ONNX

ADDING NEW OPERATORS

VIEW FROM ONNX

Michal Karzynski (Intel)

ONNX GOALS

- common format for information exchange between frameworks
- flexible, expressive, but limited in scope for implementation
- does not include a reference implementation
- inherited first operator set from Caffe2 and CNTK, but not a defined style guide

GOOD OPERATOR CANDIDATE

- should appear in published and preferably widely used models
- preferably already supported by popular frameworks
- definition must be clear (unit-tests, reference implementation) and encompass existing implementations

ADDITIONAL CONSIDERATIONS

- If operator can be decomposed to existing primitives, it should be defined as a function
- If the operators can be split to new primitives, prefer those primitives instead and the operator as a function
- Prefer static attributes over dynamic input values
- Shape inference logic should be included

STRICT VS RELAXED OPERATOR DEFINITION

- Strict:
  - Resize operator – multiple interpolation strategies, each with different attributes
- Relaxed:
  - Random number generators – define output distribution, but not specific algorithm

CONSISTENT STYLE

- For ease of use, operators should follow a consistent style for inputs and attributes
- Style element examples:
  - broadcasting rules
  - *axes* (support for negative values)
  - default *stride* and *dilation*
  - default *keepdims*
  - supported input types
- Operator families should share same interface (e.g. reduction ops, pooling ops)

OPERATOR SETS

- Allow for evolution of operator set
- Operator with new features versions replace old ones
- Add complexity to the specification
- Require upgrade/downgrade path for existing models

TESTS, TESTS AND MORE TESTS

- Make life easier for implementors, converters, upgrade/downgrade adapters
- In ONNX tests are not well adapted to code review, area for improvement
- If you can add a reference implementation for tests, do it

CHALLENGES

- Deep learning is an evolving field, new operators appear frequently
- Scope of neural network applications is growing
- Pre- and postprocessing operations need to be included to take advantage of performance

THANK YOU!

- GitHub: https://github.com/onnx/onnx
- Slack channel: https://slack.lfai.foundation and join onnx-operators