- Monitoring and Updating



Machine Learning Lifecycle



Graphics Processing Units (GPU)

GPUs are fast...

GTX 285 has 240 cores, 1 TFLOPS

GTX 480: 1345 GFLOPS 250W, March 2010

GTX 590: 2488 GFLOPS 244W, March 2011

GTX 680: 3090 GFLOPS 195W, March 2012

GTX 780Ti: 5046 GFLOPS 250W, November 2013 (649\$)

GTX 980: 4612 GFLOPS , 165W, September 2014 (549\$) (Later: 5632 GLOPS, 250W)

GTX 980 notebook: 4612 GFLOPS, 145 W, September 2015

GTX 1080: 9 TFLOPS, 180W, May 2016 (599\$)

GTX 1080TI: 11.3 TFLOPS, 250W March 2017 (699\$)

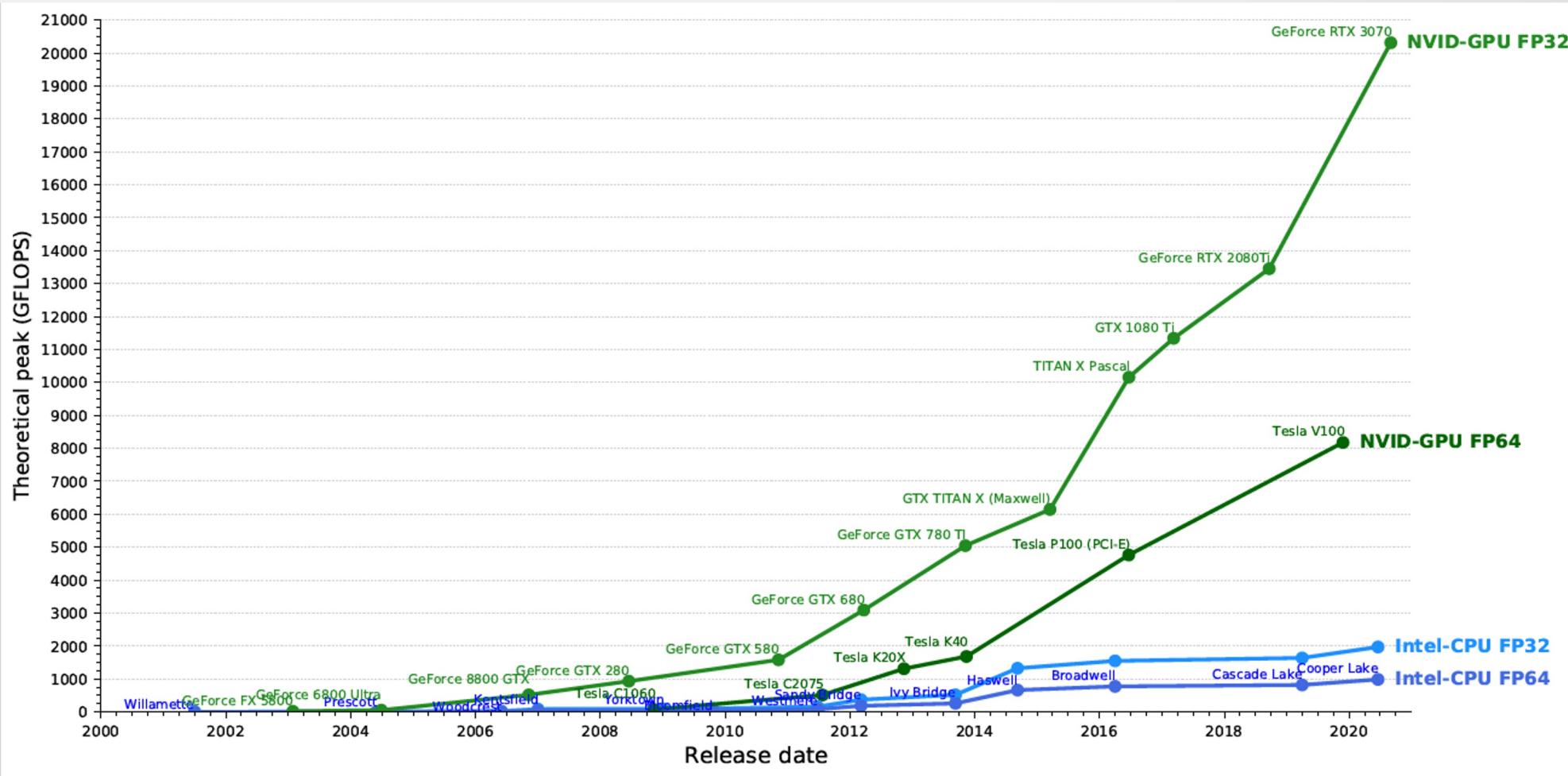
RTX 2080TI: 13.4 TFLOPS, 250W, Sept 2018 (999\$)

RTX 3080: 29.8 TFLOPS, 320W, Sept 2020 (700\$)

RTX 4080: 48.7 TFLOPS, 390 TFLOPS FP16, 780 TFLOPS INT8, 320W, Nov 2022 (1200\$)

Note: Intel Core i7-8700K 6-core CPU has a performance of 218 GFLOPS @95W

Graphics Processing Units (GPU)



Graphics Processing Units (GPU)

TRAINING

FULLY INTEGRATED DL SUPERCOMPUTER



DGX-1 & DGX Station



DESKTOP



RTX/GTX
series

DATA CENTER



Tesla A100
Tesla H100

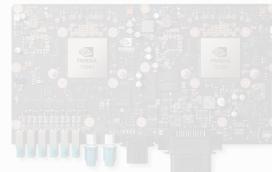
INFERENCE



Tesla A100/V100



Tesla T4



Drive PX2



Jetson TX2



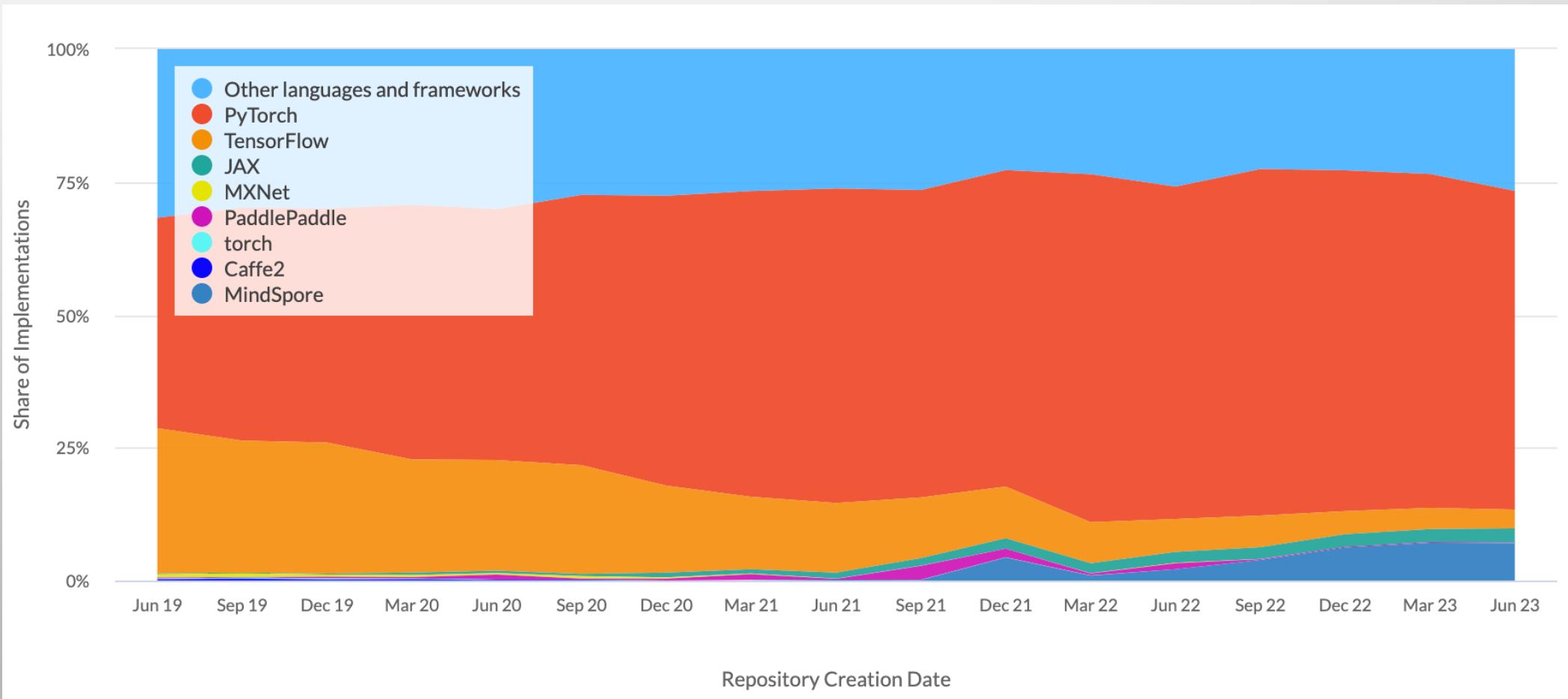
Jetson
Xavier

Training Hardware - Google

Google Cloud Tensor Processing Units (TPUs)

- TPUs are ASICs designed to accelerate machine learning algorithms.
- They mainly accelerate linear algebra operations and they can be used to train models using Tensorflow.

Training: Frameworks



Source: <https://paperswithcode.com/trends>

Training: Hyper-Parameter Optimization (HPO)

- Selecting the best hyper-parameters
- HPO methods are based on training the model several times
- Each new hyper-parameter results in adding a new dimension and exponentially increases the computational complexity. Hence, HPO requires high resource use
- Ideally HPO needs to take the target platform into account. When optimizing for embedded systems and mobile devices, energy consumption and memory limitations (hardware-aware ML) and model accuracy and hardware efficiency need to be optimized in conjunction.

Training: Hyper-Parameter Optimization (HPO)

Optuna: Optimize Your Optimization

An open-source hyper-parameter optimization framework.

Aims to automate hyperparameter search.

Key Features

Eager search spaces



Automated search for optimal hyperparameters using Python conditionals, loops, and syntax

State-of-the-art algorithms



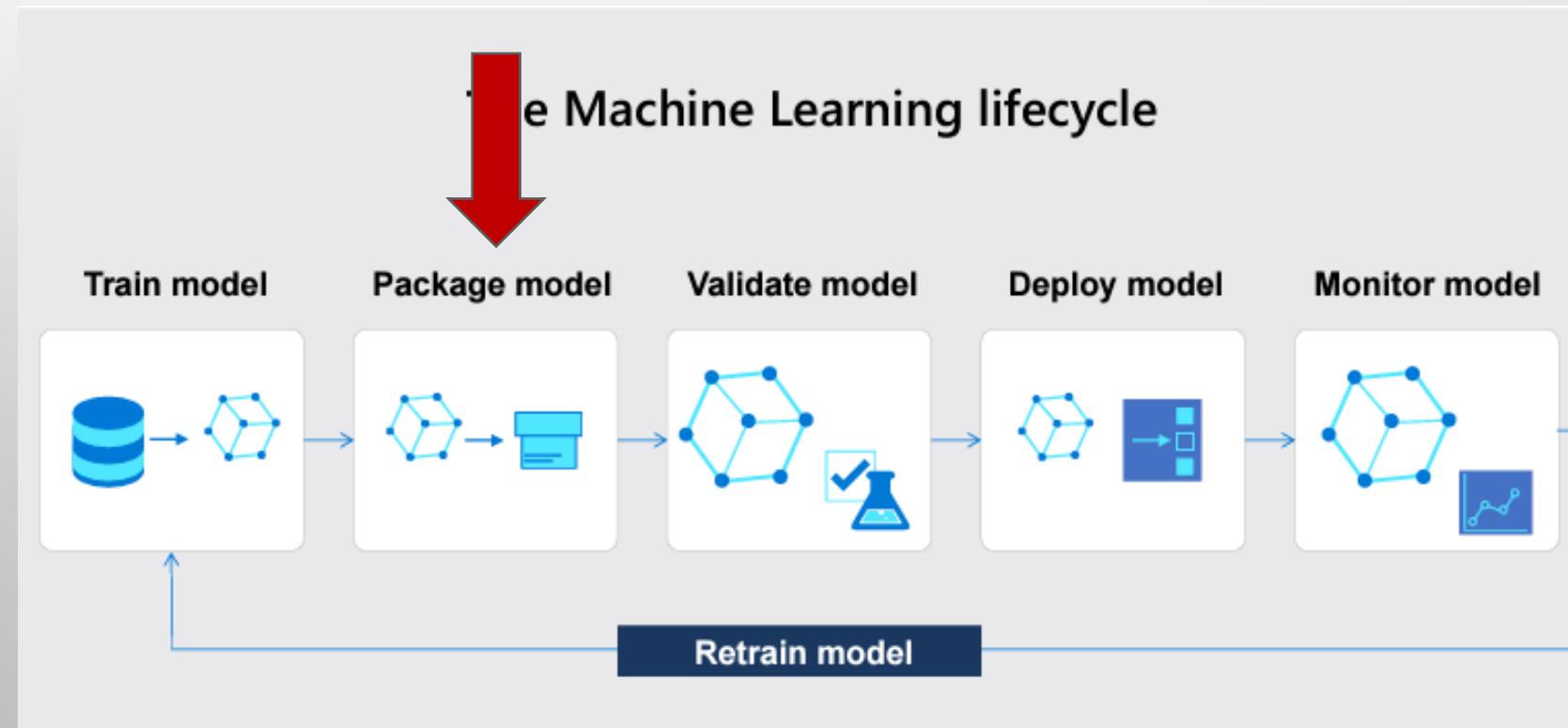
Efficiently search large spaces and prune unpromising trials for faster results

Easy parallelization



Parallelize hyperparameter searches over multiple threads or processes without modifying code

Machine Learning Lifecycle



Packaging: Open Neural Network Exchange (ONNX)

- Training frameworks are not designed for efficient inference.
- Model formats may change in the future.
- Once the training is done, it is desirable that the models are exported in a portable format to use with specialized inference engines.



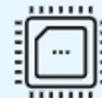
KEY BENEFITS



Interoperability

Develop in your preferred framework without worrying about downstream inferencing implications. ONNX enables you to use your preferred framework with your chosen inference engine.

[SUPPORTED FRAMEWORKS >](#)



Hardware Access

ONNX makes it easier to access hardware optimizations. Use ONNX-compatible runtimes and libraries designed to maximize performance across hardware.

[SUPPORTED ACCELERATORS >](#)

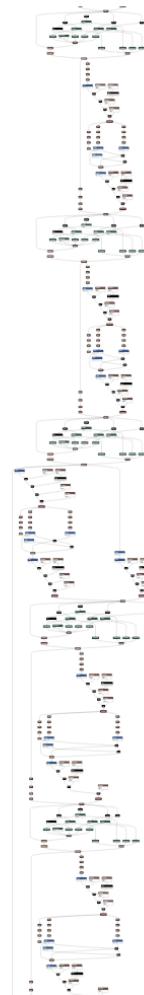
Packaging: Open Neural Network Exchange (ONNX)

- ONNX stores the data using Protocol Buffer (protobuf); a message file format developed by Google.
- This format is also used by Tensorflow and Caffe frameworks.
- In protobuf, only the data types (such as Float32) and the order of the data are specified, the meaning of each data is left up to the software used.
- ONNX outputs of the frameworks may have redundancies and a simplification step may be used: ONNX Simplifier: <https://github.com/daquexian/onnx-simplifier>
- ONNX files can be visualized using Netron.

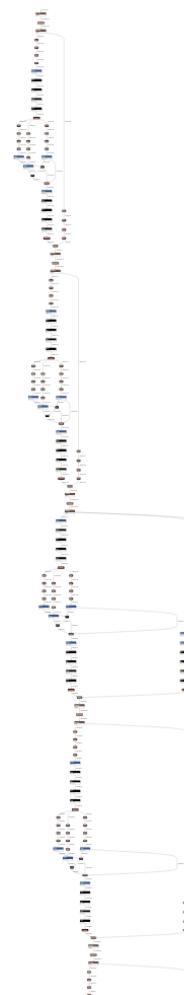
Packaging: Open Neural Network Exchange (ONNX)

ONNX outputs of the frameworks may have redundancies and a simplification step may be used: ONNX Simplifier.

<https://github.com/daquexian/onnx-simplifier>



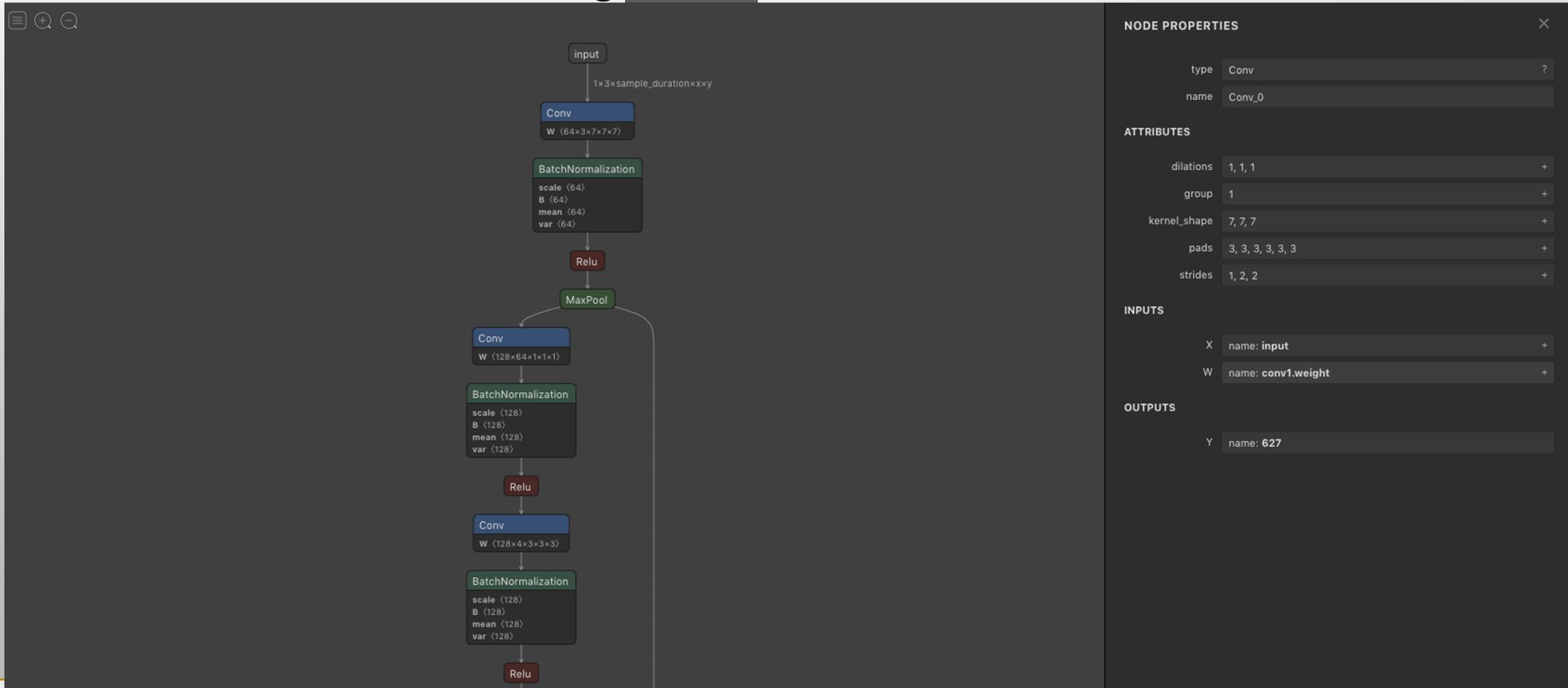
Old model
(3.4M)



New model
(1.9M)

Packaging: Open Neural Network Exchange (ONNX)

- ONNX provides a definition of an extensible computation graph model.
- ONNX files can be visualized using [Netron](#).



Machine Learning Life-Cycle

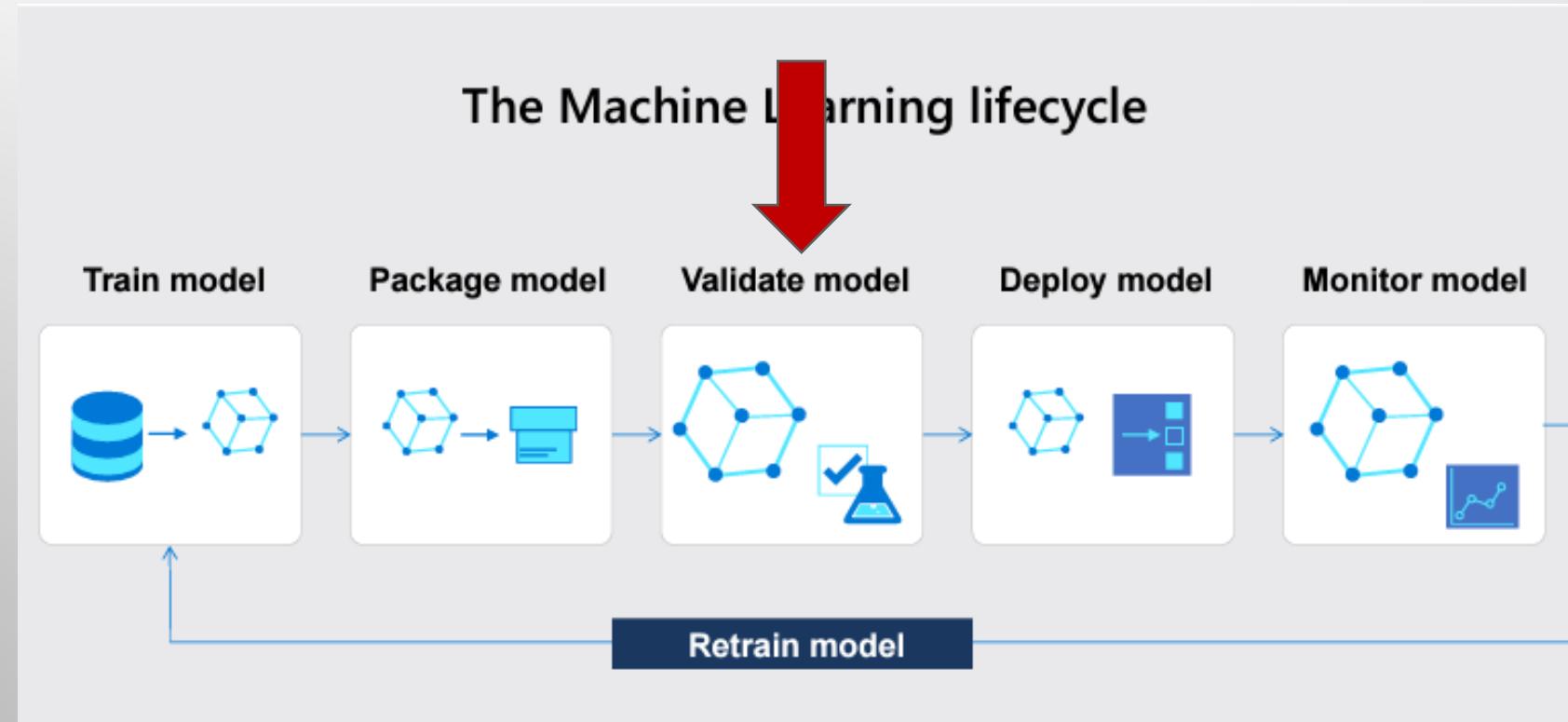


Figure from “Drive Efficiency and Productivity with Machine Learning Operations”, Microsoft Azure White Paper

Model Validation

- In addition to meeting desired functionality and performance requirements, a model must not crash or cause errors when loaded or when it receives bad or unexpected inputs.
- In addition, it must not use too many system resources.

Model Validation

Ideally, model validation has two parts:

- Unit and integration testing of the model itself,
- Functional and performance testing of the model as embedded into an app or service.

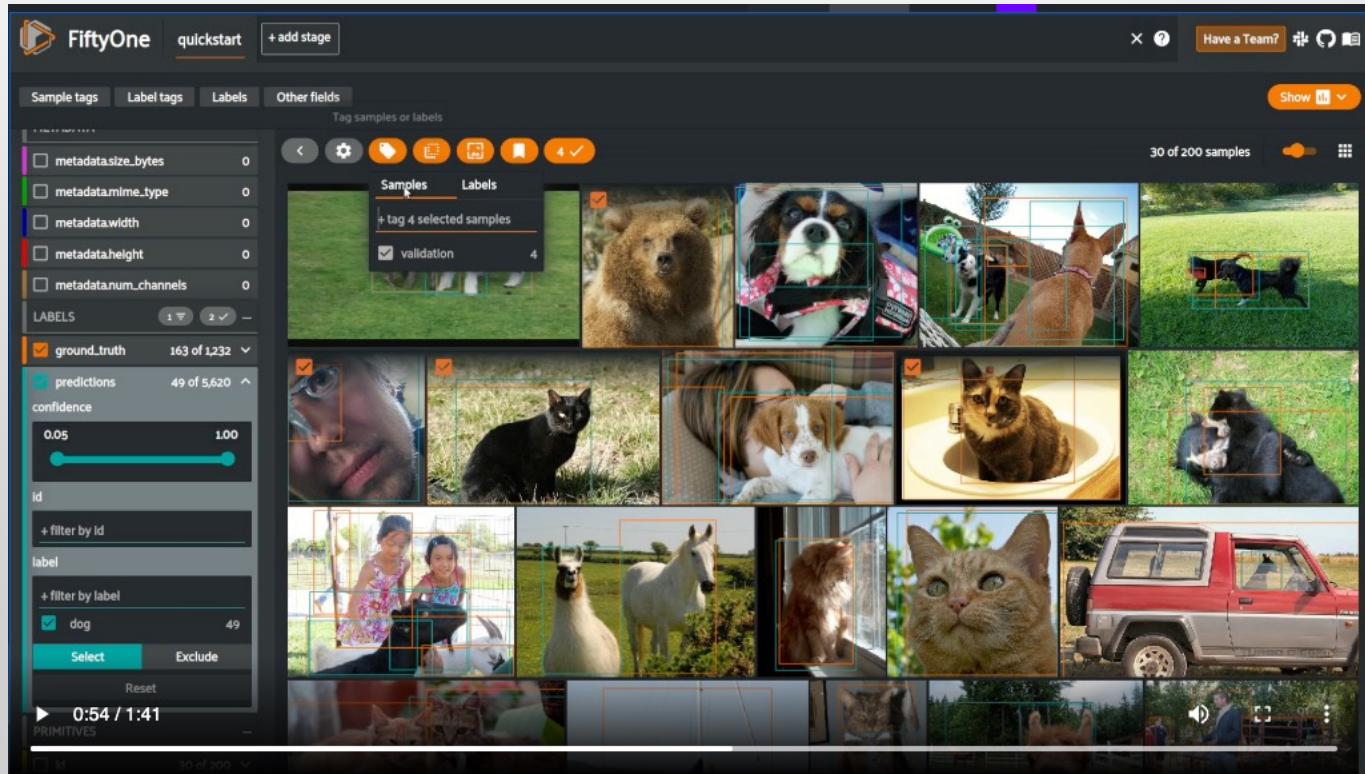
For example, if you train a model with a different format of input data than what is available to the inferencing service, it might work well during the training process, but perform poorly in production.

Model Validation – Testing the Model

- Traditional unit and integration tests that are run on a small set of inputs should produce stable results.
- Unlike a traditional (non-ML) app, these will be statistical results – that is, there will be a range of acceptable values (an expected target and its evolution) versus a finite sign-off criteria.
- This can also involve testing the data used to produce the model, as required to ensure that it matches what will be available during the scoring scenario in terms of schema and features

Model Validation – Testing the Model

This can also involve testing the data used to produce the model, as required to ensure that it matches what will be available during the scoring scenario in terms of schema and features.



Model Validation – Testing the Model

TensorBoard Projector: Visualize embeddings.

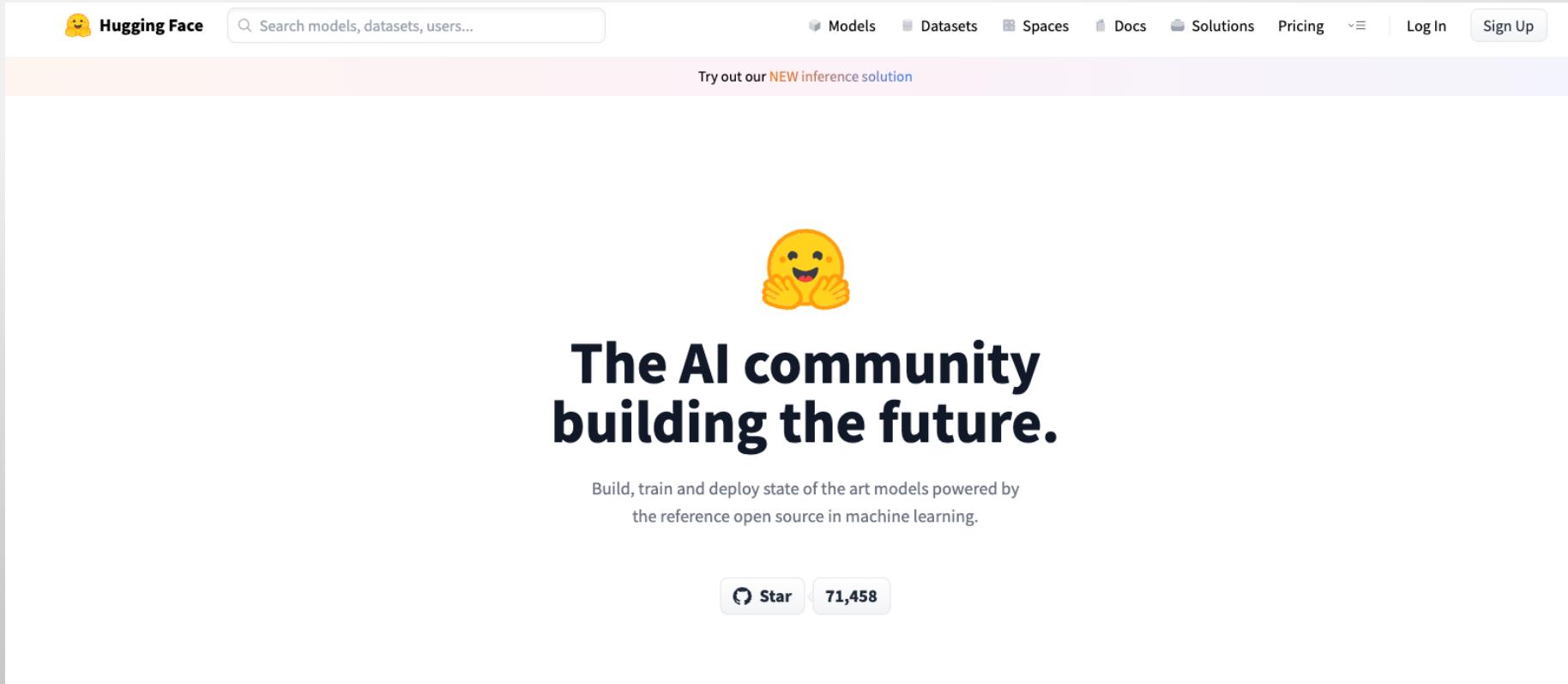
Search for specific terms, and highlights words that are adjacent to each other in the embedding (low-dimensional) space.

<https://projector.tensorflow.org/>

Model Validation – Testing the App and Model together

- To ensure that your model behaves correctly in the context of your larger app, which you can do by using an existing version of your app to execute relevant parts of the host app's own test suite.
- Such testing can also help ensure that data schemas (input/output) and behaviors for all base cases in an application are sufficiently covered.
- As they mature, most organizations build a custom stack for this level of model validation.

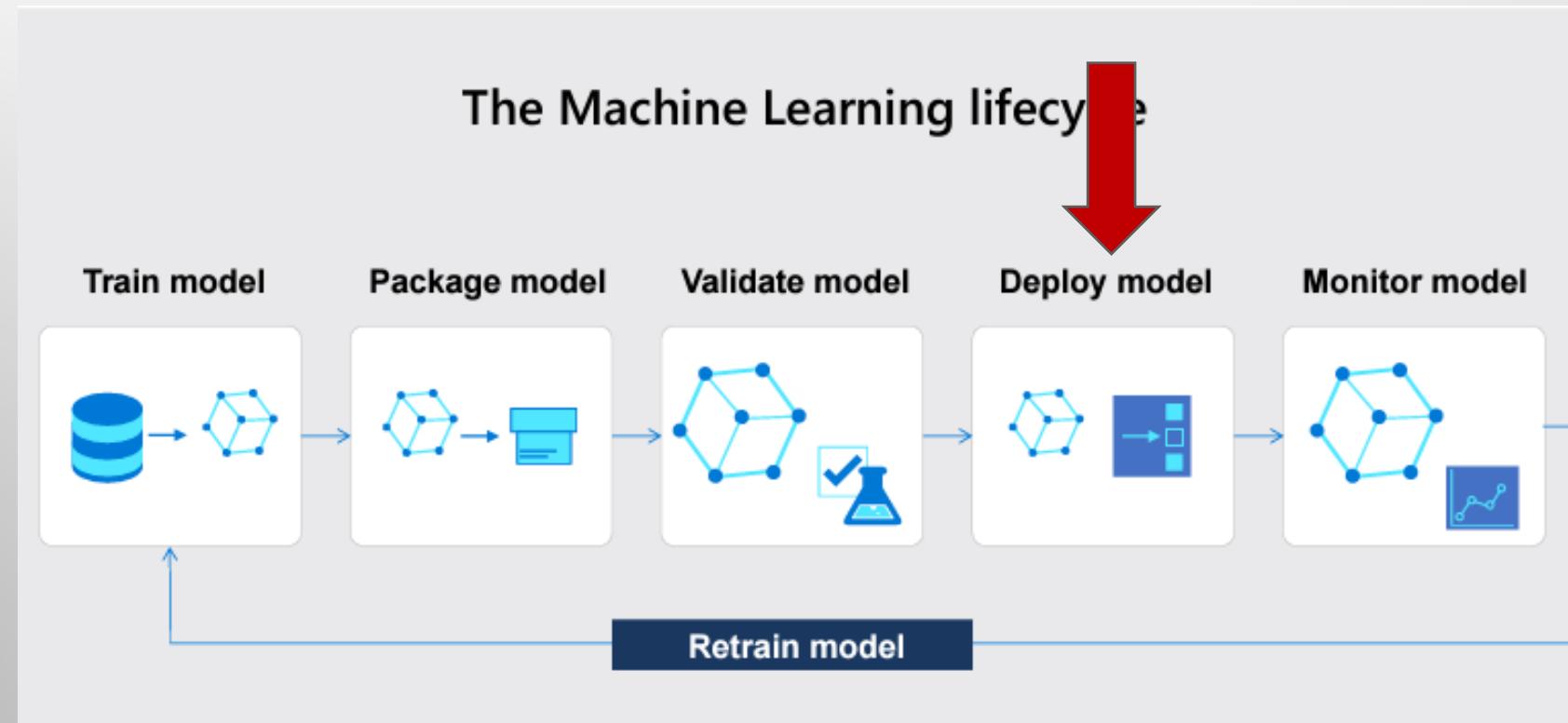
Pre-trained Models



The screenshot shows the homepage of the Hugging Face website. At the top, there is a navigation bar with the Hugging Face logo, a search bar, and links for Models, Datasets, Spaces, Docs, Solutions, Pricing, Log In, and Sign Up. A pink banner below the navigation bar encourages users to "Try out our NEW inference solution". The main content area features a large yellow emoji of a smiling face with hands clasped together. Below the emoji, the text "The AI community building the future." is displayed in a large, bold, black font. Underneath this, a smaller text reads: "Build, train and deploy state of the art models powered by the reference open source in machine learning." At the bottom of the main section, there is a button labeled "Star" with the number "71,458" next to it.

<https://huggingface.co/spaces?sort=likes>

Machine Learning Life-Cycle



Deep Learning Hardware

TRAINING



DGX-1 & DGX Station



RTX/GTX
series



Tesla A100
Tesla V100

INFERENCE

DATA CENTER



Tesla A100/V100



Tesla T4 / L4

AUTOMOTIVE



Drive AGX Orin

EMBEDDED

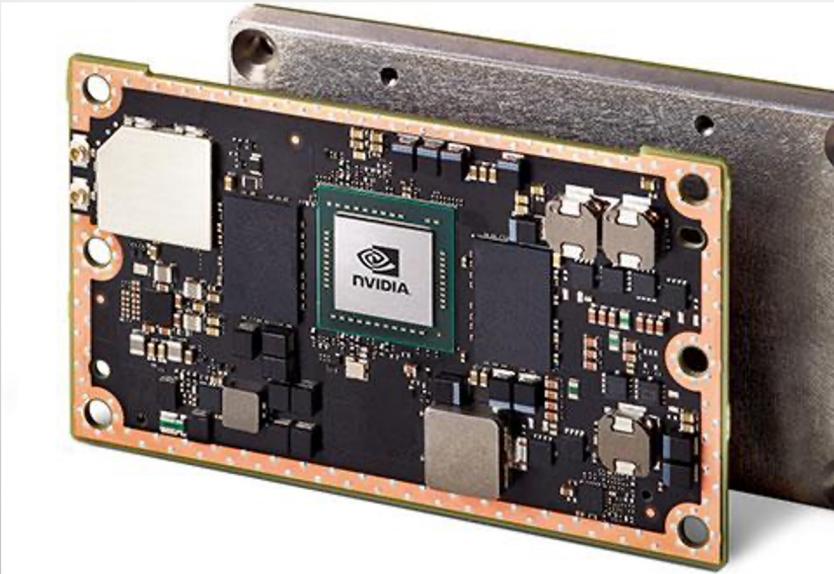


Jetson TX2



Jetson Orin

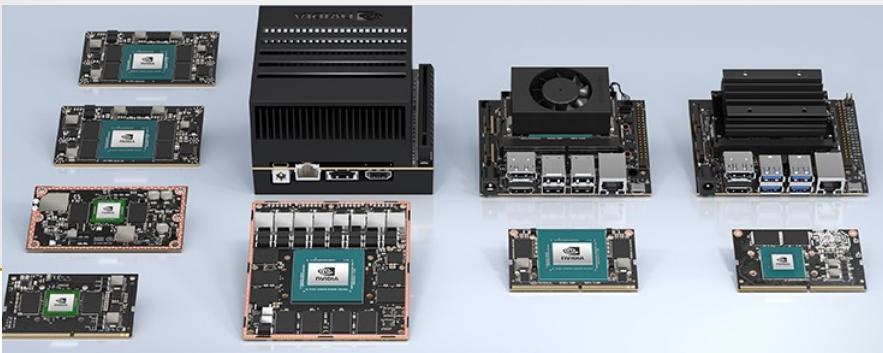
NVIDIA Jetson



	Jetson TX2	Jetson TX1
GPU	NVIDIA Pascal™, 256 CUDA cores	NVIDIA Maxwell™, 256 CUDA cores
CPU	HMP Dual Denver 2/2 MB L2 + Quad ARM® A57/2 MB L2	Quad ARM® A57/2 MB L2
Video	4K x 2K 60 Hz Encode (HEVC) 4K x 2K 60 Hz Decode (12-Bit Support)	4K x 2K 30 Hz Encode (HEVC) 4K x 2K 60 Hz Decode (10-Bit Support)
Memory	8 GB 128 bit LPDDR4 59.7 GB/s	4 GB 64 bit LPDDR4 25.6 GB/s
Display	2x DSI, 2x DP 1.2 / HDMI 2.0 / eDP 1.4	2x DSI, 1x eDP 1.4 / DP 1.2 / HDMI
CSI	Up to 6 Cameras (2 Lane) CSI2 D-PHY 1.2 (2.5 Gbps/Lane)	Up to 6 Cameras (2 Lane) CSI2 D-PHY 1.1 (1.5 Gbps/Lane)
PCIE	Gen 2 1x4 + 1x1 OR 2x1 + 1x2	Gen 2 1x4 + 1x1
Data Storage	32 GB eMMC, SDIO, SATA	16 GB eMMC, SDIO, SATA
Other	CAN, UART, SPI, I2C, I2S, GPIOs	UART, SPI, I2C, I2S, GPIOs
USB	USB 3.0 + USB 2.0	
Connectivity	1 Gigabit Ethernet, 802.11ac WLAN, Bluetooth	
Mechanical	50 mm x 87 mm (400-Pin Compatible Board-to-Board Connector)	

Mobile Devices : NVIDIA Jetson

	Jetson Nano	Jetson TX2 Series				Jetson Xavier NX Series		Jetson AGX Xavier Series			Jetson Orin NX	Jetson AGX Orin
		TX2 NX	TX2 4GB	TX2	TX2i	Jetson Xavier NX 16GB	Jetson Xavier NX	Jetson AGX Xavier 64GB	Jetson AGX Xavier	Jetson AGX Xavier Industrial		
AI Performance	472 GFLOPS	1.33 TFLOPS			1.26 TFLOPS	21 TOPS		32 TOPS		30 TOPS	100 TOPS	200 TOPS
GPU	128-core NVIDIA Maxwell™ GPU	256-core NVIDIA Pascal™ GPU				384-core NVIDIA Volta™ GPU with 48 Tensor Cores		512-core NVIDIA Volta GPU with 64 Tensor Cores			1024-core NVIDIA Ampere GPU with 32 Tensor Cores	2048-core NVIDIA Ampere GPU with 64 Tensor Cores
CPU	Quad-core ARM® Cortex®-A57 MPCore processor	Dual-core Denver 2 64-bit CPU and quad-core Arm Cortex-A57 MPCore processor				6-core NVIDIA Carmel Arm®v8.2 64-bit CPU 6MB L2 + 4MB L3		8-core NVIDIA Carmel Arm®v8.2 64-bit CPU 8MB L2 + 4MB L3		8-core NVIDIA Arm® Cortex A78AE v8.2 64-bit CPU 2MB L2 + 6MB L3	12-core NVIDIA Arm® Cortex A78AE v8.2 64-bit CPU 3MB L2 + 6MB L3	
DL Accelerator	-	-				2x NVDA		2x NVDA		2x NVDA v2	2x NVDA v2	
Vision Accelerator	-	-				2x PVA		2x PVA		1 x PVA v2	1 x PVA v2	
Safety Cluster Engine	-	-				-		-		2x Arm Cortex-R5 in lockstep	-	-
Memory	4GB 64-bit LPDDR4 25.6GB/s	4GB 128-bit LPDDR4 51.2GB/s	8GB 128-bit LPDDR4 59.7GB/s	8GB 128-bit LPDDR4 (ECC Support) 51.2GB/s	16GB 128-bit LPDDR4x 59.7GB/s	8GB 128-bit LPDDR4x 59.7GB/s	64GB 256-bit LPDDR4x 136.5GB/s	32GB 256-bit LPDDR4x 136.5GB/s	32GB 256-bit LPDDR4x (ECC support) 136.5GB/s	12GB 128-bit LPDDR5 102.4 GB/s	32GB 256-bit LPDDR5 204.8 GB/s	
Storage	16GB eMMC 5.1	16GB eMMC 5.1	32GB eMMC 5.1	32GB eMMC 5.1	16GB eMMC 5.1		32GB eMMC 5.1		64GB eMMC 5.1	- (Supports external NVMe)		64GB eMMC 5.1



Embedded Devices: Jetson AGX Orin

Figure: 1 Jetson AGX Orin delivers 6x the AI performance of Jetson AGX Xavier

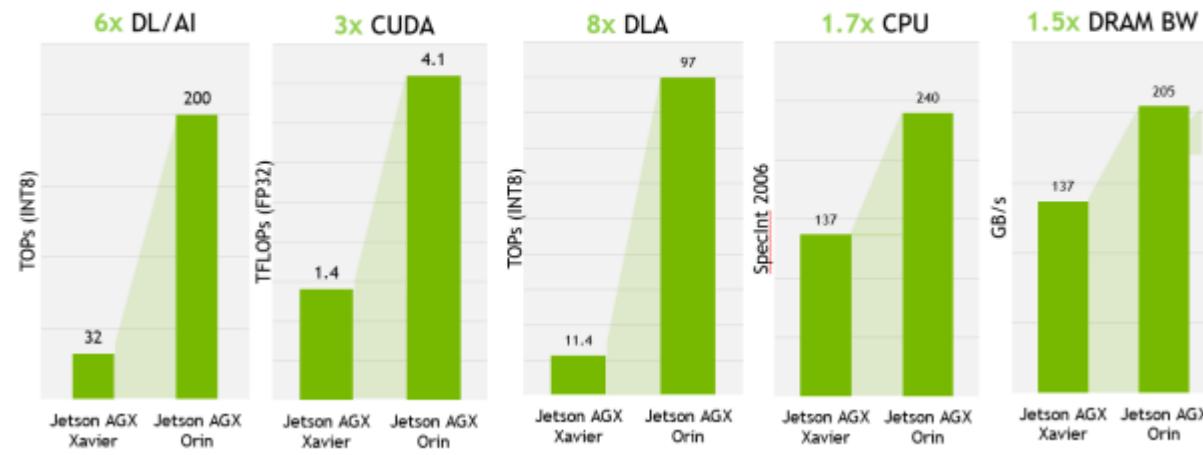


Figure 13: Jetson AGX Orin Developer Kit



Embedded Devices: Drive AGX Orin



DRIVE AGX Orin Developer Kit

Powered by a single Orin module delivering up to 254 TOPS

Includes hardware, software, and sample applications needed for the development of production-level autonomous vehicle applications

Built on production auto-grade silicon

Provides rich automotive I/O and flexibility to expand

Ready to order your DRIVE AGX Orin DevKit and Accessories for bench or in vehicle development:

- Contact Arrow (NVIDIA DRIVE Worldwide Distributor) at nvidia.drive@arrow.com
- Alternately, contact your NVIDIA representative (or [contact us](#))
- **NOTE:** DRIVE Hyperion DevKits are only available direct from NVIDIA

Component	Description	Details
One Orin SoC	Deep Learning Accelerators (DLA)	87 TOPS (INT8)
	NVIDIA Ampere architecture-class integrated GPU	167 TOPS (INT8) 5.2 TOPS (FP32)
	Image Signal Processor (ISP)	1.85Gigapixels/s
DRIVE AGX System I/O	Camera	90Gb/s over 16x GMSL(R) ports
	LIDAR/Radar	~30Gb/s over Ethernet
	Vehicle IO	6 CAN interfaces

Embedded Devices: Drive End-to-End Solutions for AVs



AV Development Platform

DRIVE Hyperion

DRIVE AGX Developer Kits and Sensor Suite

[Learn More](#)



Modular Software Stack

DRIVE SDK

DRIVE OS, DriveWorks, DRIVE AV, and DRIVE IX

[Learn More](#)



Simulation Platform

DRIVE Sim

AV Development and Validation Tools built on NVIDIA Omniverse

[Learn More](#)



DNN Training Platform

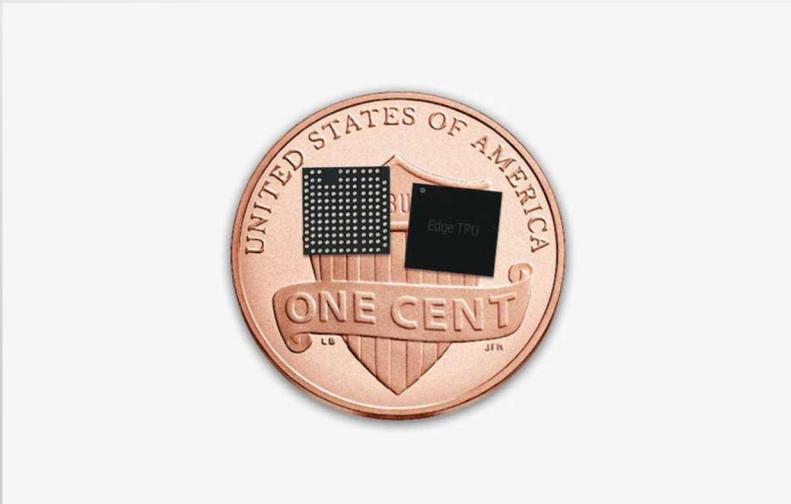
NVIDIA DGX

DGX and Deep Learning SDK

[Learn More](#)

Edge TPU

- Edge TPU: a small ASIC designed by Google that provides high performance ML inferencing for low-power devices.
- It can execute state-of-the-art mobile vision models such as MobileNet V2 at 100+ fps, in a power efficient manner.
- Supports Tensorflow Lite.



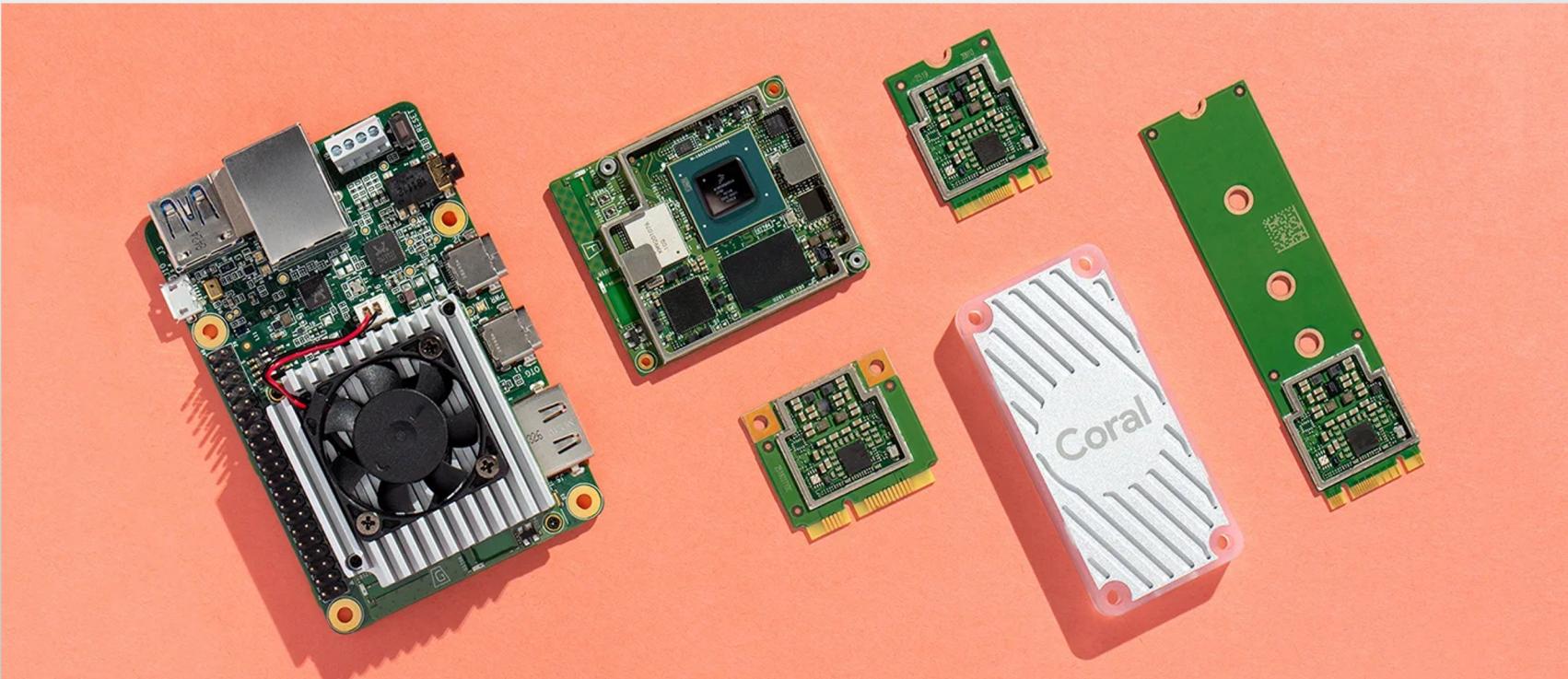
Two Edge TPU chips on a US penny



USB Accelerator with Edge TPU

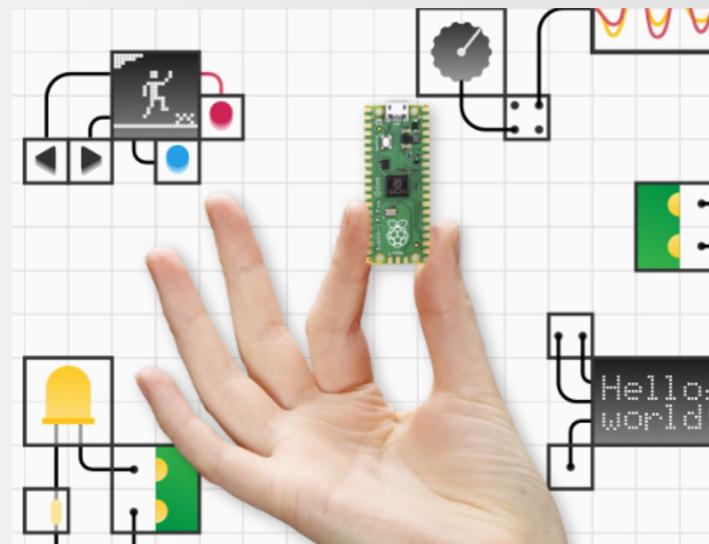
Edge TPU – Coral Toolkit

- Coral is a complete toolkit to build products with local AI.
- Prototyping devices include a single-board computer and USB accessory
- Production-ready devices include a system-on-module and PCIe module.



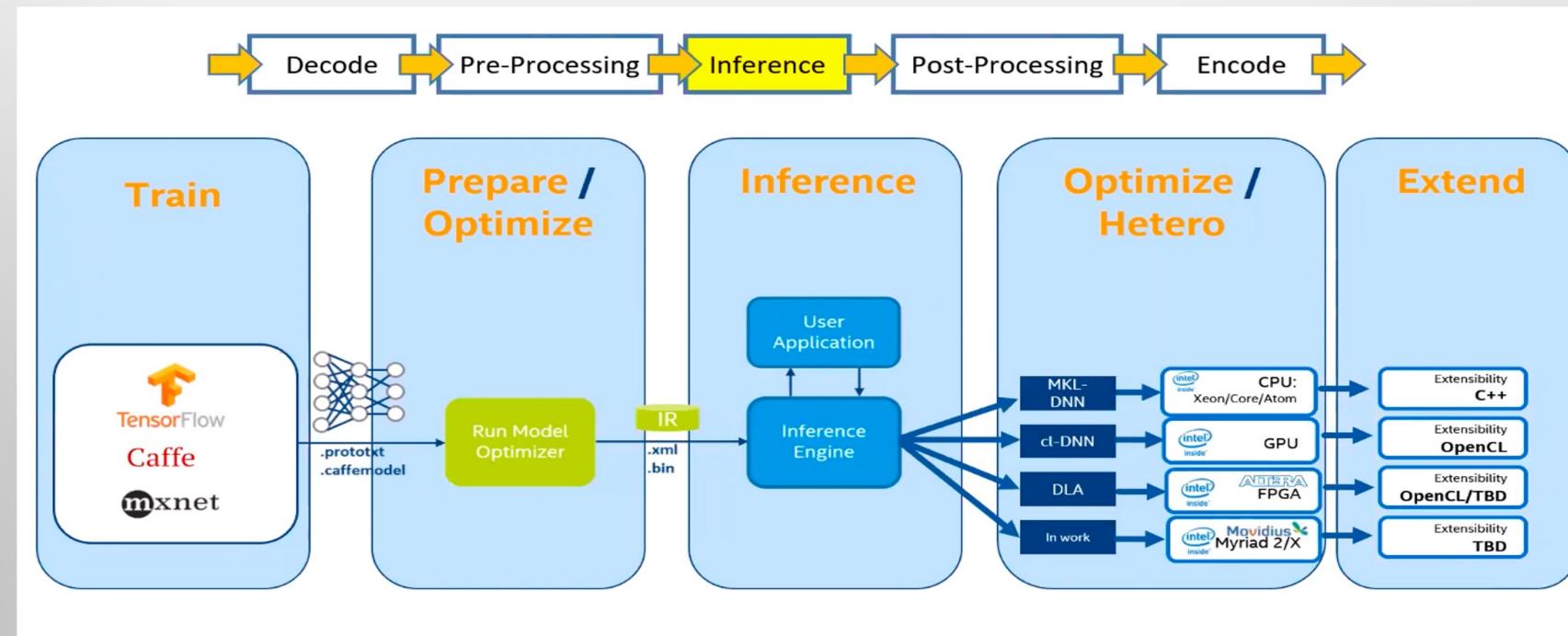
IoT Devices – Raspberry Pi

- Supports TensorFlow Lite
- Models can even be run on 4\$ Pi Pico
- This device has
 - two-core Arm Cortex-M0+ CPU
 - 264KB RAM
 - up to 16MB off-chip Flash memory



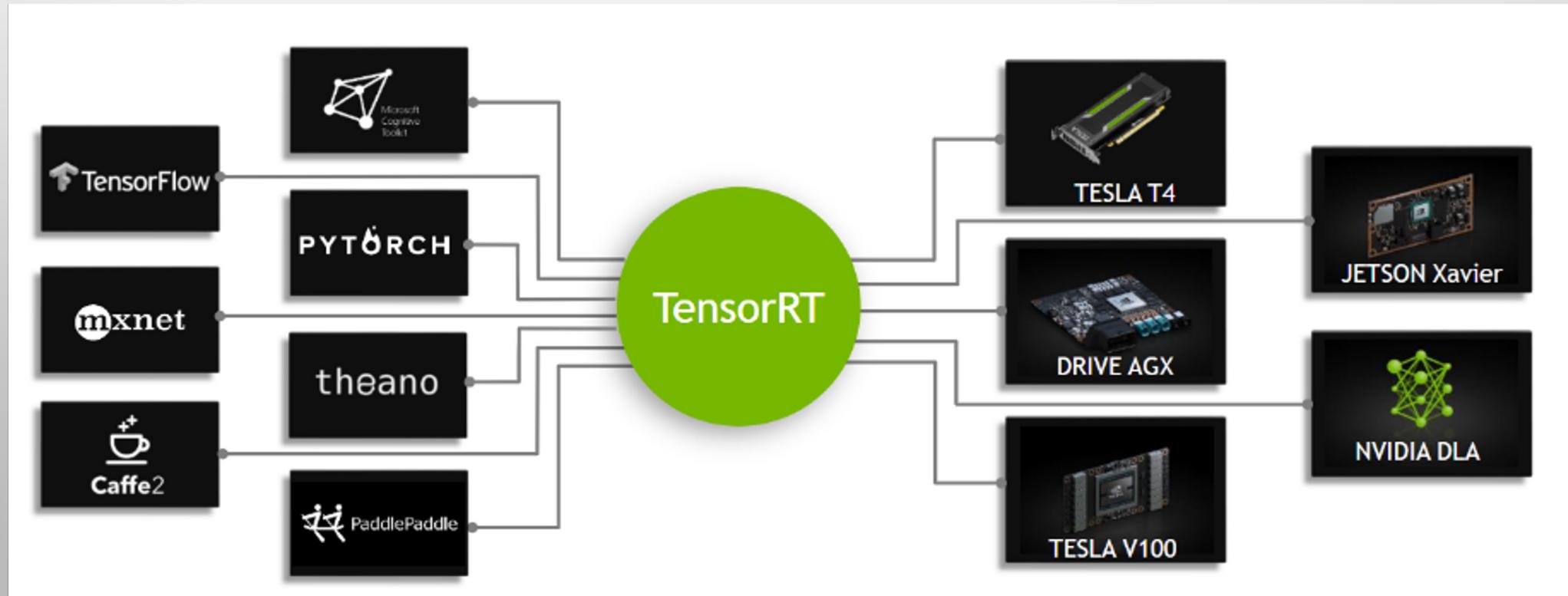
OpenVINO: Open Visual Inference and Neural Network Optimization

- OpenVINO is a toolkit developed by Intel for accelerated inference
- Allows acceleration on CPU, GPU, Intel® Movidius™ Neural Compute Stick and FPGA
- Supports various deep learning frameworks



NVIDIA TensorRT

- Trained model is fed into TensorRT for optimization
- As it accepts ONNX input, works with all platforms having ONNX support
- Output are optimized for the target platform and can directly be run on various target platforms by TensorRT Runtime



Machine Learning Life-Cycle

