Operationalizing AI - Portable ML Model Sharing across Enterprise

Adnan Masood, PhD.

October 2018

Slides Courtesy of Microsoft Corp. All product names, logos, and brands are property of their respective owners.
Adnan Masood, PhD.

Dr. Adnan Masood is an Artificial Intelligence and Machine Learning researcher, visiting scholar at Stanford AI Lab, software architect, and Microsoft MVP (Most Valuable Professional) for Artificial Intelligence. As Chief Architect of AI and Machine Learning, at UST Global, he collaborates with Stanford Artificial Intelligence Lab, and MIT AI Lab for building enterprise solutions.

Author of Amazon bestseller in programming languages, "Functional Programming with F#", Dr. Masood teaches Data Science at Park University, and has taught Windows Communication Foundation (WCF) courses at the University of California, San Diego. He is a regular speaker to various academic and technology conferences (WICT, DevIntersection, IEEE-HST, IASA, and DevConnections), local code camps, and user groups. He also volunteers as STEM (Science Technology, Engineering and Math) robotics coach for elementary and middle school students.
# Table of Contents

About this Playbook ................................................. 2
Partner Practice Development Framework .................. 5
What is Artificial Intelligence? ................................. 6
AI Opportunity ....................................................... 8
Industry Opportunities .......................................... 9
Define Your Strategy .............................................. 17
  Executive Summary ............................................. 18
  Define Your Practice Focus ................................. 19
  Understanding the AI Practice ............................. 20
  The Microsoft Approach to AI ............................. 23
  Pre-Built AI using Cognitive Services .................. 28
  Building Custom AI ........................................... 35
  Microsoft AI Platform Summary ......................... 42
  Define and Design the Solution Offer .................. 43
  Understanding Project Based Services ................ 44
  Understanding Managed Services ....................... 54
  Accelerate your Managed Service Model ................ 60
  Understanding Intellectual Property .................... 61
  Define Industry Specific Offerings ..................... 65
  Define Your Pricing Strategy ............................. 66
  Calculate Your Azure Practice Costs ................... 69
  Identify Partnership Opportunities ..................... 71
  Define Engagement Process ............................... 73
  Identify Potential Customers ............................. 74
  Join the Microsoft Partner Network .................... 75
  Stay Informed on AI Matters ............................. 77
  Identify Solution Marketplaces ......................... 78

Hire & Train .......................................................... 80
  Executive Summary .......................................... 81
  Hire, build, and train your team ....................... 82
  Job Descriptions for your Technical Team .......... 88
  Recruiting Resources ....................................... 98
  Training & Readiness ..................................... 99

Operationalize .................................................... 113
  Executive Summary ......................................... 114
  Implement a Process ....................................... 115
  Claim Your Internal Use Benefits ..................... 119
  Define Customer Support Program and Process .... 124
  Manage and Support an AI solution in Azure ....... 128
  Support Ticket Setup and Tracking ................... 130
  Implement Intellectual Property Offerings ......... 131
  Setup Social Offerings .................................. 132
  Create Engagement Checklists & Templates ........ 133

Go to Market & Close Deals .................................. 134
  Executive Summary ......................................... 135
  Marketing to the AI Buyer ............................... 136
  Engage Technical Pre-Sales in Sales Conversations .. 138
  Architecture Design Session (ADS) ................... 140
  Go-to-Market and Close Deals Guide ................. 142

Optimize & Grow .................................................. 143
  Executive Summary ......................................... 144
  Understanding Customer Lifetime Value ............ 145
  Guide: Optimize and Grow ............................. 147

AI Playbook Summary ........................................... 148

Microsoft Practice Development Playbook

aka.ms/practiceplaybooks
Microsoft AI Platform

AI Tools
- AI Toolkit (IoT)
- Azure Notebooks
- Azure ML Workbench
- VS Code Tools for AI
- ML Studio

AI Services
- Azure Bot Service
- Cognitive Services
- Machine Learning Services

AI on Data
- Cosmos DB
- SQL Database
- Data Lake Store
- AKS
- DSV
- Batch AI
- Spark

AI Compute

AI Infrastructure

Deep Learning Frameworks
- TensorFlow
- Caffe 2
- Cognitive Toolkit

See https://azure.microsoft.com/en-us/overview/ai-platform for more information about the various services and features of the Microsoft AI Platform.
Sample Real World ML Pipeline Architecture

Data Sources

Ingest / Prepare

Model

Train with Cloud AI

Deploy

Consume

Data:

- Preprocessed Text
- Deep Learning Virtual Machine (DLVM)

Model:

- Model: [Deep Neural Networks]
- Code: Python and TensorFlow

Train with Cloud AI:

- Visual Studio Tools for AI
- Azure Machine Learning
- Docker Image + IoT Hub
- Model Update + Manageability

Deploy:

- CoreML
- TensorFlow
- ONNX

Consume:

- Positive: 0.8923
- Negative: 0.1076
- Positive: 0.8923
- Negative: 0.1076
- Positive: 0.8923
- Negative: 0.1076
- Positive: 0.8923
- Negative: 0.1076

Action:

- IOT Edge device - minnowBoard

Intelligence:

- DATA
- INTELLIGENCE
- ACTION
Common AI/ML Problems:

• Most libraries provide state-of-the-art algorithms but little pertinent training data
• For many conversational domains, training data may be difficult or impossible to collect
• Pre-built domains streamline development but are largely irrelevant for most apps
• Tools for building custom domains can only handle narrow models and trivial apps
• ML capabilities only scratch the surface of what is typically required for production apps

Key Value Proposition:

• Not just offer an NLP library but provide expertise to work with bot framework for multiple modalities, commerce engine integration, and deployment infrastructure and expertise.

Machine Learning Development Lifecycle provides customized end to end solution from formal problem definition, domain modeling, creating training and test data, training models, evaluation of model, execution, deployment, and visualization.
Standard?

HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC)

SITUATION:
There are 14 competing standards.

14?! RIDICULOUS!
We need to develop one universal standard that covers everyone's use cases. YEAH!

SOON:
SITUATION:
There are 15 competing standards.
ONNX Motivation

Allow interoperability between frameworks
   Starting with CNTK, Caffe2 and PyTorch
Allow hardware vendor to focus on one IR in their backend optimization
Allow train in one toolkit and deploy in another
Deep Learning Frameworks Zoo

Framework backends

Vendor and numeric libraries

Apple CoreML, Nvidia TensorRT, Intel/Nervana ngraph, Qualcomm SNPE, ...
Open Neural Network Exchange

- Caffe2
- PyTorch
- TensorFlow
- mxnet
- Microsoft CNTK

Shared model and operator representation

From $O(n^2)$ to $O(n)$ pairs

Framework backends

- Apple CoreML
- Nvidia TensorRT
- Intel/Nervana ngraph
- Qualcomm SNPE

Vendor and numeric libraries
ONNX Vision

Language binding (C++, C#, Python, R, Javascript...etc)

- CNTK API
- Caffe 2 API
- PyTorch API
- Other ML toolkits

Open Neural Network Exchange (ONNX)

- MS Backend
- FB Backend
- Other Backend

CPU

GPU

FPGA

Custom silicon

Generated code
Open Neural Network Exchange (ONNX) is the first step toward an open ecosystem that empowers AI developers to choose the right tools as their project evolves. ONNX provides an open source format for AI models. It defines an extensible computation graph model, as well as definitions of built-in operators and standard data types. Initially we focus on the capabilities needed for inferencing (evaluation).

Caffe2, PyTorch, Microsoft Cognitive Toolkit, Apache MXNet and other tools are developing ONNX support. Enabling interoperability between different frameworks and streamlining the path from research to production will increase the speed of innovation in the AI community. We are an early stage and we invite the community to submit feedback and help us further evolve ONNX.
PyTorch

PyTorch is the framework for AI research at Facebook which enables rapid experimentation.

- Flexibility
- Debugging
- Dynamic neural networks

*Not* optimized for production and mobile deployments (Python)

When research projects produce valuable results, *the models need to be transferred to production.*

Traditionally, rewriting the training pipeline in a product environment with other frameworks.
ONNX Runtime for inferencing machine learning models now in preview

Posted on October 16, 2018

Faith Xu, Senior Program Manager, Machine Learning Platform

We are excited to release the preview of ONNX Runtime, a high-performance inference engine for machine learning models in the Open Neural Network Exchange (ONNX) format. ONNX Runtime is compatible with ONNX version 1.2 and comes in Python packages that support both CPU and GPU to enable inferencing using Azure Machine Learning service and on any Linux machine running Ubuntu 16.

ONNX is an open source model format for deep learning and traditional machine learning. Since we launched ONNX in December 2017 it has gained support from more than 20 leading companies in the industry. ONNX gives data scientists and developers the freedom to choose the right framework for their task, as well as the confidence to run their models efficiently on a variety of platforms with the hardware of their choice.
## Importing and Exporting from frameworks

<table>
<thead>
<tr>
<th>Framework / tool</th>
<th>Installation</th>
<th>Exporting to ONNX (frontend)</th>
<th>Importing ONNX models (backend)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caffe2</td>
<td>onnx/onnx-caffe2</td>
<td>Exporting</td>
<td>Importing</td>
</tr>
<tr>
<td>PyTorch</td>
<td>part of pytorch package</td>
<td>Exporting, Extending support</td>
<td>coming soon</td>
</tr>
<tr>
<td>Cognitive Toolkit (CNTK)</td>
<td>built-in</td>
<td>Exporting</td>
<td>Importing</td>
</tr>
<tr>
<td>Apache MXNet</td>
<td>onnx/onnx-mxnet</td>
<td>coming soon</td>
<td>Importing [experimental]</td>
</tr>
<tr>
<td>Chainer</td>
<td>chainer/onnx-chainer</td>
<td>Exporting</td>
<td>coming soon</td>
</tr>
<tr>
<td>TensorFlow</td>
<td>onnx/onnx-tensorflow</td>
<td>coming soon</td>
<td>Importing [experimental]</td>
</tr>
<tr>
<td>Apple CoreML</td>
<td>onnx/onnx-coreml</td>
<td>coming soon</td>
<td>Importing</td>
</tr>
</tbody>
</table>
Interoperability

- Having at disposal several libraries how we can interoperate between then for reusing training for inference, or transfer learning?
- Fight against fragmentation

For a while Caffe models have been used for exchange, ONNX or NNEF are proposed as interoperable solutions
  - Open Neural Network Exchange Format or Neural Network Exchange Format
- Tools around ONNX
  - Direct or indirect support for specific libraries
  - Runtime support by Nvidia TensorRT
ONNX

• Which kind of format is ONNX?
  – Based on Google Protobuf serialization
  – Describes network layers eventually with trained parameters
  – Node, Graph, Attribute, Operator, Value, Shape
  – All operators here:
    https://github.com/onnx/onnx/blob/master/docs/Operators.md

• Example with TF
  – https://github.com/onnx/tutorials/blob/master/tutorials/OnnxT
    ensorflowImport.ipynb

• Repository of Pre-trained Networks
  – https://github.com/onnx/models
  – E.g. ResNet-50 is 92MB
Sample application for ONNX models exported from Custom Vision Service

by Kurt Kramer
Last updated: 5/8/2018
Edit on GitHub

This sample application demonstrates how to take a model exported from the Custom Vision Service in the ONNX format and add it to an application for real-time image classification.

Getting Started
Prerequisites
- Visual Studio 17
- Windows 10 Insider Preview
- An account at Custom Vision Service

Quickstart
- Clone the repository and open the project in Visual Studio
- Build and run the sample Application
Open community

- Framework agnostic
- GitHub from the beginning
- Close partnerships and OSS contributions
ONNX is a community project.

https://onnx.ai
https://github.com/onnx
CNTK Latest Features (v2.2, v2.3)

New tutorials/examples/manuals
NCCL2 support
MKL-DNN integration
ONNX support
C#/.NET API
R-binding for CNTK
Model simplification/compression support
New ops and perf-improvements
Tensorboard support
Open Neural Network Exchange (ONNX)

ONNX is an open format to represent deep learning models

Supported by:
- CNTK
- PyTorch
- Caffe 2
- MxNet

Enabled interop-ability between frameworks
For more information: https://onnx.ai/
ONNX Motivation

Allow interoperability between frameworks
Allow hardware vendor to focus on one IR in their backend optimization
Allow train in one toolkit and deploy in another
ONNX Vision

Language binding (C++, C#, Python, R, Javascript...etc)

- CNTK API
- Caffe 2 API
- PyTorch API
- Other ML toolkits

Open Neural Network Exchange (ONNX)

- MS Backend
- FB Backend
- Other Backend

Generated code

- CPU
- GPU
- FPGA
- Custom silicon
ONNX Status in CNTK

V1 release in Github, focus on the basics
Support only inference, no loop, no condition and no gradient
Supported by CNTK, Caffe2, PyTorch and MxNet
Upcoming work:
  - Refined RNN support
  - Loop and control
Converter for other toolkits are coming soon
Open Neural Network Exchange (ONNX)

An open source intermediate representation (IR) of computation graph (https://github.com/onnx/onnx)

With defined common OPs and their semantics
Released on Sep. 7, 2017

Collaboration between Microsoft and Facebook
A share library with a Caffe2 example as reference
Permissive MIT license and no patents
Caffe2

Facebook’s in-house *production* framework
For training and deploying large-scale machine learning models
Focuses on several key features required by products:

- Performance
- Cross-platform support
- Coverage for fundamental machine learning algorithms (convolutional neural networks (CNNs), recurrent networks (RNNs), and multi-layer perceptrons (MLPs)) and up to tens of billions of parameters
Thank You!

https://ONNX.AI
https://github.com/onnx/onnx