ONNX

Converters SIG Updates

Guenther Schmuelling (Microsoft)
Kevin Chen (Nvidia)
Chin Huang (IBM)

PyTorch Exporter

- PyTorch 1.8 released on March 4.
- 12+ new torch operators supported for export.
- Support for ONNX Opset 13.
- Updated support for 10+ existing operators, updates to existing optimizations
- Support for named optional arguments in model's forward method.
- Improvements to ScriptModule export:
- Added shape and type inference in Torch graphs during export.
  - Enabled export of previously blocked scripting scenarios, e.g. Transpose.
  - Moved over to formal JIT APIs, such as *freeze_module*, in implementation.
- Tested support for several newly added PyTorch and HF hub models, both for inference and training.
- Complete list of changes gone into PyTorch 1.8 can be found here (see under ONNX header)
  - https://github.com/pytorch/pytorch/releases

sklearn-onnx

Changes since last October

- Support for opset 13 #536
- Implements option zipmap='columns' to split output of a classifier into multiple vectors #550
- Support double for linear models, VotingRegressor #561
- Add option return_std to BayesianRidge converter #580
- Support attributes of type numpy.matrix #598
- Support types int8, uint8 as inputs #544
- Bagging converters support max_features, bootstrap_features #515
- GaussianProcessClassifier #579
- Fix issue with CalibratedClassifier 0.24 #556
- Fix bug in RadiusNeighborsClassifier, RadiusNeighborsRegressor #596
- Fix issue HistGradientBoosting 0.24 #525
- Fix OneVsRestClassifier converter for multilabel classification #568
- Fix function update_registered_converter for custom classifier #606

sklearn-onnx

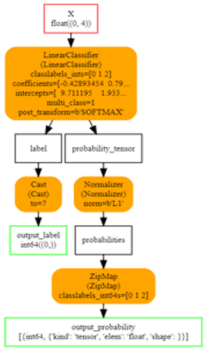
3 schemas for classifiers

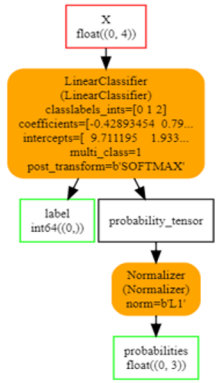
Implements option zipmap='columns' to split output of a classifier into multiple vectors #550

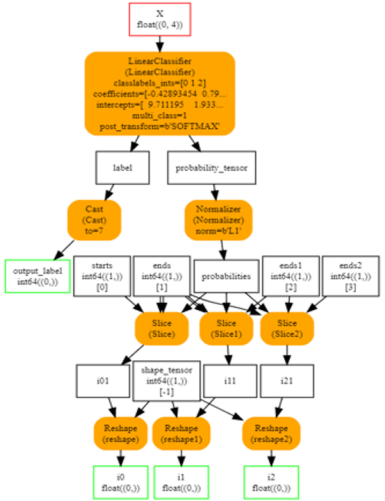
```
to_onnx(lr, X_train
        options={LogisticRegression: {'zipmap': 'columns'}})
```

```
to_onnx(lr, X_train
        options={LogisticRegression: {'zipmap': False}})
```

```
to_onnx(lr, X_train
```







tensorflow-onnx

- support for tensorflow-2.4 and opset-13
- support for tflite
  - direct conversion from tflite to onnx
  - quantized models are supported via QDQ
  - for an end to end example see here
- new, simplified python api to make use from keras easier, examples are here
- changes to avoid using gpu during conversion
- testing for tf/huggingface models, examples are here
- support for structured_outputs so output names are identical to keras models
- fixes

ONNX-TensorRT (TRT 7.2.3)

- Monthly container releases for ONNX-TensorRT (link)
  - Added support for 8 new operators in the past two releases
  - Added support for opset 13 definitions of existing operators
  - Open-sourced Python bindings

- TRT 8.0 - Planned for release at end of Q1
  - Focused on improved ONNX QDQ tooling and support
  - Improved OOTB performance for NLP models

- Tooling updates
  - ONNX-GraphSurgeon (link)
  - Polygraphy (link)
  - Pytorch QAT toolkit (link)

ONNX-Tensorflow

- Model zoo conversion and inference verification in CI and reporting to wiki, link
- ONNX opset 13 support
- Inference graph based training and examples
- Upcoming
  - ONNX opset 14 support
  - Investigate ONNX native training
  - Investigate TFLite and NHWC support

Backend roadmap

- Support channel last (NHWC) option in ONNX, at the model level, and/or certain ops and inputs
  - Motivation:
  - Optimized and simplified model conversion for backend frameworks and devices that are natively or preferably channel last
  - Flexibility in data format for model and ops
  - Affected ops: conv, pooling, resize, batch normalization
- Leverage Onnx model zoo models as standard test and verification for all backends
  - Motivation:
  - High quality conversion against state-of-art models
  - Standardized verification data and process to save time and development for individual converters
  - Affected projects: model zoo, backend converters