Using ONNX for accelerated inferencing on cloud and edge

Prasanth Pulavarthi (Microsoft)

Kevin Chen (NVIDIA)

Agenda

- What is ONNX
- How to create ONNX models
- How to operationalize ONNX models (and accelerate with TensorRT)

Open and Interoperable AI

Open Neural Network Exchange

Open format for ML models

github.com/onnx

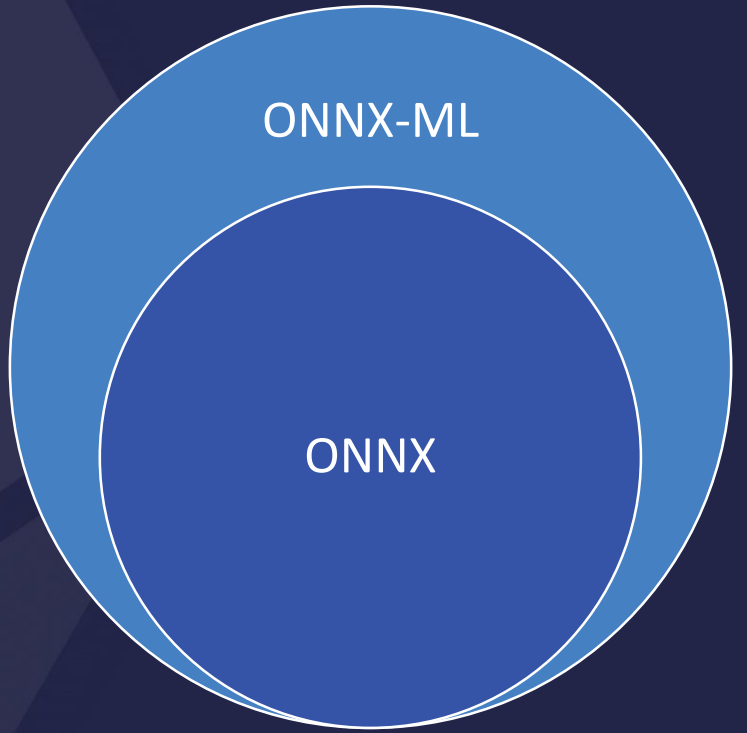
ONNX Partners

Key Design Principles

- Support DNN but also allow for traditional ML
- Flexible enough to keep up with rapid advances
- Compact and cross-platform representation for serialization
- Standardized list of well defined operators informed by real world usage

ONNX Spec

- File format
- Operators

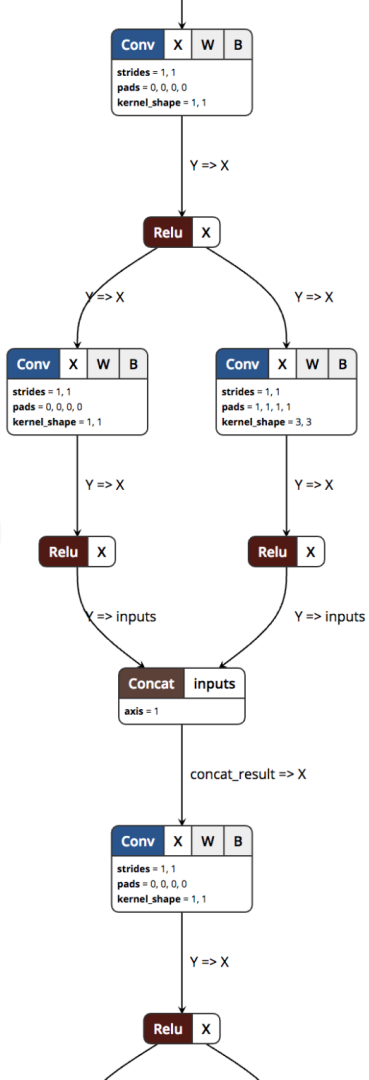


File format

Model
- Version info
- Metadata
- Acyclic computation dataflow graph

Graph
- Inputs and outputs
- List of computation nodes
- Graph name

Computation Node
- Zero or more inputs of defined types
- One or more outputs of defined types
- Operator
- Operator parameters



Data types

- Tensor type
  - Element types supported:
    - int8, int16, int32, int64
    - uint8, uint16, uint32, uint64
    - float16, float, double
    - bool
    - string
    - complex64, complex128
- Non-tensor types in ONNX-ML:
  - Sequence
  - Map

```
message TypeProto {
  message Tensor {
    optional TensorProto.DataType elem_type = 1;
    optional TensorShapeProto shape = 2;
  }
  // repeated T
  message Sequence {
    optional TypeProto elem_type = 1;
  };
  // map<K,V>
  message Map {
    optional TensorProto.DataType key_type = 1;
    optional TypeProto value_type = 2;
  };

  oneof value {
    Tensor tensor_type = 1;
    Sequence sequence_type = 4;
    Map map_type = 5;
  }
}
```

Operators

An operator is identified by <name, domain, version>

Core ops (ONNX and ONNX-ML)

- Should be supported by ONNX-compatible products
- Generally cannot be meaningfully further decomposed
- Currently 124 ops in ai.onnx domain and 18 in ai.onnx.ml
- Supports many scenarios/problem areas including image classification, recommendation, natural language processing, etc.

Custom ops

- Ops specific to framework or runtime
- Indicated by a custom domain name
- Primarily meant to be a safety-valve

Relu

Relu takes one input data (Tensor) and produces one output data (Tensor) where the rectified linear function, y = max(0, x), is applied to the tensor elementwise.

Version

This version of the operator has been available since version 6 of the default ONNX operator set. Other versions of this operator: Relu-1

Inputs

X : *T*
Input tensor

Outputs

Y : *T*
Output tensor

Type Constraints

T : *tensor(float16), tensor(float), tensor(double)*
Constrain input and output types to float tensors.

Examples

▼ relu

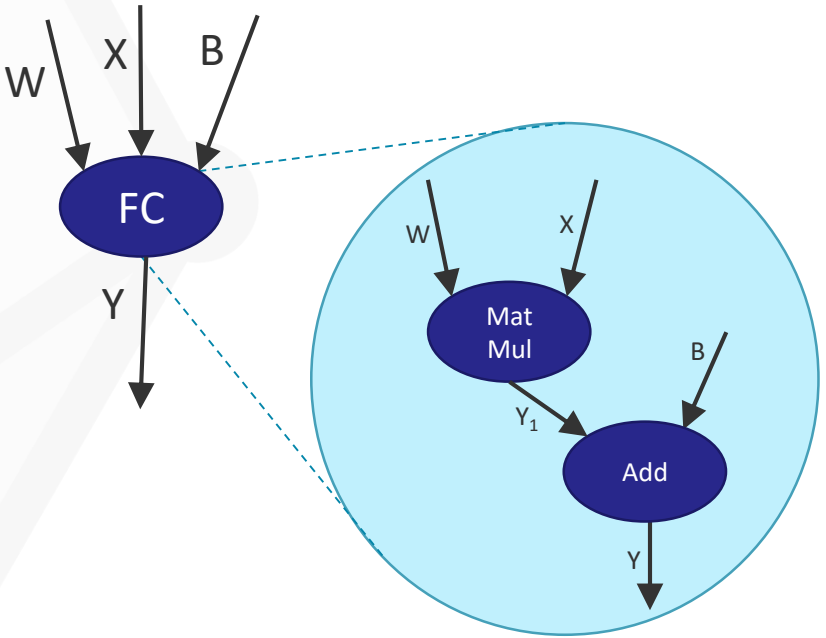
```
node = onnx.helper.make_node(
    'Relu',
    inputs=['x'],
    outputs=['y'],
)
x = np.random.randn(3, 4, 5).astype(np.float32)
y = np.clip(x, 0, np.inf)

expect(node, inputs=[x], outputs=[y],
       name='test_relu')
```

Functions

- Compound ops built with existing primitive ops

- Runtimes/frameworks/tools can either have an optimized implementation or fallback to using the primitive ops



ONNX is a Community Project

Get Involved

Discuss

Participate in discussions for advancing the ONNX spec.

gitter.im/onnx

Contribute

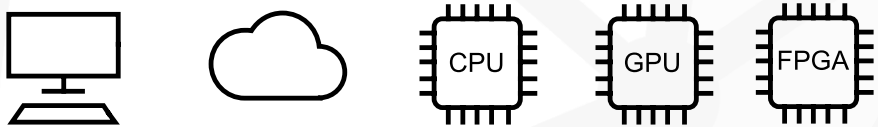
Make an impact by contributing feedback, ideas, and code.

github.com/onnx

ML @ Microsoft

- LOTS of internal teams and external customers
- LOTS of models from LOTS of different frameworks

- Different teams/customers deploy to different targets



Open and Interoperable AI

ONNX @ Microsoft

- ONNX in the platform
  - Windows
  - ML.net
  - Azure ML
- ONNX model powered scenarios
  - Bing
  - Ads
  - Office
  - Cognitive Services
  - more

ONNX @ Microsoft

Bing QnA - List QnA and Segment QnA

- Two models used for generating answers
- Up to 2.8x perf improvement with ONNX Runtime

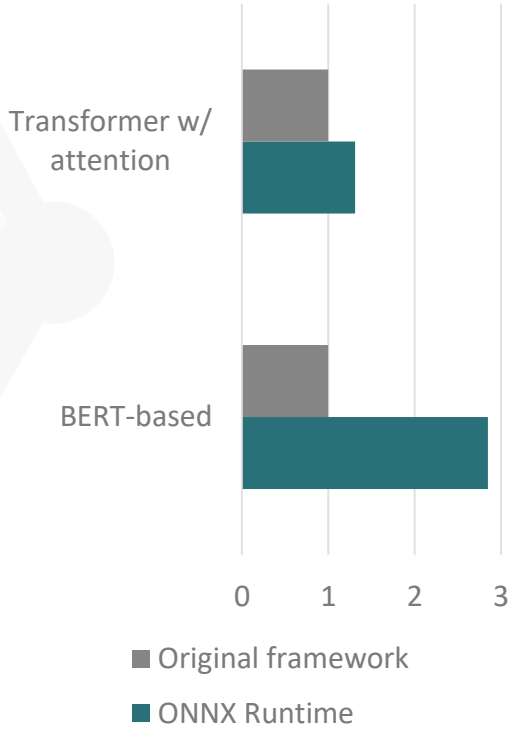
Query: empire earth similar games

Games Like Empire Earth

- Total War: Arena.
- Stronghold Kingdoms.
- Rise of Nations.
- Age of Empires 3.
- Rise of Nations: Rise of Legends.
- ... *(more items)*

19 Games Like Empire Earth - Games Finder
gameslikefinder.com/games-like-empire-earth/

Is this answer helpful?

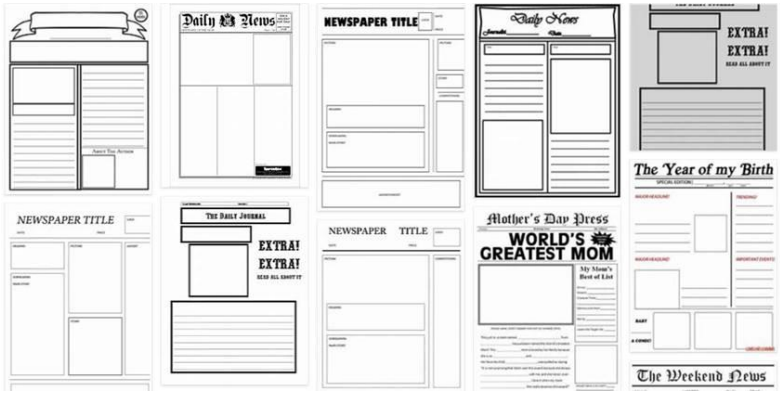


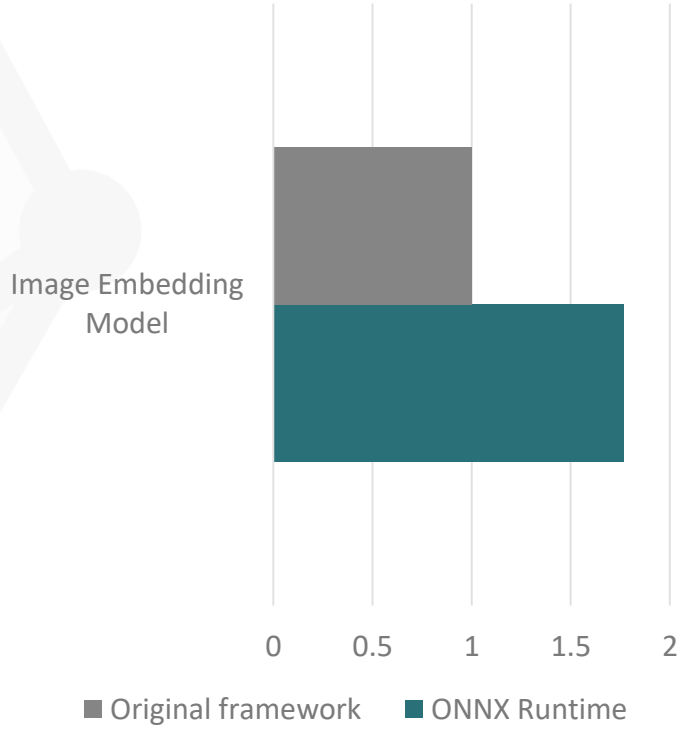
ONNX @ Microsoft

Bing Multimedia - Semantic Precise Image Search

- Image Embedding Model - Project image contents into feature vectors for image semantic understanding
- 1.8x perf gain by using ONNX and ONNX Runtime

Query: newspaper printouts to fill in for kids





ONNX @ Microsoft

- Teams are organically adopting ONNX and ONNX Runtime for their models – cloud & edge

- Latest 50 models converted to ONNX showed average 2x perf gains on CPU with ONNX Runtime

Agenda

- ✓ What is ONNX
- ❑ How to create ONNX models
- ❑ How to operationalize ONNX models

4 ways to get an ONNX model

ONNX Model Zoo

Services like Azure Custom Vision

Convert existing models

Train models in systems like Azure Machine Learning service

ONNX Model Zoo: github.com/onnx/models

Image Classification

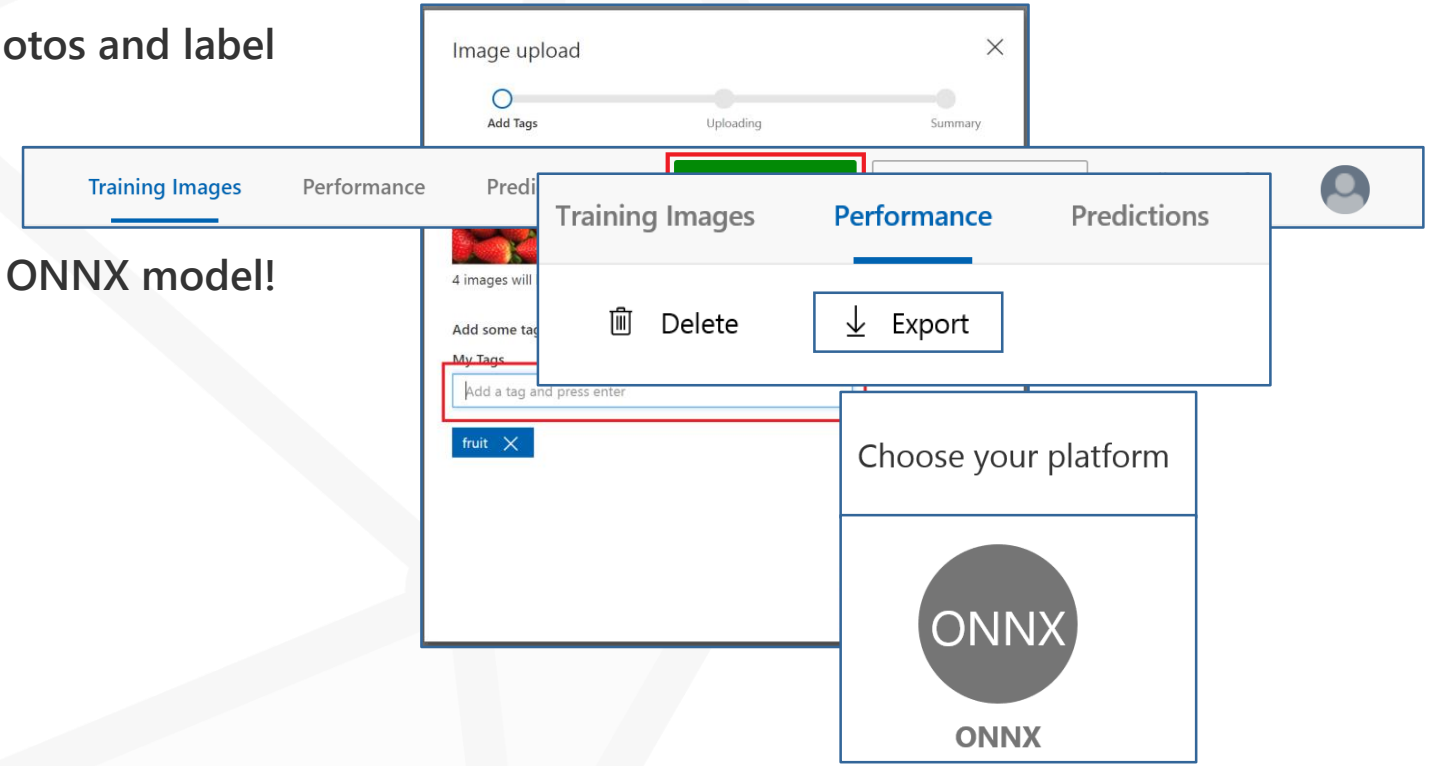
This collection of models take images as input, then classifies the major objects in the images into a set of predefined classes.

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 Challe... Top-5
SqueezeNet	Iandola et al.	A ligh... fewer... Top-5
VGG	Simonyan et al.	Deep... Challe... Top-5

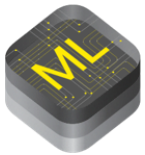
Model	Download	Checksum	Download (with sample test data)	ONNX version	Opset version	Top-1 accuracy (%)	Top-5 accuracy (%)
ResNet-18	44.6 MB	MD5	42.9 MB	1.2.1	7	69.70	89.49
ResNet-34	83.2 MB	MD5	78.6 MB	1.2.1	7	73.36	91.43
ResNet-50	97.7 MB	MD5	92.0 MB	1.2.1	7	75.81	92.82
ResNet-101	170.4 MB	MD5	159.4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

```
from keras.models import load_model
import keras2onnx
import onnx

keras_model = load_model("model.h5")

onnx_model = keras2onnx.convert_keras(keras_model, keras_model.name)

onnx.save_model(onnx_model, 'model.onnx')
```

Convert models: Chainer

```
import numpy as np
import chainer
from chainer import serializers
import onnx_chainer

serializers.load_npz("my.model", model)

sample_input = np.zeros((1, 3, 224, 224), dtype=np.float32)
chainer.config.train = False

onnx_chainer.export(model, sample_input, filename="my.onnx")
```

Convert models: PyTorch

```
import torch
import torch.onnx

model = torch.load("model.pt")

sample_input = torch.randn(1, 3, 224, 224)

torch.onnx.export(model, sample_input, "model.onnx")
```

PyTorch

Convert models: TensorFlow

Convert TensorFlow models from

- Graphdef file
- Checkpoint
- Saved model

```
python -m tf2onnx.convert
    --input SOURCE_GRAPHDEF_PB
    --graphdef SOURCE_GRAPHDEF_PB
    --checkpoint SOURCE_CHECKPOINT
    --saved-model SOURCE_SAVED_MODEL
    [--inputs GRAPH_INPUTS]
    [--outputs GRAPH_OUTPUS]
    [--inputs-as-nchw inputs_provided_as_nchw]
    [--target TARGET]
    [--output TARGET_ONNX_GRAPH]
    [--target TARGET]
    [--continue_on_error]
    [--verbose]
    [--custom-ops list-of-custom-ops]
    [--opset OPSET]
    [--fold_const]
```

ONNX-Ecosystem Container Image

- Quickly get started with ONNX
- Supports converting from most common frameworks
- Jupyter notebooks with example code
- Includes ONNX Runtime for inference

```
docker pull onnx/onnx-ecosystem
docker run -p 8888:8888 onnx/onnx-ecosystem
```

- TensorFlow
- Keras
- PyTorch
- MXNet
- SciKit-Learn
- LightGBM
- CNTK
- Caffe (v1)
- CoreML
- XGBoost
- LibSVM

Demo

BERT model using onnx-ecosystem container image

Agenda

- ✓ What is ONNX
- ✓ How to create ONNX models
- ❑ How to operationalize ONNX models

Create

Frameworks

Services

Azure Custom Vision Service

Native support

Converters

Native support

Converters

ONNX Model

Deploy

Azure

- Azure Machine Learning services
- Ubuntu VM
- Windows Server 2019 VM

Windows Devices

Linux Devices

Other Devices
(iOS, etc)

Demo

Style transfer in a Windows app

ONNX RUNTIME

- High performance
- Cross platform
- Lightweight & modular
- Extensible

ONNX Runtime

- High performance runtime for ONNX models
- Supports full ONNX-ML spec (v1.2 and higher, currently up to 1.4)
- Works on Mac, Windows, Linux (ARM too)
- Extensible architecture to plug-in optimizers and hardware accelerators
- CPU and GPU support
- Python, C#, and C APIs

ONNX Runtime - Python API

```
import onnxruntime

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

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

ONNX Runtime – C# API

```
using Microsoft.ML.OnnxRuntime;

var session = new InferenceSession("model.onnx");

var results = session.Run(input);
```

ONNX Runtime – C API

```
#include <core/session/onnxruntime_c_api.h>

// Variables
OrtEnv* env;
OrtSession* session;
OrtAllocatorInfo* allocator_info;
OrtValue* input_tensor = NULL;
OrtValue* output_tensor = NULL;

// Scoring run
OrtCreateEnv(ORT_LOGGING_LEVEL_WARNING, "test", &env)
OrtCreateSession(env, "model.onnx", session_options, &session)
OrtCreateCpuAllocatorInfo(OrtArenaAllocator, OrtMemTypeDefault, &allocator_info)
OrtCreateTensorWithDataAsOrtValue(allocator_info, input_data, input_count * sizeof(float), input_dim_values,
num_dims, ONNX_TENSOR_ELEMENT_DATA_TYPE_FLOAT, &input_tensor)
OrtRun(session, NULL, input_names, (const OrtValue* const*)&input_tensor, num_inputs, output_names,
num_outputs, &output_tensor));
OrtGetTensorMutableData(output_tensor, (void **) &float_array);

//Release objects
…
```

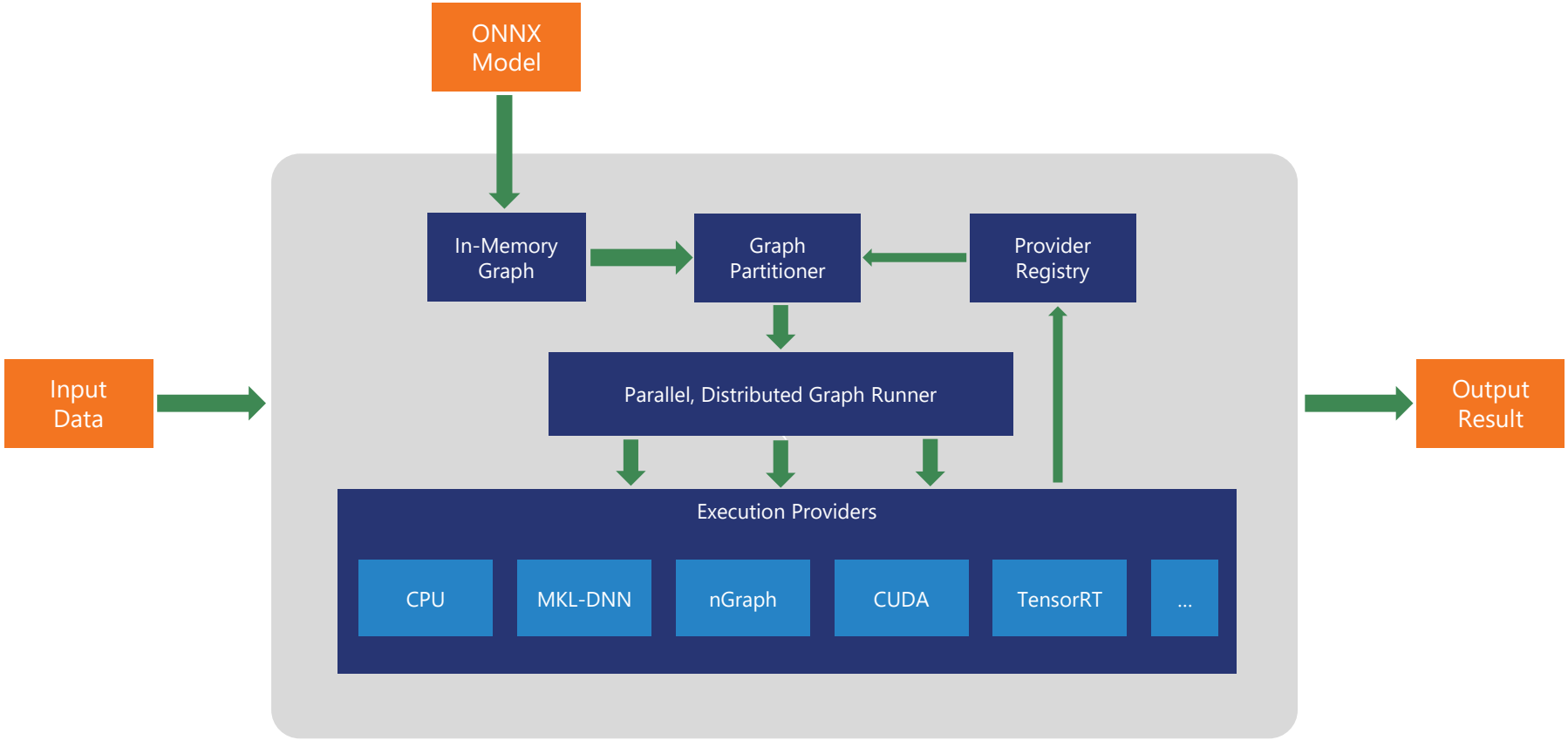
Demo

Action detection in videos

Evaluation videos from:
Sports Videos in the Wild (SVW): A Video Dataset for Sports Analysis
Safdarnejad, S. Morteza and Liu, Xiaoming and Udpa, Lalita and Andrus, Brooks and Wood, John and Craven, Dean

Demo

Convert and deploy object detection model as Azure ML web service


ONNX Model
In-Memory Graph
Graph Partitioner
Provider Registry
Input Data
Parallel, Distributed Graph Runner
Output Result
Execution Providers
CPU
MKL-DNN
nGraph
CUDA
TensorRT
...

Industry Support for ONNX Runtime

ONNX Runtime + TensorRT

- Now released as preview!
- Run any ONNX-ML model
- Same cross-platform API for CPU, GPU, etc.
- ONNX Runtime partitions the graph and uses TensorRT where support is available

NVIDIA TensorRT

Platform for High-Performance Deep Learning Inference

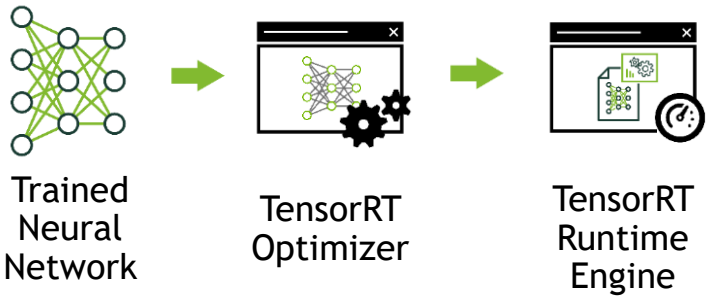
Optimize and deploy neural networks in production environments

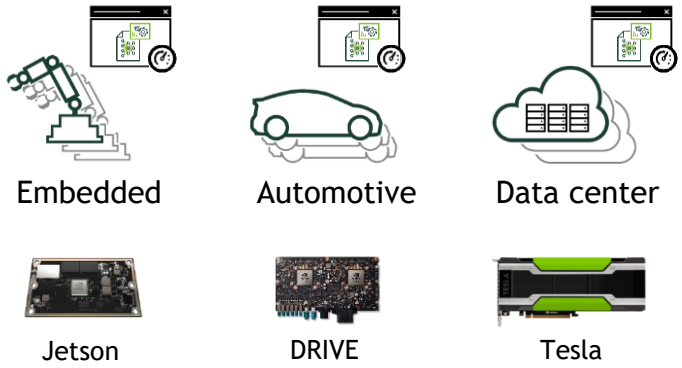
Maximize throughput for latency-critical apps with optimizer and runtime

Optimize your network with layer and tensor fusions, dynamic tensor memory and kernel auto tuning

Deploy responsive and memory efficient apps with INT8 & FP16 optimizations

Fully integrated as a backend in ONNX runtime





developer.nvidia.com/tensorrt

ONNX-TensorRT Parser

Available at https://github.com/onnx/onnx-tensorrt

ONNX-TensorRT Ecosystem

OPset<=9
ONNX >= 1.3.0

Public APIs

C++
Python

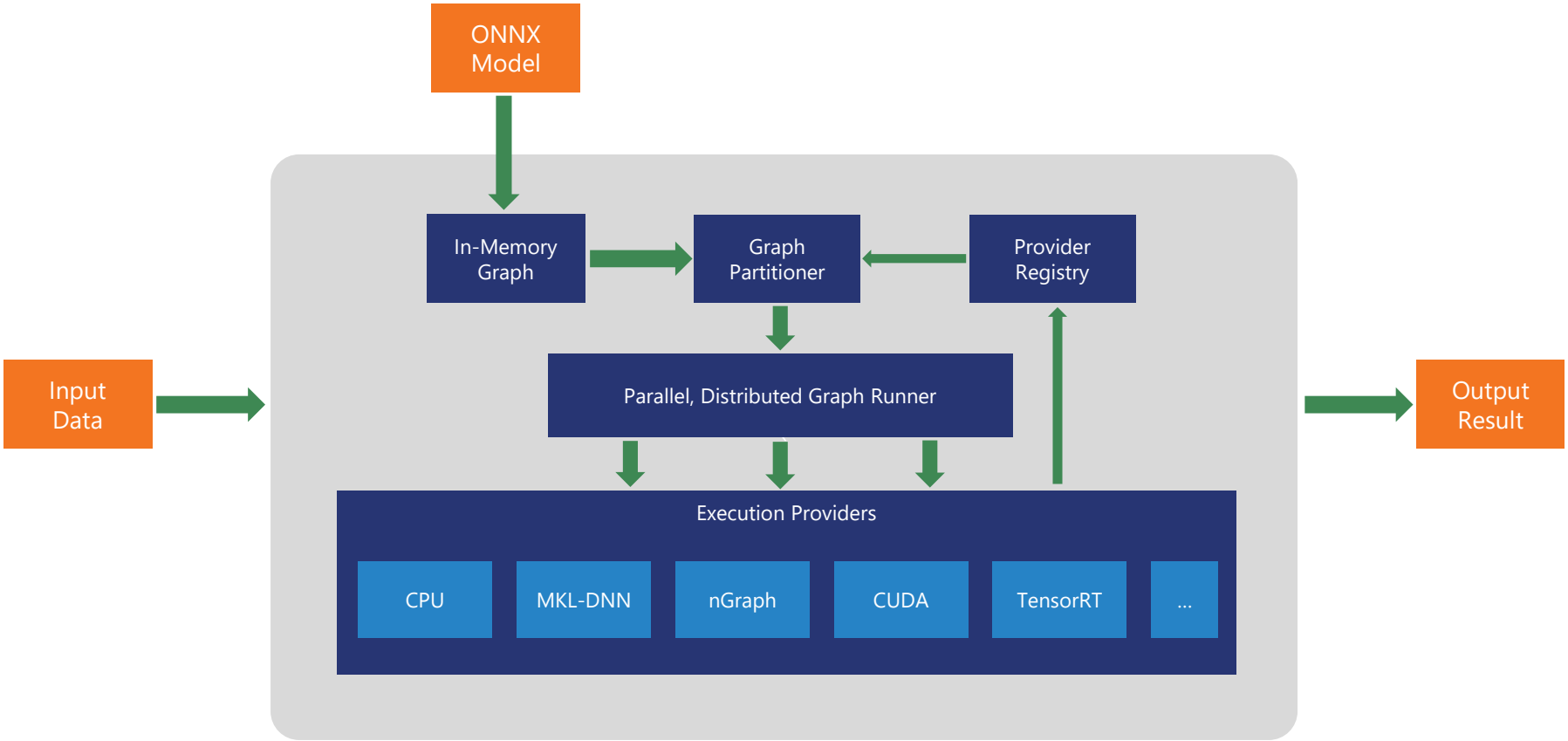
Supported Platforms

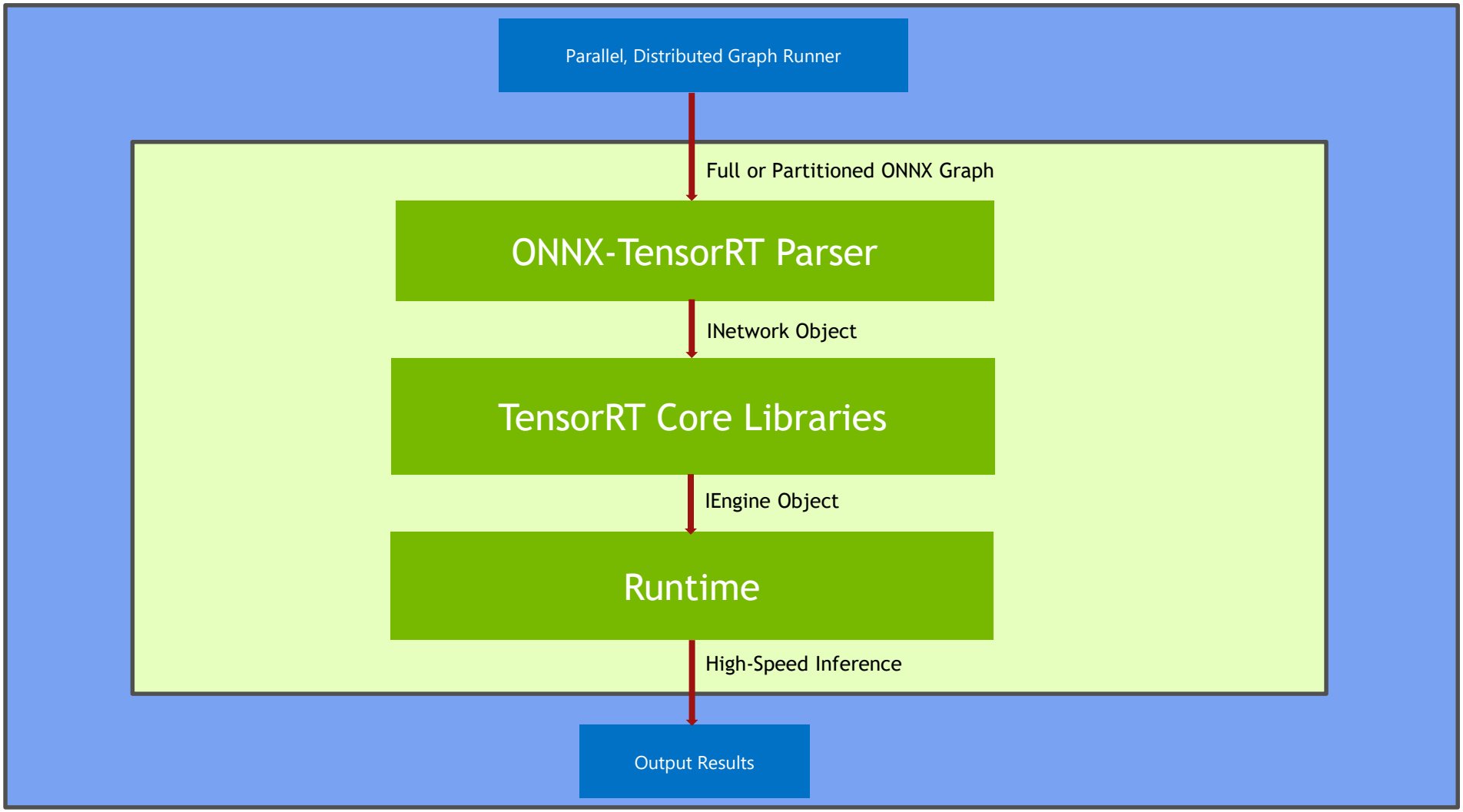
Desktop
+
Embedded
Linux

Upcoming Support

Windows
CentOS
IBM PowerPC

TensorRT Execution Provider in ONNX Runtime




Parallel, Distributed Graph Runner
Full or Partitioned ONNX Graph
ONNX-TensorRT Parser
INetwork Object
TensorRT Core Libraries
IEngine Object
Runtime
High-Speed Inference
Output Results

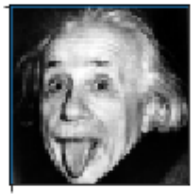
Demo

Comparing backend performance on emotion_ferplus ONNX zoo model

Demo performance comparison

ONNXRUNTIME-CPU

Model prediction: surprise
Inference time: 61.03 ms
Model Input image:

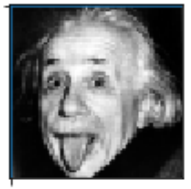


ONNXRUNTIME-GPU (using CUDA)

Model prediction: surprise
Inference time: 3.63 ms
Model Input image:

ONNXRUNTIME-TensorRT

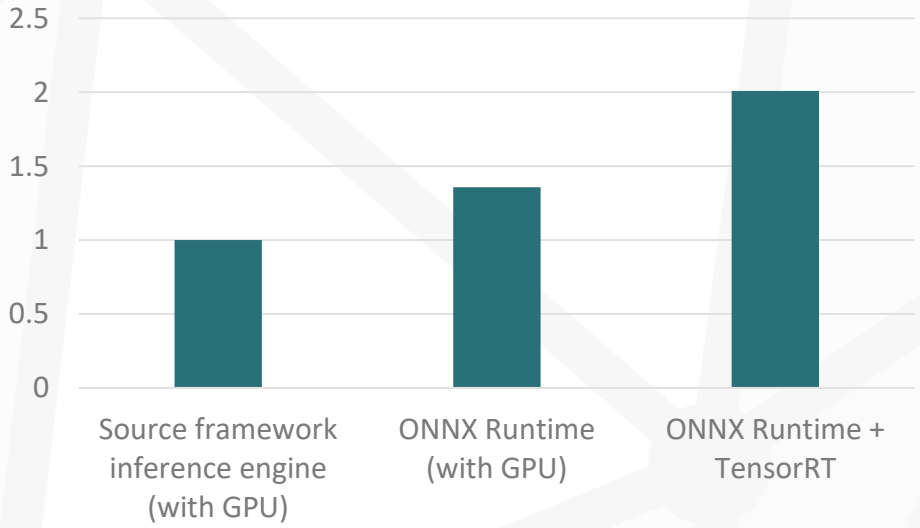
Model prediction: surprise
Inference time: 2.47 ms
Model Input image:



Model: Facial Expression Recognition (FER+) model from ONNX model zoo
Hardware: Azure VM – NC12 (K80 NVIDIA GPU)
CUDA 10.0, TensorRT 5.0.2

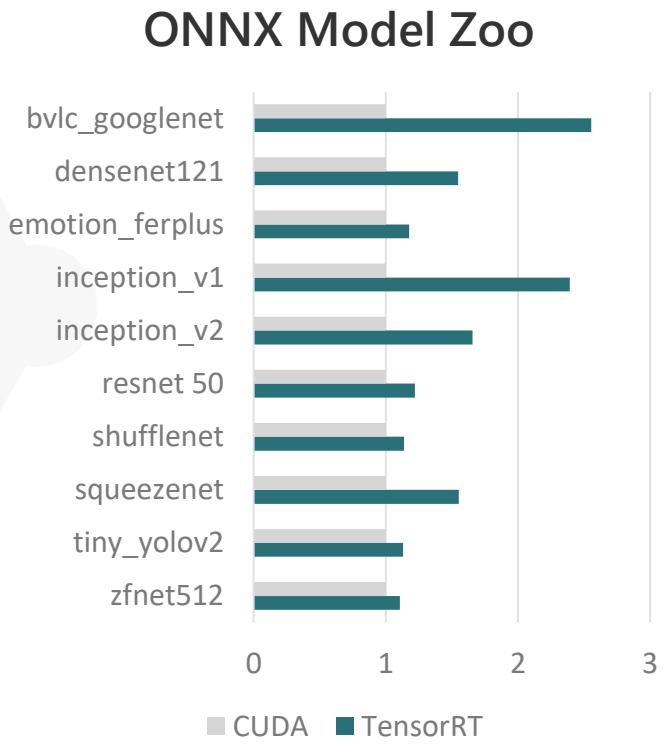
ONNX Runtime + TensorRT @ Microsoft

Bing Multimedia team seeing 2X perf gains



ONNX Runtime + TensorRT

- Best of both worlds
- Run any ONNX-ML model
- Easy to use API across platforms and accelerators
- Leverage TensorRT acceleration where beneficial



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 – use onnx-ecosystem container image to get started quickly

- ✓ **How to operationalize ONNX models**

  ONNX models can be deployed to the edge and the cloud with the high performance, cross platform ONNX Runtime and accelerated using TensorRT

ONNX RUNTIME

Try it for yourself

Available now with TensorRT integration preview!

Instructions at aka.ms/onnxruntime-tensorrt

Open sourced at github.com/microsoft/onnxruntime