Request for Quotations

Caffe2 to NNEF Converter

January 2018

Notice

ALL KHRONOS SPECIFICATIONS AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." KHRONOS MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Information furnished is believed to be accurate and reliable. However, Khronos assumes no responsibility for the consequences of use of such information or for any infringement of patents or other rights of third parties that may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Khronos. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied.

Trademarks

Khronos and NNEF, and associated logos are trademarks or registered trademarks of Khronos Group Inc. All other product names, trademarks, and/or company names are used solely for identification and belong to their respective owners.

1 Background

Converters from Training Frameworks to NNEF, and from NNEF to Training Frameworks, are an essential part of NNEF success as an exchange format for neural networks.

The NNEF working group focuses on converters to and from a handful of selected frameworks. Conversion to NNEF is more important initially, in order to allow HW vendors to write their own converters from NNEF to the vendor specific distribution format. However, converters are easier to design and implement with both directions in mind. Therefore we describe this project with both directions in mind, putting more emphasis on the framework to NNEF direction.

The goal of this project is to procure an implementation of a converter between Caffe2 and NNEF. This converter will be uploaded by Khronos to GitHub.

2 Requirements

2.1 General

The project will deliver Caffe2 to NNEF converter that receives a set of Caffe2 protobuf files and generates semantically and functionally equivalent NNEF container.

2.2 Implementation

The converter will be written in Python, with minimal dependency on 3rd party components (either as source or as binary).

The implementer is required to provide the list of 3rd party components believed will be needed to implement the converter as part of the SOW definition. This list should include only components which are open source, preferably under Apache 2.0 license, and should get Khronos approval as part of accepting the work. Any additional component found during the work on the converter should be communicated to Khronos and get approved as well as a condition for accepting the resulted converter.

caffe2_pb2 (see Ref [2]) is considered required and approved for this converter work, as it contains Caffe2 protobuf python module. Furthermore, the NNEF parser open-sourced by the NNEF group [5] is also considered approved and required.

2.3 Input of Converter to NNEF

The converter's input will be the adequate set of protobuf files, and a few optional parameters.

Required:

- A network definition protobuf file (also referred in some links as predict net)
- Weights protobuf file (also referred in some links as exec net or init net)

Optional:

- Output path for creating the textual file and subfolders tree. If not provided, outputs will be written to current directory.
- External Input list. If provided, overrides the external_input list provided in the predict net NetDef
- External Output list. If provided, overrides the external_output list provided in the predict net NetDef
- Generate a compressed zip container. If provided, compress the textual file and subfolders into a zip file.

2.4 Output of Converter to NNEF

The converter will generate a textual file and tree of subfolders conforming to the container organization defined in NNEF Chapter 5.

In addition, the Converter will output a conversion report log, describing the number of operations successfully converted, warnings on possible conversion issues, and errors encountered during the conversion.

2.5 Conversion Requirements

The converter is required to analyze the NetDef description which appears in the network definition protobuf file in the following manner:

1. Inputs
   - 1.1. If the converter received a list of external inputs as a parameter, it shall use them as the list of graph inputs for this conversion operation. Any other inputs in the provided NetDef shall be discarded, and corresponding connected graph operations shall not be converted.
   - 1.2. If external list is not provided, the converter will look for list of inputs defined by the NetDef external_input optional field. If such list exists, it shall use them as the list of graph inputs for this conversion operation. Any other inputs in the provided NetDef shall be discarded, and corresponding connected graph operations shall not be converted.
   - 1.3. If list of inputs is not provided (either by parameter or by the optional NetDef field), the converter is required to analyze the graph expressed by the NetDef, find all graph inputs, and write them into the graph in the converted NNEF textual file.
2. Outputs
   - 2.1. If the converter received a list of external outputs as a parameter, it shall use them as the list of graph outputs for this conversion operation. Any other outputs in the provided NetDef shall be discarded, and corresponding connected graph operations that generate this discarded outputs shall not be converted.
   - 2.2. If external list is not provided, the converter will look for list of outputs defined by the NetDef external_output optional field. If such list exists, it shall use them as the list of graph outputs for this conversion operation. Any other outputs in the provided NetDef shall be discarded, and corresponding connected graph operations that generate this discarded outputs shall not be converted.
   - 2.3. If list of outputs is not provided (either by parameter or by the optional NetDef field), the converter is required to analyze the graph expressed by the NetDef, find all graph outputs, and write them into the graph in the converted NNEF textual file.
3. Operations
   - 3.1. For each operation in the NetDef file, the converter is required to extract the operation type and identify if it is supported by NNEF. A minimal list of supported types for conversion is provided in section 2.6.
   - 3.2. If supported, the converter will create an NNEF equivalent description in the textual file of the operation type, inputs, outputs, arguments and any additional field required to create the description.
   - 3.3. The converter will correctly map tensor connections between operations outputs to other operations inputs, ensuring graph-level uniqueness of the tensor names.
   - 3.4. The converter will correctly handle connection of external tensors such as weights, biases, etc, ensuring correct name mapping to relevant tensor folder and file name.

2.6 Backward Conversion

The conversion from NNEF to Caffe2 will be implemented as a separate tool (Python script). This tool may receive any NNEF file. For NNEF files that are the result of a conversion from Caffe2 to NNEF, this tool must successfully convert the network back to Caffe2. For other NNEF files, the conversion may fail if it encounters features of NNEF that are unsupported by Caffe2 (certain operations or parameters of operations). The backward conversion process is otherwise similar to the conversion to NNEF described above, with the simplification that no handling of externally provided list of inputs and/or outputs need to be supported, only what is described by the NNEF file itself.

2.7 Required Operations

The table below provides the minimal list of operations required to be supported by the converter.

Caffe2 Operation	Suggested NNEF primitive	Notes/Restrictions
Add	add	
Allreduce	sum_reduce	
And	and	
Append	concat	
AveragePool	avg_pool	
Concat	concat	
Conv	conv	
ConvTranspose	deconv	
DepthConcat	concat	
DepthSplit	split	
DotProduct	mul	
FC	linear	
Flatten	flatten	
Elu	elu	
Exp	exp	
GE	ge	

GT	gt	
LE	le	
LeakyRelu	leaky_relu	
LRN	local_response_normalization	
LT	lt	
MatMul	matmul	
Max	max	
MaxPool	max_pool	
MaxPoolWithIndex	max_pool_with_index	
Mul	mul	
Negative	neg	
Not	not	
Normalize	l2_normalization	
Or	or	
Reduce	sum_reduce	
ReduceFrontMean	mean_reduce	
Reshape	reshape	
Sigmoid	sigmoid	
Softmax	softmax	
Softplus	softplus	
Split	split	
Sub	sub	
Sum	add_n	
Tanh	tanh	
Transpose	transpose	

Relu	relu	

Notes:

- Some operations may require two-pass conversion. NNEF members with experience of conversion will be available for more information.
- Some operations may be broken up into primitives by Caffe2, which are not supported by NNEF, but the whole higher level operation is supported (for example padded convolution with certain cases of padding). In this case, the higher level operation is the only way to export to NNEF.

3 Deliverables and Acceptance Criteria

3.1 Deliveries

The scope of the Caffe2 to NNEF converter Implementation project will include the following deliverables:

- All source code for the converter
- Implementation notes document summarizing implementation decisions made during the course of the project
- A set of simplified Caffe2 models which test the operations defined in section 2.6.
- A detailed log of running the converter on the Caffe2 model zoo models defined in 3.2, and the simplified Caffe2 models described on previous bullet.

3.2 Review Period

1. The working group will have a review period of 4 weeks to review the converter code. At the end, it will provide a list of issues to be fixed, ranging these issues as critical/high/medium/low.
2. All critical and high issues need to be fixed/addressed as part of the Acceptance Criteria.

3.3 Acceptance Criteria

1. The Caffe2 to NNEF converter is required to be able to successfully convert the following Caffe2 models which appear on the Model Zoo (See Ref [3]) , under the “Caffe2” column: AlexNet, GoogleNet, Squeezenet.
2. The Caffe2 to NNEF converter is required to convert a set of simplified Caffe2 models which cover all operations described in section 2.6.
3. All converted models from #1 and #2, provided as NNEF containers, should successfully pass the NNEF validator check.
4. All converted models are required to be possible to convert back to Caffe2 from NNEF, and the result of the backward conversion should result in networks functionally equivalent to the ones from which the conversion to NNEF was started.
5. All issues found during the review period and classified as critical/high should have been fixed.

4 Project Scoping and Schedule

The NNEF working group estimates that the project can achieve complete implementation, testing and documentation, in no more than 6 man weeks.

Below are the suggested project milestones. We will assess progress on a weekly basis, so the feature coverage timeline below is only a rough guideline to the order in which we expect to have validator and tests written. Please provide detailed milestone dates that you can commit to delivering:

Milestone	Date	Content	Notes
M1		Khronos releases RFQ	
M2	M1 + 4 weeks	RFQ responses received by Khronos	
M3	M2 + 2 weeks	Contractor selected and notified	
M4	M3 + 3 weeks	Contract executed and start of work	10% of money is provided
M5	M4 + 3 weeks	Simple network end to end functional with minimum operations implemented.	30% of money is provided
M6	M5 + 3 weeks	100% of converter implemented	30% of money is provided
M7	M6 + 4 weeks	Review Period over	
M8	M7 + 3 weeks	Issues found during review period fixed, project is complete.	30% of money is provided

5 Khronos NDA, Contractor and Membership Agreement

The selected contractor will be required to execute the standard Khronos Contractors Agreement with milestones and costs entered into Exhibit B and Contractor Disclosures entered into Exhibit C.

If the selected contractor is not a Khronos member, the contractor shall also be required to execute the standard Khronos membership agreement (with fees waived) for the duration of the project in order to gain access to confidential materials and meetings for the sole purpose completing deliverables in this RFQ.

No work shall begin, and Khronos shall be liable for no costs or expenses, until the selected contractor is in receipt of an executed contractor's agreement.

It is important that contractors understand that Khronos will be assessing progress on a regular basis, and reserve the right to terminate or renegotiate the contract in the event insufficient progress is being made.

6 RFQ Responses

The RFQ response materials will form the basis for detailed milestone and cost negotiations for the final contract with the selected vendor or vendors. Please provide the following information in the format of your choice:

- Identification of deliverables on which you wish to bid;
- Proposed schedule, highlighting any differences from the suggested milestones in Section 4;
- The hourly cost for engineering resources from your company, the minimum and maximum number of hours you can commit to this project on a weekly basis, and a description of the qualification of the engineering resource(s) which would be used;
- The total project cost to Khronos. We can accept time and material or fixed cost bids – but strongly prefer fixed cost proposals;
- An indication you are willing to work under the terms of the standard Khronos Contractor Agreement and execute the Khronos membership agreement if necessary;
- Any particular issues or risk factors that you wish to highlight;
- Supporting materials, including background materials about your company, highlighting other relevant experience and expertise for this project.

RFQ responses are requested by the close of business on TBD and should be sent to nnef-rfq@khronos.org.

7 References

[1] Caffe2 project in Github : https://github.com/caffe2/caffe2

[2] Caffe2 Protobuf : https://github.com/caffe2/caffe2/blob/master/caffe2/proto/caffe2.proto

[3] Caffe2 Model Zoo : https://github.com/caffe2/caffe2/wiki/Model-Zoo

[4] Caffe2 Operations Catalogue : https://caffe2.ai/docs/operators-catalogue.html

[5] NNEF parser : https://github.com/KhronosGroup/NNEF-Tools