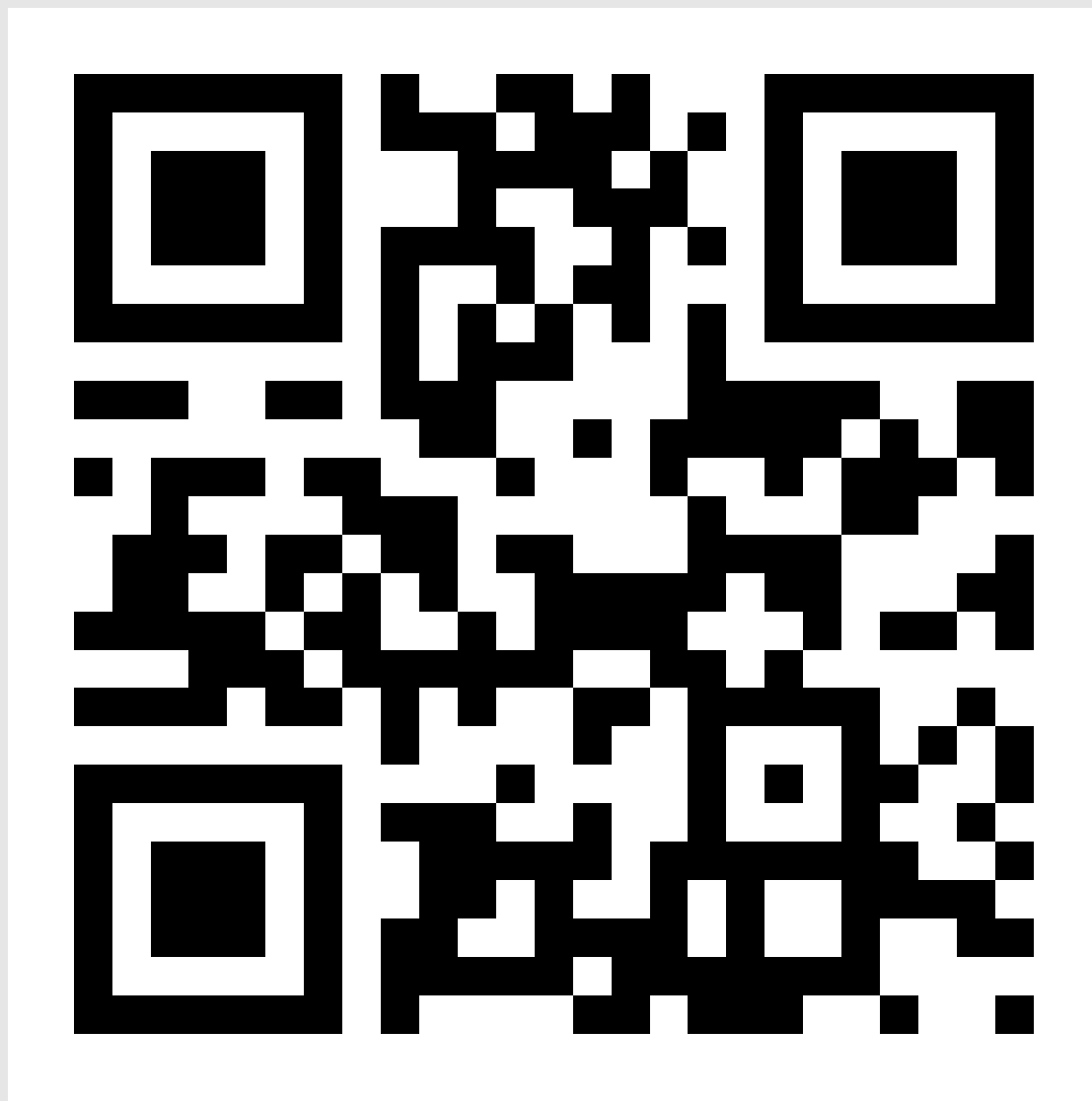


Leverage Power of Machine Learning with ONNX

Ron Dagdag
@rondagdag






<http://bit.ly/ml-onnx>

Hackster Portfolio

www.dagdag.net

@rondagdag



Ron Dagdag
Dad / Lead Software Engineer / 3D Developer / Tax Return Preparer.
Passionate to learn about Robotics, VR, AR, Artificial Intelligence, IOT
@rondagdag
FORT WORTH, United States
Team [Augmented Reality](#)
Team [Virtual Reality](#)

INTERNET OF "KINECT" 2,443 16 44
IoT Gateway Azure Event Hub Stream Analytics Production Base
Posture Recognition using K...
Ron Dagdag

Easy 60 0
Littlebits Arduino Keyboard ...
Ron Dagdag

Intermediate 701 9
Alexa, tell Echobot to fly
Ron Dagdag

Advanced 1,345 12
Hello World!
Control your "Earth Rover" i...
Ron Dagdag

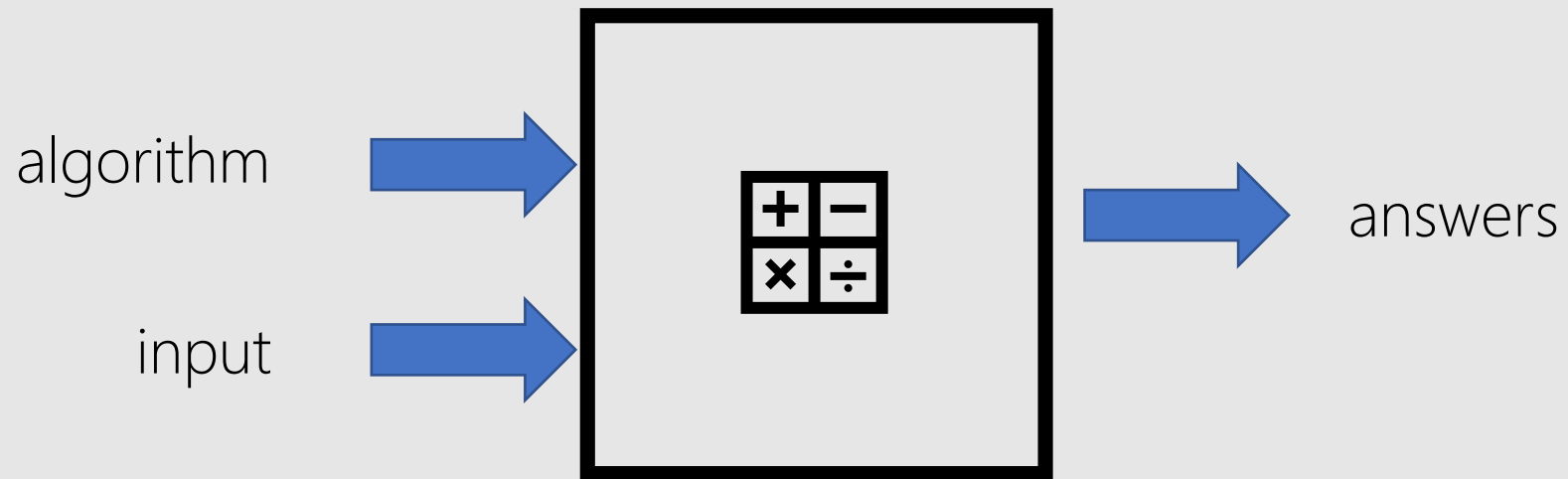
Advanced 2,256 30
ConstructAR - The Holograp...
TEAM ConstructAR

Intermediate 449 4
Color Changing Fireworks in...
Ron Dagdag

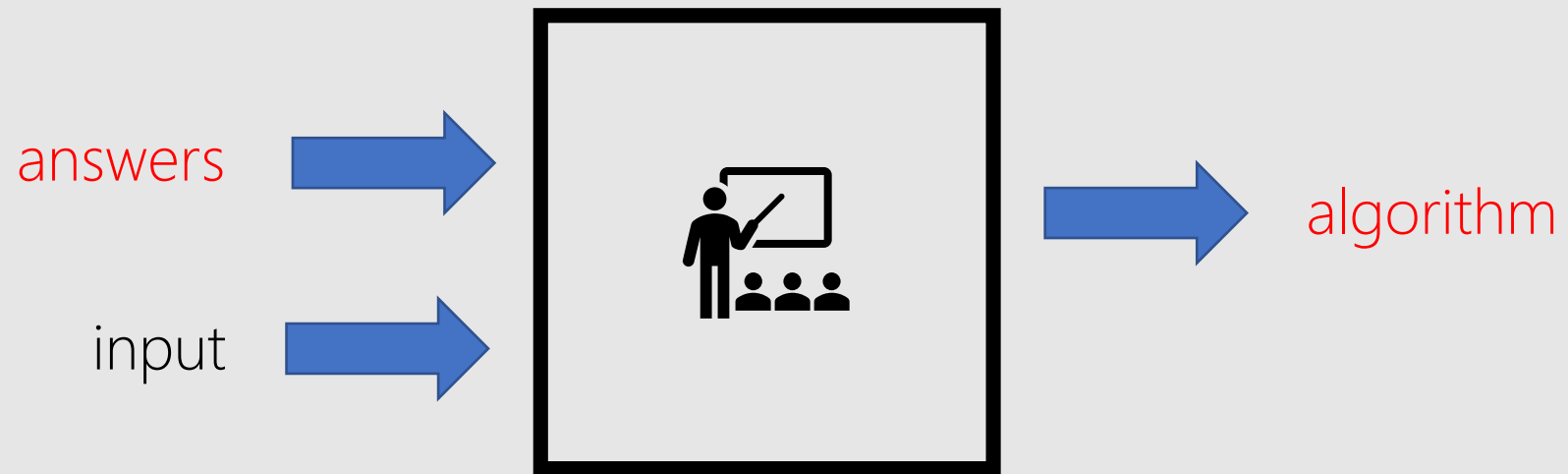
ONNX, Not ONIX



programming



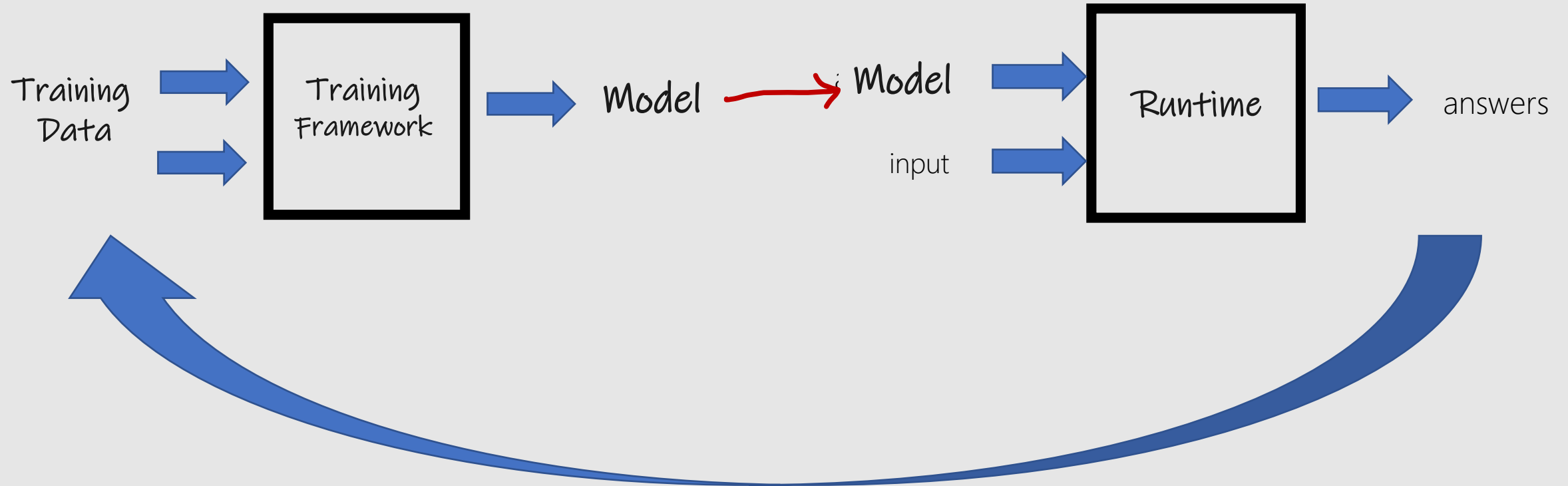
machine learning



ML Primer

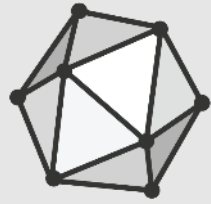
Machine learning

Inferencing



Open and Interoperable AI





ONNX

Open Neural Network Exchange

Open format for ML models

github.com/onnx

onnx.ai/



ONNX Partners



Agenda

- ✓ What is ONNX
- ☐ How to create ONNX models
- ☐ How to deploy ONNX models

Create

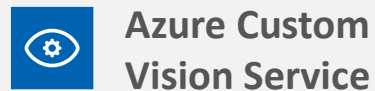
Frameworks



Native support

Converters

Services



Native support

ONNX Model

Deploy

Cloud Services

Azure Machine Learning services

Ubuntu VM

Windows Server 2019 VM

Windows Devices

IoT Edge Devices

Other Devices
(iOS, Android, etc)

Native support

Converters

Frameworks



**Step 1:
Create**

Services



Native support

Converters

Native support



ONNX Model

Azure

Azure Machine Learning services

Ubuntu VM

Windows Server 2019 VM



**Step 2:
Deploy**

Other Devices
(iOS, etc)

Native support

Converters

A still life composition featuring several brown eggs in a cardboard carton, a pair of wire-rimmed glasses, and an open book. The text "Secret Recipe" is overlaid in white, with a vertical line separating the word "Secret" from "Recipe".

Secret Recipe

4 ways to get an ONNX model



ONNX Model Zoo



Azure Custom Vision Service



Convert existing models



Train models in Azure Machine Learning

Automated Machine Learning

ONNX Model Zoo: github.com/onnx/models

Image Classification

This collection of models take images as input, then classifies the major objects in the images into a set of predefined classes.

Model Class	Reference	Description
MobileNet	Sandler et al.	Efficient CNN model for mobile and embedded vision applications. Top-5 error from paper - ~10%
ResNet	He et al., He et al.	Very deep CNN model (up to 152 layers), won the ImageNet Challenge in 2015. Top-5 error from paper - ~3.6%
SqueezeNet	Iandola et al.	A lightweight CNN model with fewer layers and parameters. Top-5 error from paper - ~4.8%
VGG	Simonyan et al.	Deep CNN model, won the ImageNet Challenge in 2014. Top-5 error from paper - ~7.4%

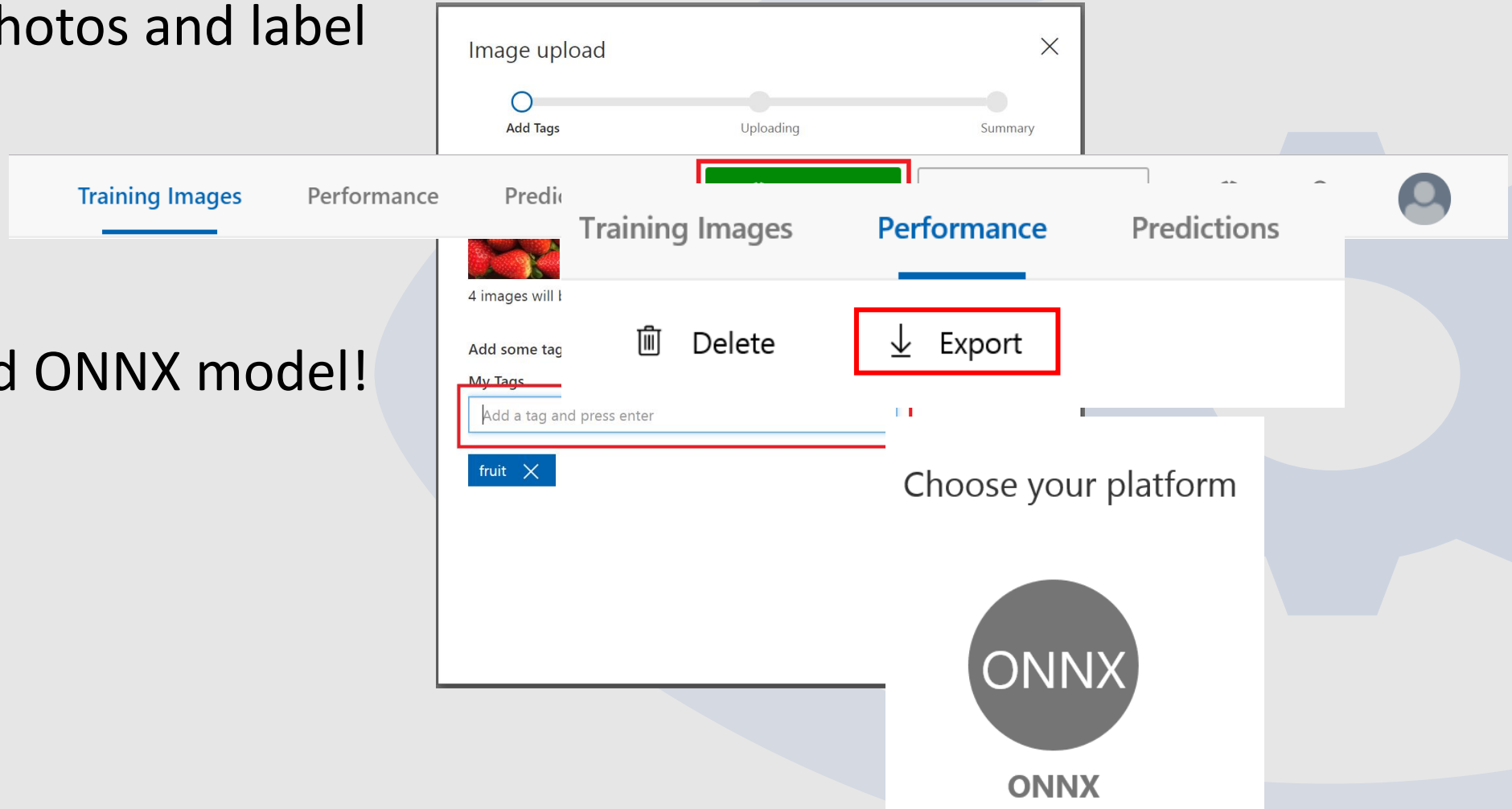
Model	Download	Checksum	Download (with sample test data)	ONNX version	Opset version	Top-1 accuracy (%)	Top-5 accuracy (%)
ResNet-18	44.6 MB	MD5	42.9 MB	1.2.1	7	69.70	89.49
ResNet-34	83.2 MB	MD5	78.6 MB	1.2.1	7	73.36	91.43
ResNet-50	97.7 MB	MD5	92.0 MB	1.2.1	7	75.81	92.82
ResNet-101	170.4 MB	MD5	159.4 MB	1.2.1	7	77.42	93.61
ResNet-152	230.3 MB	MD5	216.0 MB	1.2.1	7	78.20	94.21

Custom Vision Service: customvision.ai

1. Upload photos and label

2. Train

3. Download ONNX model!



Convert
models



Convert models

1. Load existing model
2. (Convert to ONNX)
3. Save ONNX model

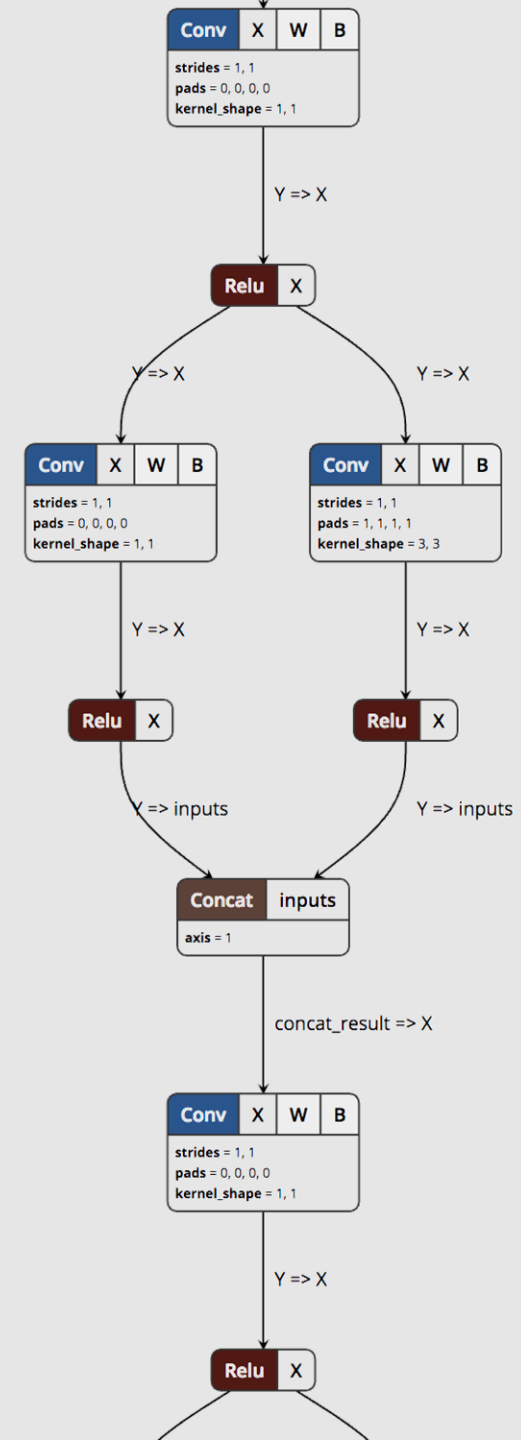


ONNX Models

Graph of operations

Netron

<https://lutzroeder.github.io/netron/>




Convert models: Keras

```
import numpy as np
from keras.preprocessing import image
from keras.applications.resnet50 import preprocess_input
import keras2onnx
import onnxruntime

# load keras model
from keras.applications.resnet50 import ResNet50
model = ResNet50(include_top=True, weights='imagenet')

# convert to onnx model
onnx_model = keras2onnx.convert_keras(model, model.name)
```



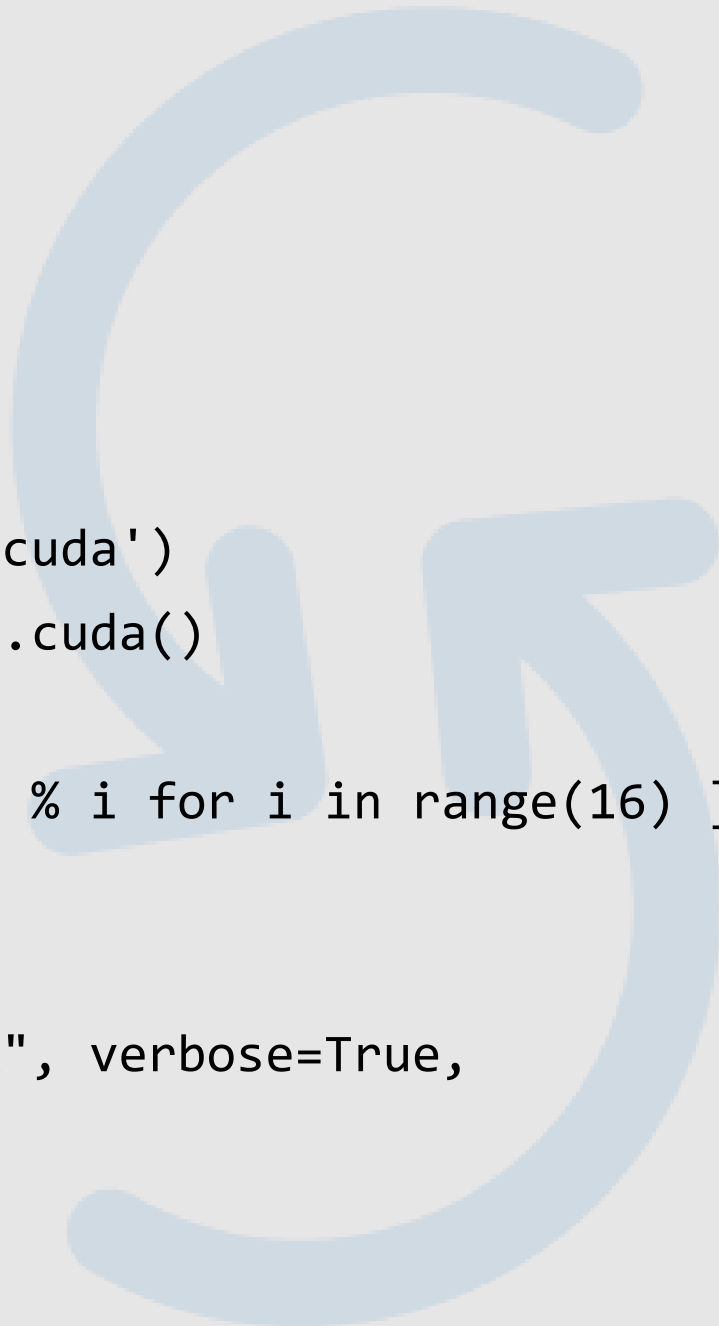
Convert models: Pytorch

```
import torch
import torchvision

dummy_input = torch.randn(10, 3, 224, 224, device='cuda')
model = torchvision.models.alexnet(pretrained=True).cuda()

input_names = [ "actual_input_1" ] + [ "learned_%d" % i for i in range(16) ]
output_names = [ "output1" ]

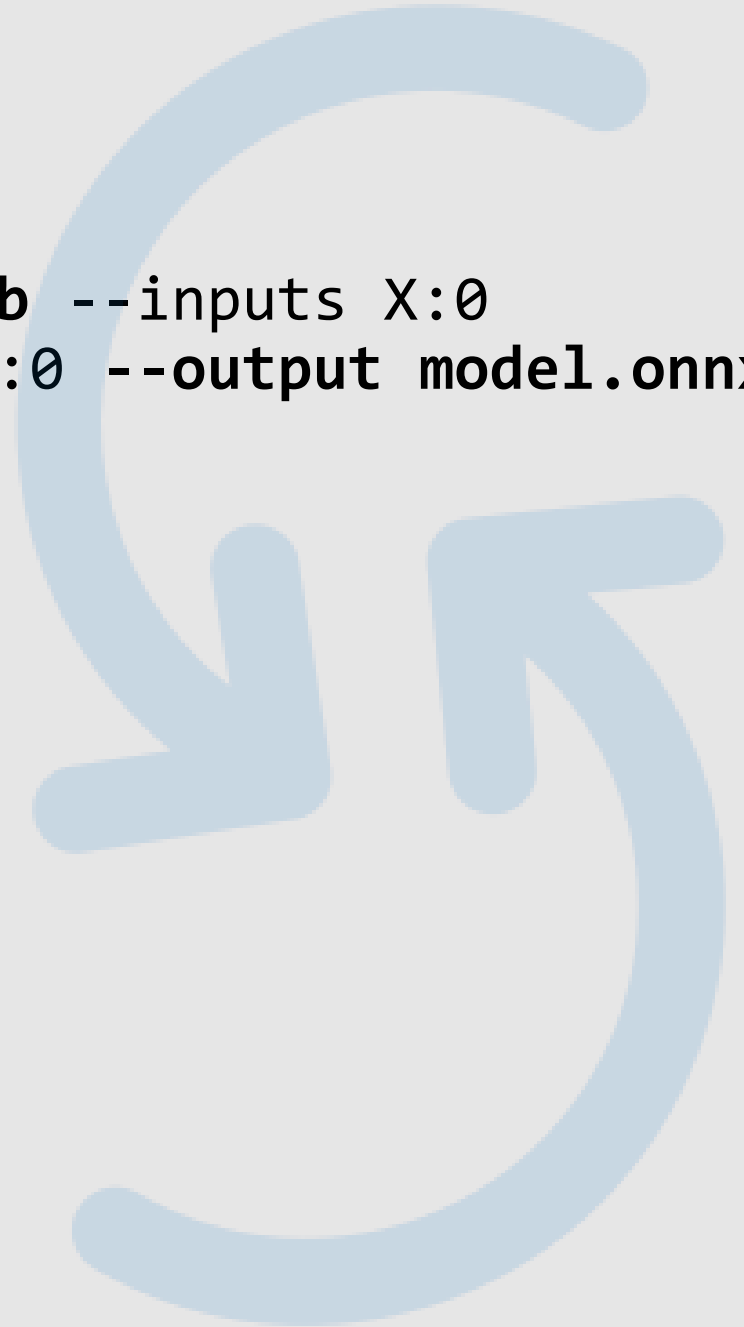
torch.onnx.export(model, dummy_input, "alexnet.onnx", verbose=True,
input_names=input_names, output_names=output_names)
```

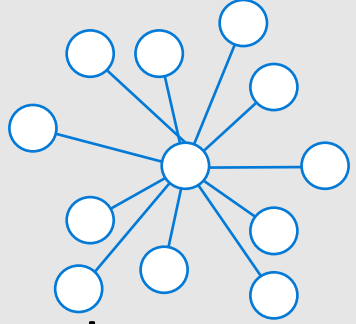


Convert models: TensorFlow

```
python -m tf2onnx.convert --input frozen.pb --inputs X:0  
--outputs output:0 --output model.onnx
```

<https://github.com/onnx/tensorflow-onnx>



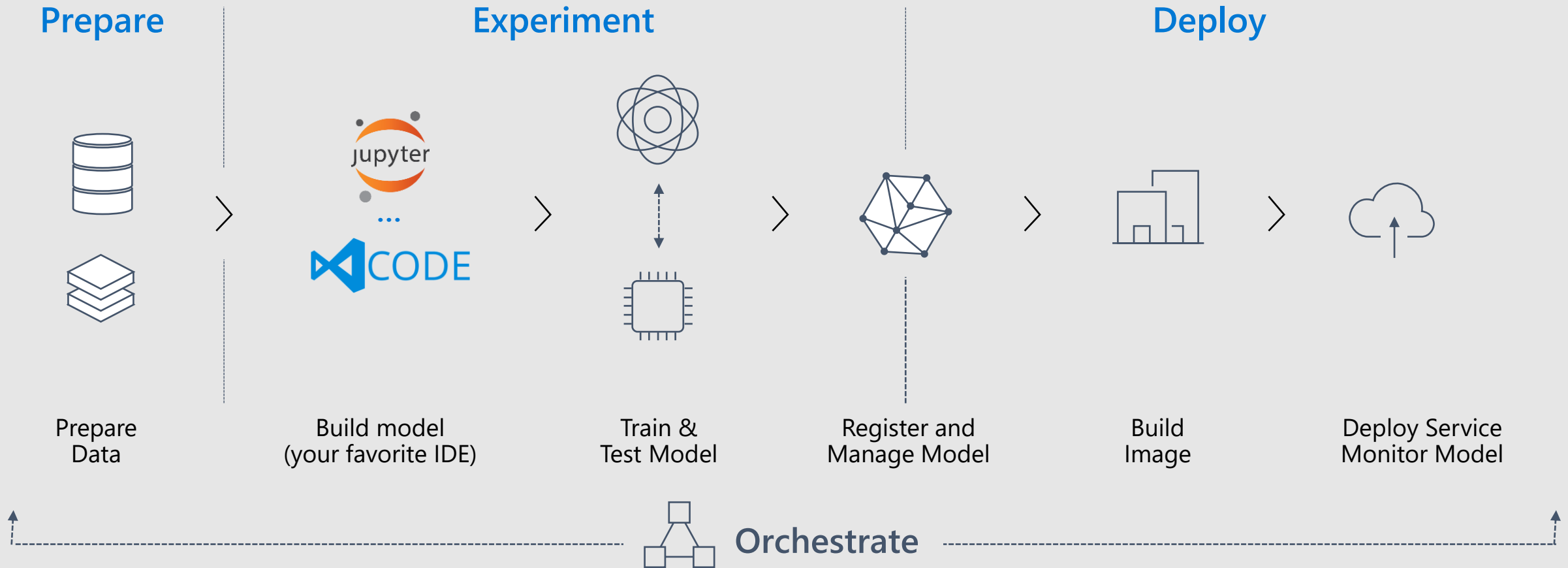


Train models in Azure Machine Learning

- Experiment locally then quickly scale with GPU clusters in the cloud
- Use automated machine learning and hyper-parameter tuning.
- Keeping Track of experiments, manage models, and easily deploy with integrated CI/CD tooling

Machine Learning

Typical E2E Process



high
dimensional
matrices

tensor

't'
'e'
'n'
's'
'o'
'r'

tensor of dimensions [6]
(vector of dimension 6)

3	1	4	1
5	9	2	6
5	3	5	8
9	7	9	3
2	3	8	4
6	2	6	4

tensor of dimensions [6,4]
(matrix 6 by 4)

2	1	2	1
2	4	9	4
2	5	6	2
7	7	3	2

tensor of dimensions [4,4,2]

Frameworks

 **Caffe2**  **Chainer**  **Cognitive Toolkit**

 **mxnet**  **PyTorch**  **PaddlePaddle**

ML.NET  **MathWorks**  **dmlc XGBoost**

Step 1: Create

Services



Azure Custom
Vision Service

Native
support

Converters

Native
support



ONNX Model

Azure

Azure Machine Learning services

Ubuntu VM

Windows Server 2019 VM




Step 2: Deploy

Other Devices
(iOS, etc)

Native
support

Converters

A baker in a white shirt and apron is shown from the waist down, working on a wooden table. The table is covered with a thick layer of white flour. The baker is shaping a large, round loaf of bread. The background is a plain, light-colored wall.

Baker vs Starting a Bakery

Create

Frameworks



Native support

Converters

Services



Native support

ONNX Model

Deploy

Azure

Azure Machine Learning services

Ubuntu VM

Windows Server 2019 VM

Windows Devices

IoT Edge Devices

Other Devices
(iOS, etc)

Native support

Converters

A person's hands are visible, holding a large, round, rustic loaf of bread. The bread has a thick, golden-brown crust with some darker, caramelized spots. It is wrapped in a blue and white striped cloth, which is tied around the middle. The background is a blurred wooden surface.

Cloud or Edge



Cloud
or
Edge

Deploy with Azure Machine Learning

- Model management services
- Deploy as web service to ACI or AKS
- Capture model telemetry



Azure
Machine Learning

Machine Learning

Typical E2E Process

Prepare



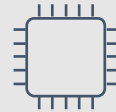
Prepare
Data



Experiment



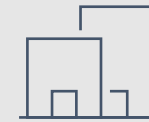
Build model
(your favorite IDE)



Train &
Test Model



Register and
Manage Model



Build
Image



Deploy Service
Monitor Model

Deploy

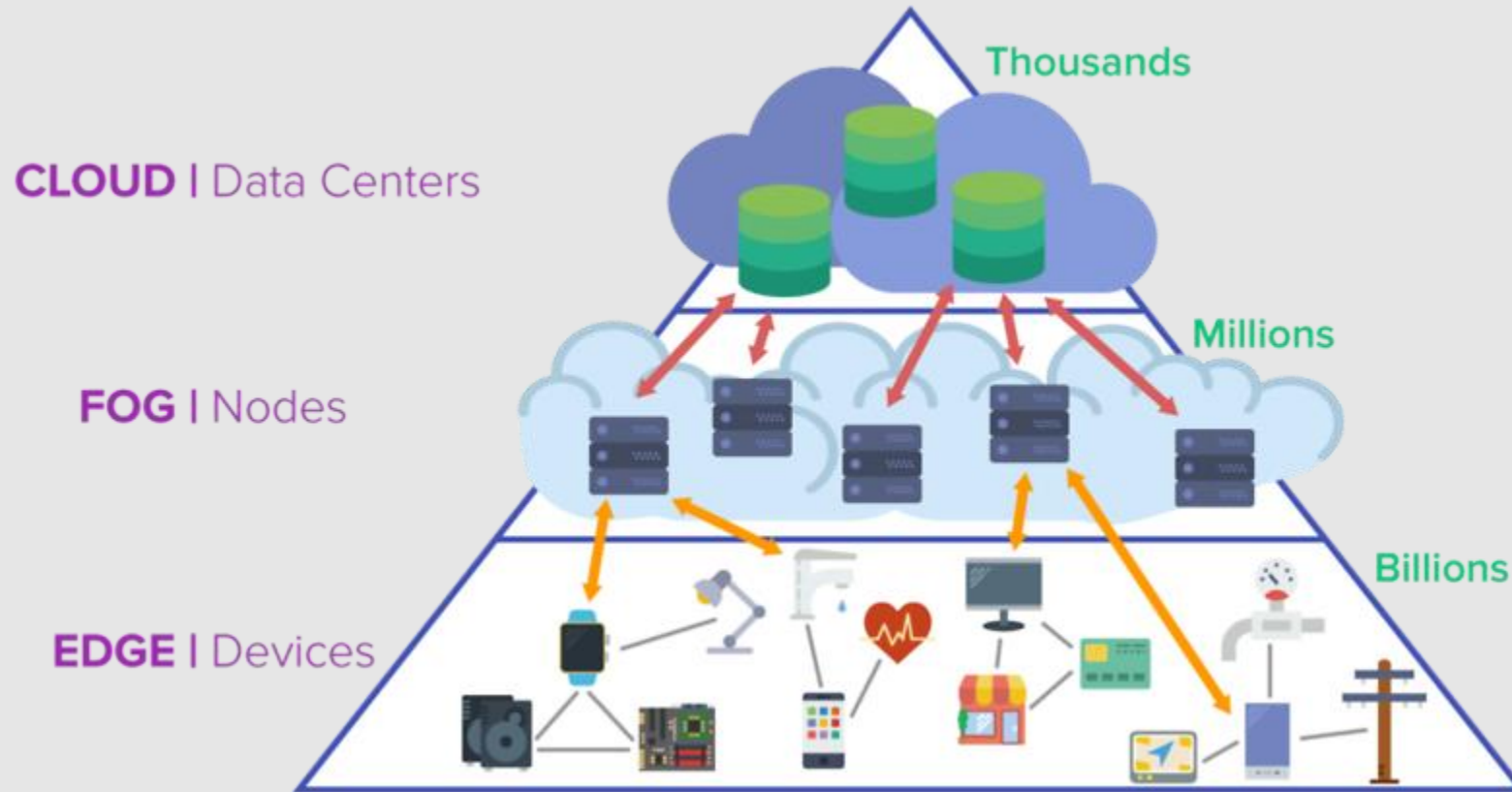




Demo

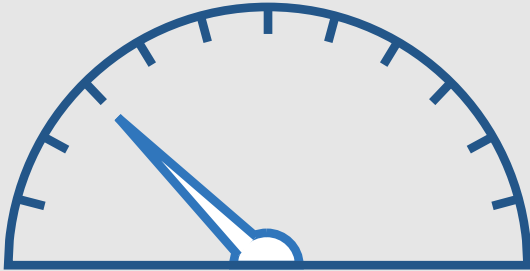
<https://github.com/rondagdag/onnx-pected/tree/master/ONNX-AML>

What is the Edge?

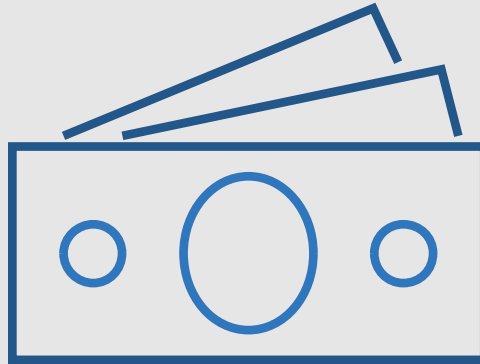


[Imagimob AB](#)

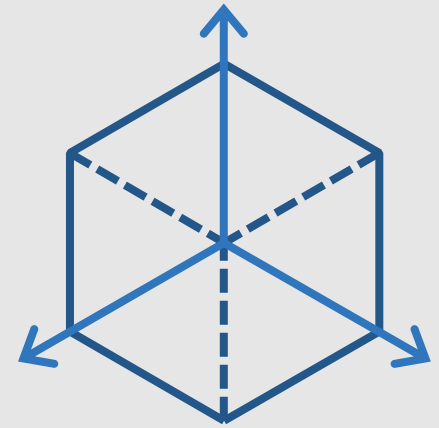
AI on the edge



Low latency



Scalability



Flexibility

ONNX as an intermediary format

- **Convert to Tensorflow for Android**
 - [Convert a PyTorch model to Tensorflow using ONNX](#)
- **Convert to CoreML for iOS**
 - <https://github.com/onnx/onnx-coreml>
- **Fine-tuning an ONNX model with MXNet/Gluon**
 - https://mxnet.apache.org/versions/master/tutorials/onnx/fine_tuning_gluon.html



Deploy to Windows Devices

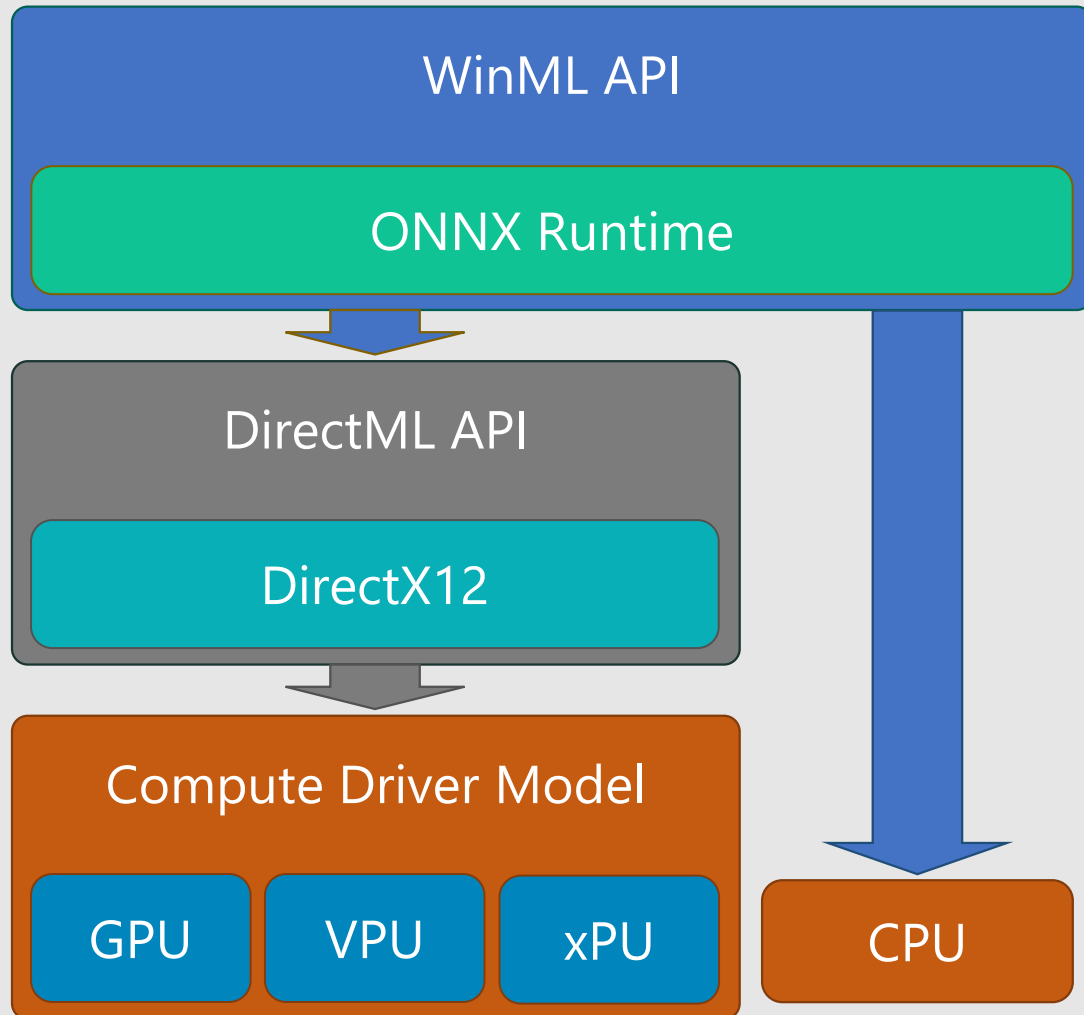
Windows ML

- Available across Windows family of devices
- Hardware abstraction via DirectML
- Unified API for Win32 and WinRT
- Optimized for performance
- Virtualization ready



Windows®

Windows AI platform



- WinML
 - **Practical**, simple model-based API for ML inferencing on Windows
- DirectML
 - **Realtime, high control** ML operator API; part of DirectX family
- Compute Driver Model
 - Robust **hardware reach**/abstraction layer for compute and graphics silicon



Demo

<https://github.com/rondagdag/onnx-pected/tree/master/GenerateONNX-AutoML>

ONNX Runtime

- High performance runtime for ONNX models
- Supports full ONNX-ML spec (currently v1.2+)
- Extensible architecture to plug-in hardware accelerators
- Simple Python API



ONNX

ONNX Runtime



ONNX

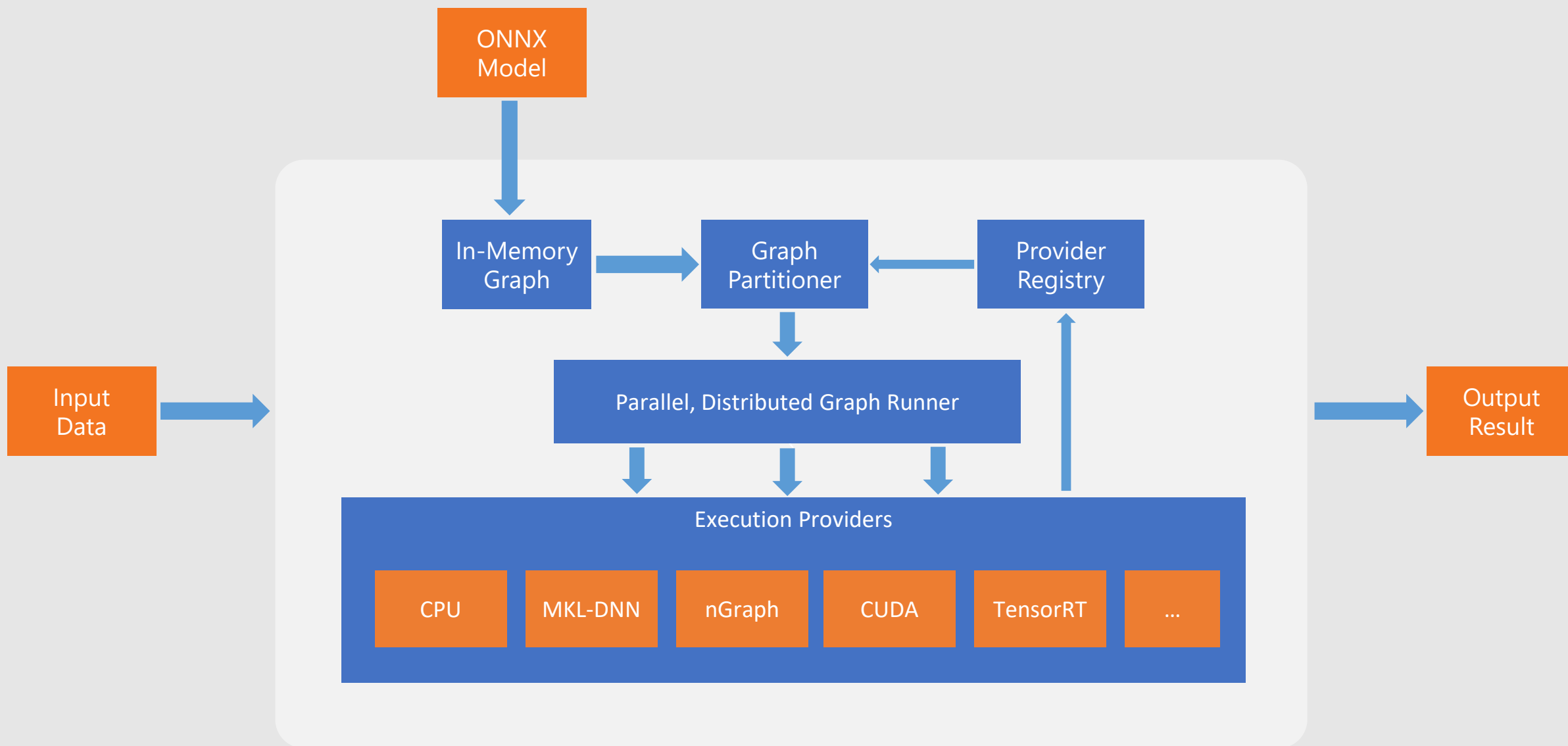
Get Started Easily

Select your requirements and use the resources provided to get started quickly

OS	Windows		Linux		Mac		
Language	Python (3.5-3.7)	C++	C#	C	Java		
Architecture	X64		X86		ARM64		ARM32
Hardware Acceleration	Default CPU	CUDA		TensorRT	DirectML		MKL-DNN
	MKL-ML		nGraph		NUPHAR		OpenVINO
Installation Instructions	Install Nuget package Microsoft.ML.OnnxRuntime.Gpu						

ONNX Runtime

- Microsoft services have seen an **average 2x performance gain on CPU**
- Office team saw a **14.6x reduction in latency** for a grammar checking model (thousands of queries per minute)
- Azure Cognitive Services saw a **3.5x reduction in latency** for an optical character recognition (OCR) model
- Bing QnA saw a **2.8x reduction in latency** for a model that generates answers to questions
- Bing Visual Search saw a **2x reduction in latency** for a model that helps identify similar images



ONNX Runtime - Python API

```
import onnxruntime
```

```
session = onnxruntime.InferenceSession("mymodel.onnx")
```

```
results = session.run([], {"input": input_data})
```



ONNX



ONNX Docker Image

- [onnx-docker-cpu](#): Image with ONNX, PyTorch, Tensorflow support
- [onnx-docker-gpu](#): Image with ONNX, PyTorch (CUDA), Caffe2 support
- [onnx-ecosystem](#): Jupyter notebook environment for getting started quickly with ONNX models, ONNX converters, and inference using ONNX Runtime.

Reference implementation to use ONNX Runtime with Azure IoT Edge



- <https://github.com/Azure-Samples/onnxruntime-iot-edge>



ONNX

ONNX.js

- ONNX.js is a JavaScript library for running ONNX models on browsers and on Node.js.
- ONNX.js has adopted Web Assembly and WebGL technologies
- optimized ONNX model inference runtime for both CPUs and GPUs.

<https://github.com/microsoft/onnxjs>



ONNX

ONNX.js

Compatibility

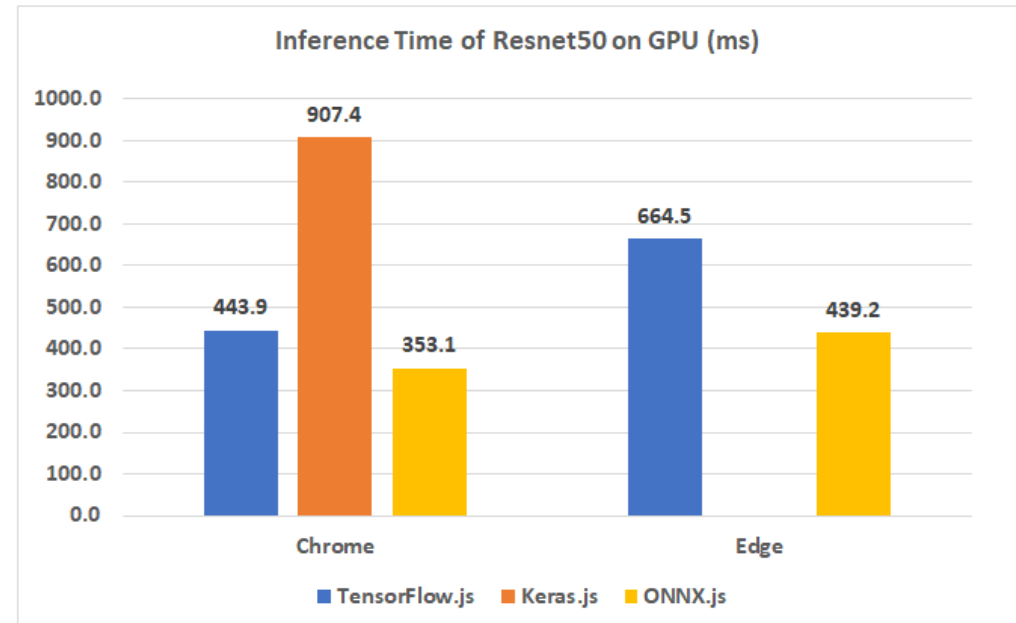
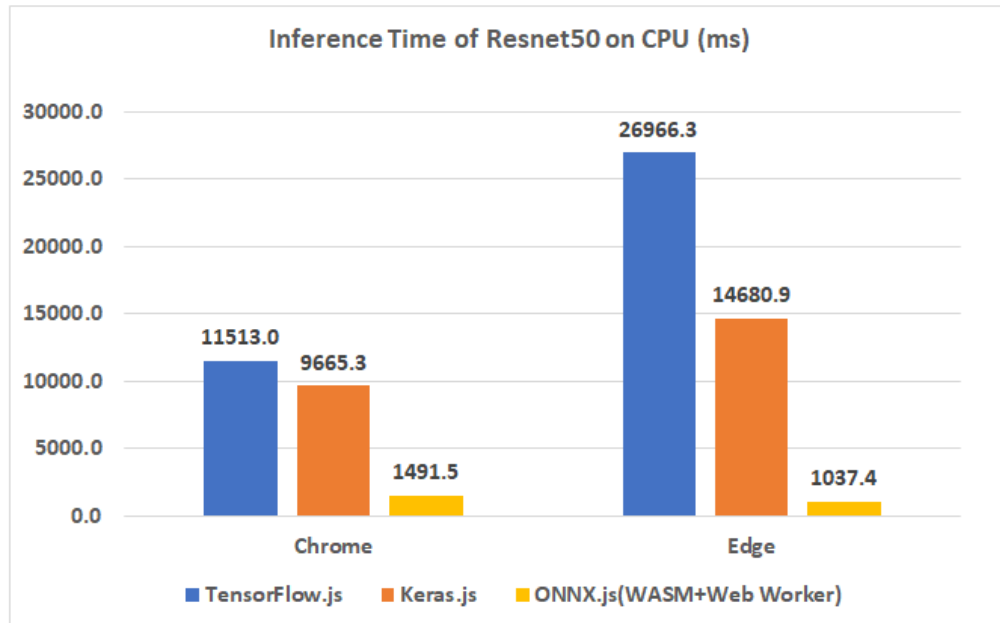
Desktop Platforms

OS/Browser	Chrome	Edge	FireFox	Safari	Opera	Electron	Node.js
Windows 10	✓	✓	✓	-	✓	✓	✓
macOS	✓	-	✓	✓	✓	✓	✓
Ubuntu LTS 18.04	✓	-	✓	-	✓	✓	✓

Mobile Platforms

OS/Browser	Chrome	Edge	FireFox	Safari	Opera
iOS	✓	✓	✓	✓	✓
Android	✓	✓	Coming soon	-	✓

ONNX.js



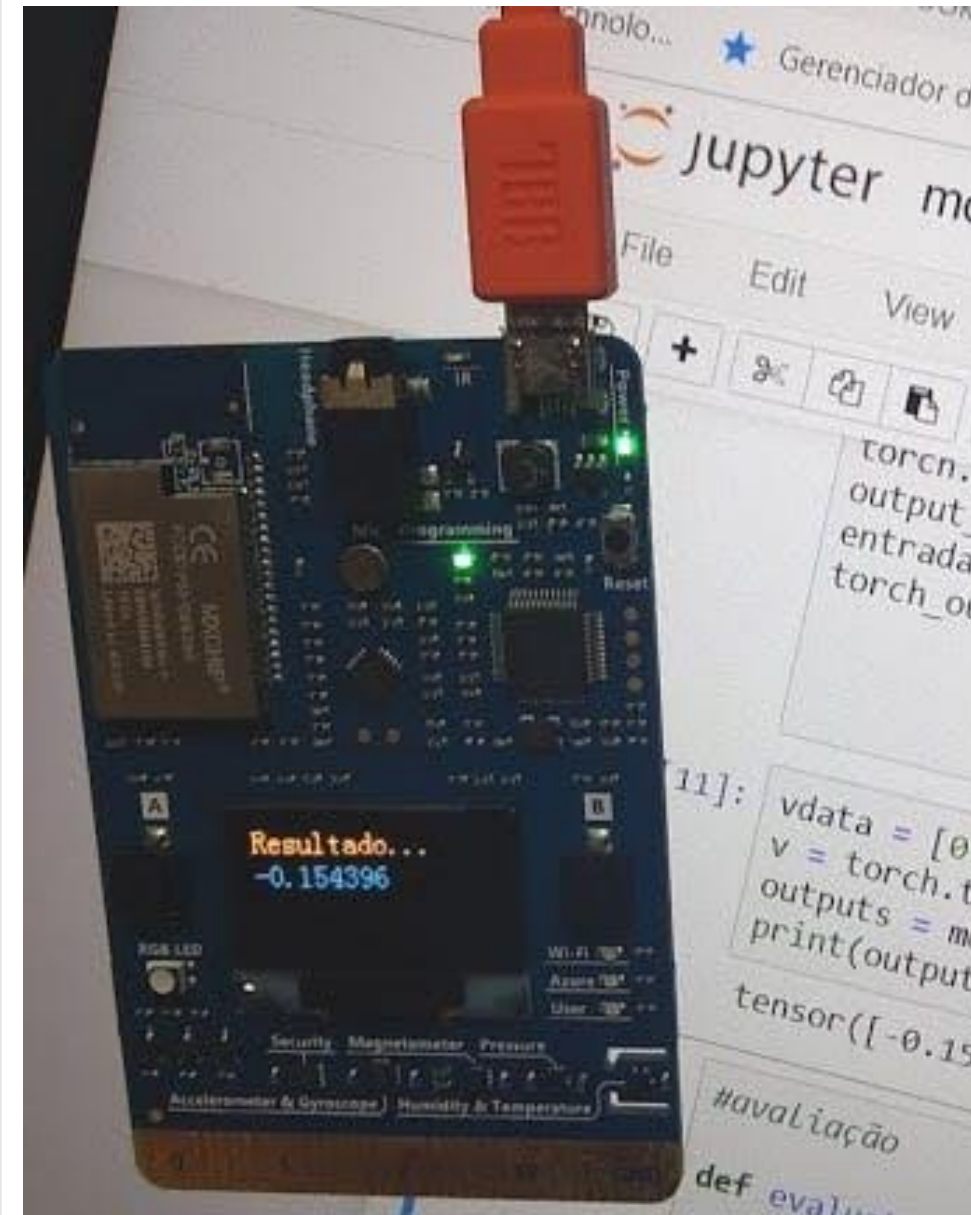
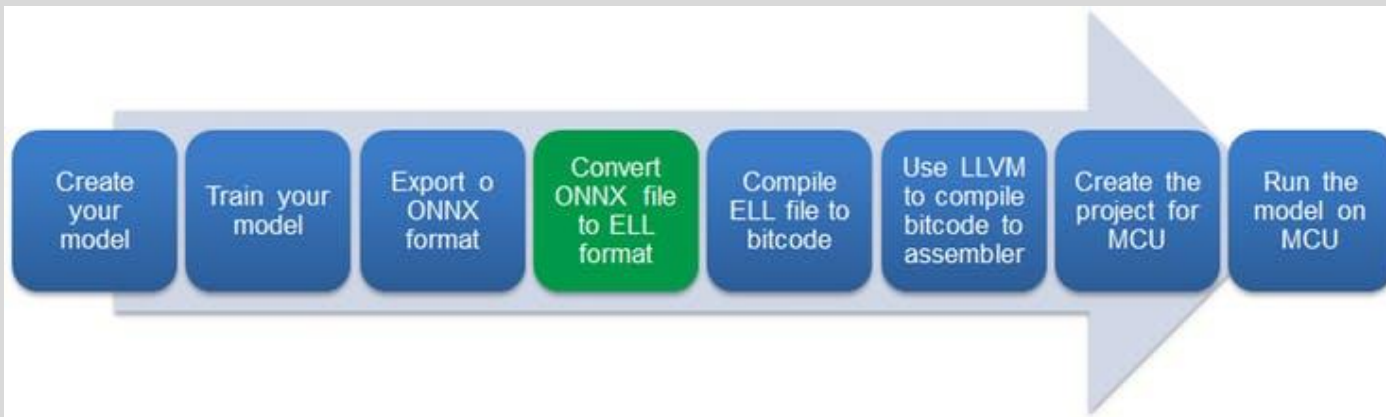


Demo

<https://github.com/rondagdag/onnx-pected/tree/master/webmnist-master>

Wait... there's more

- Embedded Learning Library
 - <https://github.com/microsoft/ELL>
- Machine Learning Model Running on Azure IoT Starter Kit
 - <https://www.hackster.io/waltercoan/machine-learning-model-running-on-azure-iot-starter-kit-f9608b>





Recap

- ✓ What is ONNX

ONNX is an open standard so you can use the right tools for the job and be confident your models will run efficiently on your target platforms

- ✓ How to create ONNX models

ONNX models can be created from many frameworks

- ✓ How to deploy ONNX models

ONNX models can be deployed with Windows ML, .NET/Javascript/Python and to the cloud with Azure ML and the high performance ONNX Runtime

Try it for yourself!

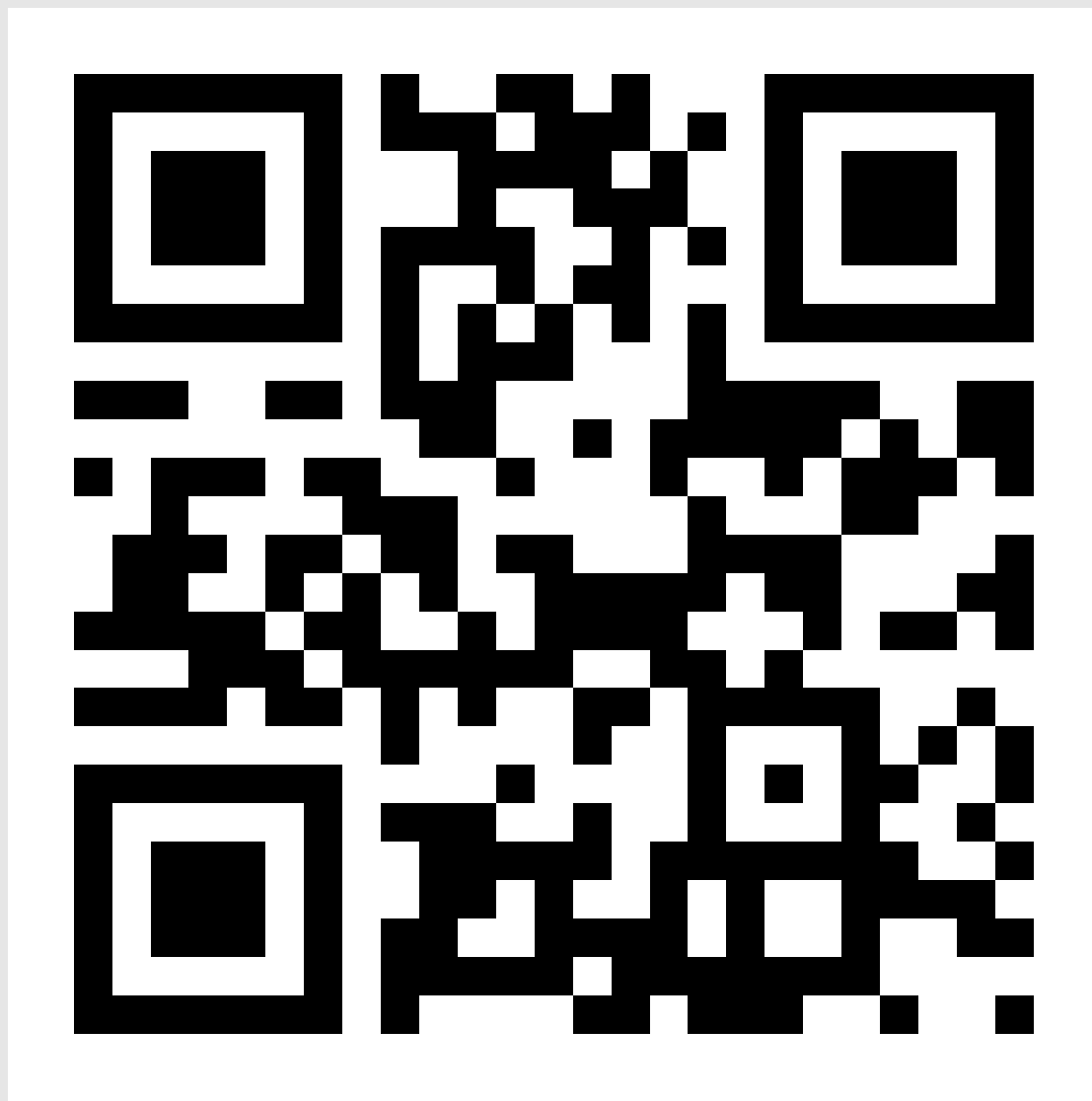
ONNX Runtime is available now!

```
pip install onnxruntime  
pip install onnxruntime-gpu
```

Documentation and samples at aka.ms/onnxruntime

Source for Demo:

<https://github.com/rondagdag/onnx-pected>



<http://bit.ly/ml-onnx>

About Me

Ron Dagdag



Ron Lyle Dagdag

Immersive Experience Developer

Cell: 682-560-3988

ron@dagdag.net



Experience AR

www.dagdag.net

@rondagdag

<http://ron.dagdag.net>

Sr. Software Developer/AI Edge Engineer

Microsoft MVP award – Windows Development

Personal Projects
www.dagdag.net

Email: ron@dagdag.net
Twitter @rondagdag

Connect me via Linked In
www.linkedin.com/in/rondagdag/

Feedback appreciated, help improve my presentation skills