ONNX Model Zoo Mixed Precision

Presenter(s): Bhargavi Karumanchi, Rodolfo Esteves, Rajeev Nalawadi
Intel Corporation

Reviewer(s): Andrew Sica (IBM)

Models with Mixed Precision

Mixed Precision Trends:

- Trend continues towards availability of “Models with Mixed Precision”
- TCO/Performance driving the need for mixed precisions
- Limited loss of accuracy while deploying mixed precisions
- Various data precisions available in hardware
  - FP32
  - FP16 / FP32
  - BF16 / FP32
  - Int8 / BF16
  - Int8 / BF16 / FP32
  - …..
  - ..many other potential combinations in future (eg: FP8, …)

ONNX Model Zoo Proposal

- Model zoo repository captures the mixed precision specifics as part of metadata

Name	Type	Description
ir_version	int64	The ONNX version assumed by the model.
opset_import	OperatorSetId	A collection of operator set identifiers made available to the model. An implementation must support all operators in the set or reject the model.
producer_name	string	The name of the tool used to generate the model.
producer_version	string	The version of the generating tool.
domain	string	A reverse-DNS name to indicate the model namespace or domain, for example, 'org.onnx'
model_version	int64	The version of the model itself, encoded in an integer.
doc_string	string	Human-readable documentation for this model. Markdown is allowed.
graph	Graph	The parameterized graph that is evaluated to execute the model.
metadata_props	map<string,string>	Named metadata values; keys should be distinct.
training_info	TrainingInfoProto[]	An optional extension that contains information for training.
functions	FunctionProto[]	An optional list of functions local to the model.

Capture in metadata the combinations of data precisions supported in model

Benefits of Mixed Precision (metadata)

Mixed Precision conveyed using metadata:

- Enables model authors to convey all the data precisions utilized in the model
- Allows model consumers to make relevant decisions based on hardware features supported

Thank You !!

ONNX Model Provenance

Presenter: Bhargavi Karumanchi (Intel)
Contributors: Rodolfo Esteves, Rajeev Nalawadi
Intel Corporation

Why ONNX Model Provenance

- Industry trends towards digitization have accelerated into broader AI infusion across various vertical segments
- While AI models get integrated into End-2-End process flows
- Its becoming increasingly hard to detect whether specific models being integrated can meet the characteristics
  - Fairness, Transparency, Human centered approach, Trusted, Secure, Privacy protections, machine readable
- Once the ONNX model is created using converters, provide the ONNX model creators/converters an additional option for establishing provenance prior to broader publishing
- ONNX model provenance as a step towards Responsible AI

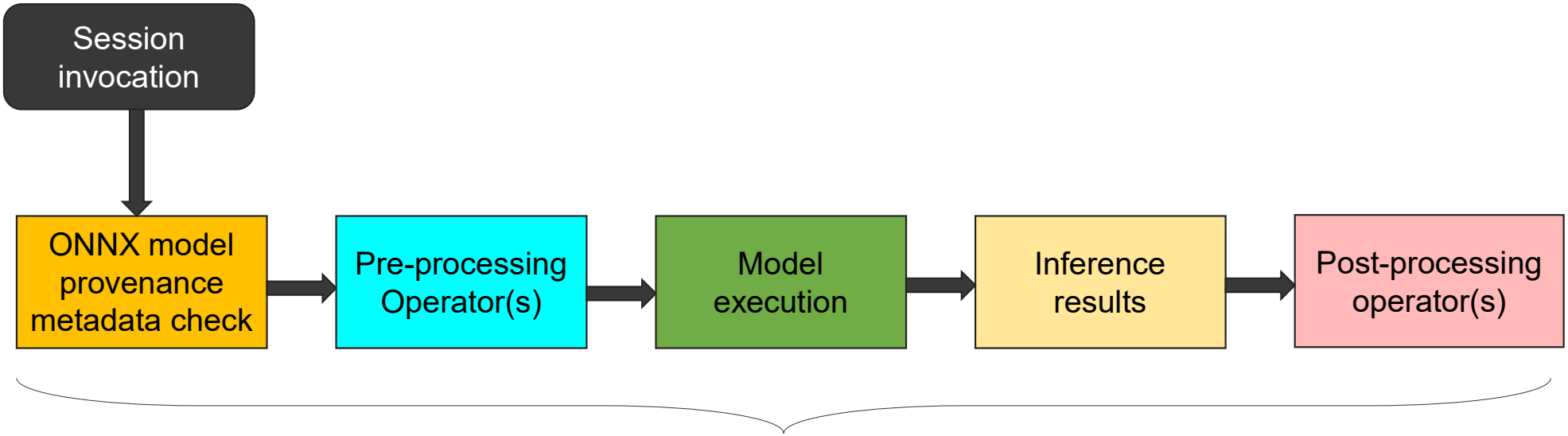
Goals & Requirements

- Native framework / converter tools to generate metadata properties for model
  - Model Information: <Description of model>
  - Model Architecture: <NLP/CNN/…, Dataset(s) used, Format of Inputs & Outputs, Accuracy expected>
  - Usage scenarios: <Typical applications of model>
  - Privacy considerations: <consent required etc..>
  - ……other metadata properties….
- Requires ONNX model provenance (metadata) check to be implemented in runtime(s)
  - Model metadata characteristics performed queried at start of session

Recommended guidelines

- Minimal additions to ONNX(memory) payload
- Minimal requirements for extra processing
- As backwards-compatible as possible (rejecting non-signed/non-annotated models should be an opt-in)

Flow with Model provenance metadata check (proposal)



ONNX Graph

Name	Type	Description
ir_version	int64	The ONNX version assumed by the model.
opset_import	OperatorSetId	A collection of operator set identifiers made available to the model. An implementation must support all operators in the set or reject the model.
producer_name	string	The name of the tool used to generate the model.
producer_version	string	The version of the generating tool.
domain	string	A reverse-DNS name to indicate the model namespace or domain, for example, 'org.onnx'
model_version	int64	The version of the model itself, encoded in an integer.
doc_string	string	Human-readable documentation for this model. Markdown is allowed.
graph	Graph	The parameterized graph that is evaluated to execute the model.
metadata_props	map<string,string>	Named metadata values; keys should be distinct.
training_info	TrainingInfoProto[]	An optional extension that contains information for training.
functions	FunctionProto[]	An optional list of functions local to the model.

Capture in metadata the model provenance aspects

Maybe we should mandate a format for this information that is also machine-readable. For example, the **Semantic Web infrastructure**!

The Semantic Web infrastructure (RDF)

- Structured metadata, extensible and machine readable
- Embeddable (compatible with existing ONNX metadata fields)
- Controlled vocabularies for Provenance, Explainable and Ethical ML are already being developed (eg IEEE P7003)

Future Thoughts for consideration

- Should we establish a model lineage tracker for QAT & finetuning scenarios
  - Custom datasets used to generate a model by fine-tuning & QAT flows
  - Accuracy trends as models progress through the lineage

- Tracking hash/checksum of models progression
  - Any considerations for 3rd party verifications (CCF, ledgers, etc..)
  - Potential monetization avenue for model creators and their customized datasets

Backup

The Semantic Web infrastructure (RDF)

```
@prefix schema: <http://schema.org/>
.
@prefix gndo: <https://d-nb.info/standards/elementset/gnd#> .
@prefix lib: <http://purl.org/library/> .
@prefix marcRole: <http://id.loc.gov/vocabulary/relators/> .
@prefix dcmitype: <http://purl.org/dc/dcmitype/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix geo: <http://www.opengis.net/ont/geosparql#> .

<https://d-nb.info/gnd/1045328480> a gndo:BuildingOrMemorial ;
        foaf:page <http://de.wikipedia.org/wiki/The_Shard> ;
        gndo:gndIdentifier "1045328480" ;
        gndo:geographicAreaCode <https://d-nb.info/standards/vocab/gnd/geographic-area-code#XA-GB> ;
        gndo:definition "72-stöckiges u. 310 m hohes, multifunktionales Hochhaus am Südufer"@de ;
        gndo:dateOfProduction "16.03.2009-01.02.2013" ;
        gndo:preferredNameForThePlaceOrGeographicName "The Shard (London)" .
```

Controlled vocabularies for Provenance, Explainable and Ethical ML are already being developed (eg IEEE P7003)

The Semantic Web infrastructure (RDF)

```
@prefix schema: <http://schema.org/>
.
@prefix gndo: <https://d-nb.info/standards/elementset/gnd#> .
@prefix lib: <http://purl.org/library/> .
@prefix marcRole: <http://id.loc.gov/vocabulary/relators/> .
@prefix dcmitype: <http://purl.org/dc/dcmitype/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix geo: <http://www.opengis.net/ont/geosparql#> .

<https://d-nb.info/gnd/1045328480> a gndo:BuildingOrMemorial ;
        foaf:page <http://de.wikipedia.org/wiki/The_Shard> ;
        gndo:gndIdentifier "1045328480" ;
        gndo:geographicAreaCode <https://d-nb.info/standards/vocab/gnd/geographic-area-code#XA-GB> ;
        gndo:definition "72-stöckiges u. 310 m hohes, multifunktionales Hochhaus am Südufer"@de ;
        gndo:dateOfProduction "16.03.2009-01.02.2013" ;
        gndo:preferredNameForThePlaceOrGeographicName "The Shard (London)" .
```

Controlled vocabularies for Provenance, Explainable and Ethical ML are already being developed (eg IEEE P7003)

RDF alternative serialization: LD-JSON

```
<script type="application/ld+json">
{
  "@context":"http://schema.org",
  "@type":"NewsArticle",“
  description":"Don’t fall for the Trump infrastructure scam.",
  mainEntityOfPage":"https://www.nytimes.com/2016/11/21/opinion/build-he-wont.html",
  "url":"https://www.nytimes.com/2016/11/21/opinion/build-he-wont.html",
  "author":[{
    "@context":"http://schema.org",“
    @type":"Person",“
    url":"https://www.nytimes.com/by/paul-krugman",
    "name":"Paul Krugman"}
  ],
  “dateModified":"2016-11-21T16:31:49.000Z",
  "datePublished":"2016-11-21T08:21:07.000Z“
}
</script>
```

Can be embedded in HTML pages and readily processed in Javascript

RDF alternative serialization: yaml embedded in Markdown

---

description: Use this topic to help manage Windows and Windows Server technologies with Windows PowerShell.

Download Help Link: https://aka.ms/winsvr-2022-pshelp

Help Version: 5.0.2.1

Locale: en-US

Module Guid: af4bddd0-8583-4ff2-84b2-a33f5c8de8a7

Module Name: Hyper-V

ms.date: 12/20/2016

title: Hyper-V

Hyper-V Module

Description

This reference provides cmdlet descriptions and syntax for all Hyper-V-specific cmdlets. It lists the cmdlets in alphabetical order based on the verb at the beginning of the cmdlet.

Can be embedded (unobstrusively) in the Markdown for Model Cards

The Semantic Web infrastructure (SPARQL)

```
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>
PREFIX p: <http://www.wikidata.org/prop/>
PREFIX ps: <http://www.wikidata.org/prop/statement/>
PREFIX pq: <http://www.wikidata.org/prop/qualifier/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT DISTINCT ?laureateName ?awardYear ?warName
WHERE {
  ?laureate p:P166 ?award .          # Winner of some prize
  ?award ps:P166 wd:Q37922 .         # Prize is Nobel Pr. in Lit.
  ?award pq:P585 ?awardDate .        # Get the date of the award
  BIND(YEAR(?awardDate) AS ?awardYear) # Get the year of the award
  ?laureate wdt:P607 ?war .          # Find war(s) laureate was in
  ?war rdfs:label ?warName .
  FILTER(LANG(?warName)="en"         # Only English labels
  && LANG(?laureateName)="en")       #  ...  names only
} ORDER BY ?awardYear ?warStart      # Oldest award (then war) first
```

ONNX APIs can be provided to extract and query (SPARQL) semantic content. Queries useful in provenance and fairness checks can be made available.

Thank You !!