1000 и 1 боль при конвертации модели

Андрей Шадриков

Октябрь 2020

Зачем конвертировать?

• Фреймворки в исследовании:













• Фреймворки в продакшене:













Откуда возникают проблемы?

- Не полная поддержка слоёв.
- Разный формат слоёв (NonMaximumSuppression, Upsample).
- Конвертаторы не поддерживают последние версии фреймворков.
- Не всегда можно оценить скорость работы модели в проде.
- Некорректная сериализация моделей.

Есть же туториалы!

- У каждого фреймворка есть свои способы конвертировать модели.
- Часто рассматриваются только базовые случаи.
- Могут требовать старых версий фреймворков для стабильной работы.

Open Neural Network Exchange



- Инициатива от Facebook и Microsoft для конвертации между PyTorch, Caffe2, CNTK.
- Позже поддержали другие команды.
- Статический граф через protobuf.
- Версионность используемых операторов.

Что хорошо?

- Большая поддержка конвертации между фреймворками.
- Большое коммьюнити, поддерживающее редкие конвертаторы.
- Активное развитие.
- Версии opset'ов для хранения старых моделей.

Когда не надо использовать

- Разработка и прод в одной экосистеме (TensorFlow, MxNet).
- Конвертация через ONNX требует больше усилий, чем конвертация напрямую.
- Модель оптимизировалась под конкретный фреймворк (и это не ONNX).

Что хотелось бы

- Отладка моделей (вывод типов, размерностей).
- Мета-поля для слоёв.
- Более удобные инструменты работы с сериализованными моделями.

XKCD #927

KAK MHOXATCЯ CTAHDAPTЫ:

(СМ.: ЗАРЯДНЫЕ УСТРОЙСТВА, КОДИРОВКИ, МГНОВЕННЫЕ СООБЩЕНИЯ И Т.Д.)

СИТУАЦИЯ: ЕСТЬ 14 КОНКУРИРУЮЩИХ СТАНДАРТОВ. 14?! ABCYPA! HAM
HEOBXOAMMO
PASPABOTATЬ OANH
YHUBEPCANSHIN
CTAHAAPT HA BCE CNYUAN
ЖИЗНИ.

AA!

CNTYAUNA:
ECTS 15
KOHKYPUPYOUUNX
CTAHLAPTOB.

Альтернативы

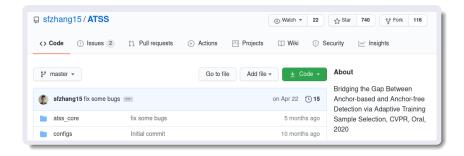
TensorFlow

- Очень популярный, большая поддержка.
- Можно не выходить за рамки экосистемы.
- Мало конвертаций.

MMdnn

- Ctapee ONNX.
- Поддерживается очень малым числом участников.
- Собственный внутренний формат не стандартизован.

PyTorch в ONNX



/home/andrey/tmp/atss/ATSS tmp/atss core/modeling/rpn/atss/inference.pv:63: UserWarning: This

/home/andrey/tmp/atss/ATSS_tmp/atss_core/modeling/backbone/fpn.py:62: TracerWarning: Converting a tensor to a Python integer might cause the trace to be incorrect. We can't record the data flow of Python values, so this value will be treated as a constant in the future. This means that the trace

Заключение

Не всё так просто

>nonzero()

overload of nonzero is deprecated:

Consider using one of the following signatures instead:

→nonzero(*, bool as_tuple) (Triggered internally at /pytorch/torch/csrc/utils/python arg parser.cpp:766.)

per_candidate_nonzeros = per_candidate_inds.nonzero()[top_k_indices. :]

```
might not generalize to other inputs!
 last_inner, size=(int(inner_lateral.shape[-2]), int(inner_lateral.shape[-1])),
/home/andrey/tmp/atss/ATSS tmp/atss core/structures/bounding box.py:21: TracerWarning:
torch as tensor results are registered as constants in the trace. You can safely ignore this
warning if you use this function to create tensors out of constant variables that would be the same
every time you call this function. In any other case, this might cause the trace to be incorrect.
 bbox = torch.as tensor(bbox, dtype=torch.float32, device=device)
/home/andrey/tmp/atss/ATSS tmp/atss core/structures/bounding box.pv:26: TracerWarning: Converting a
tensor to a Python boolean might cause the trace to be incorrect. We can't record the data flow of
Python values, so this value will be treated as a constant in the future. This means that the trace
might not generalize to other inputs!
 if bbox.size(-1) != 4:
/home/andrey/.local/lib/python3.8/site-packages/torch/tensor.py:452: RuntimeWarning: Iterating over
a tensor might cause the trace to be incorrect. Passing a tensor of different shape won't change
the number of iterations executed (and might lead to errors or silently give incorrect results).
 warnings.warn('Iterating over a tensor might cause the trace to be incorrect. '
/home/andrey/tmp/atss/ATSS_tmp/atss_core/modeling/rpn/atss/inference.py:61: TracerWarning:
Converting a tensor to a Python index might cause the trace to be incorrect. We can't record the
data flow of Python values, so this value will be treated as a constant in the future. This means
that the trace might not generalize to other inputs!
 per box cls, top k indices = per box cls.topk(per pre nms top n, sorted=False)
/home/andrev/tmp/atss/ATSS tmp/atss core/modeling/rpn/atss/inference.pv:107: TracerWarning:
Converting a tensor to a Python index might cause the trace to be incorrect. We can't record the
data flow of Python values, so this value will be treated as a constant in the future. This means
that the trace might not generalize to other inputs!
 number of detections = len(result)
```

/home/andrey/tmm/ates/ATSS tmm/ates core/modeling/rnm/ates/inforcase nyull6. TracerWarning.

Не всё так просто

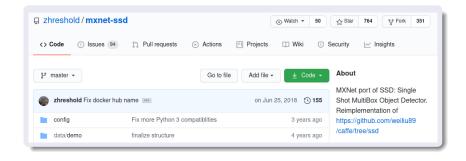
```
Traceback (most recent call last):
    File "convert.py", line 87, in <module>
        convert()
    File "convert.py", line 67, in convert
        sess = rt.InferenceSession(output_onnx, sess_options)
    File "/home/andrey/.local/lib/python3.8/site-packages/onnxruntime/capi/session.py", line 158, in ___init__
        self._load_model(providers or [])
    File "/home/andrey/.local/lib/python3.8/site-packages/onnxruntime/capi/session.py", line 177, in _load_model
        self._sess.load_model(providers)
    onnxruntime.capi.onnxruntime_pybind1l_state.InvalidGraph: [ONNXRuntimeError] : 10 : INVALID_GRAPH : This is an invalid model. Type Error: Type 'tensor(int64)' of input parameter (3925) of operator (Clip) in node (Clip_2738) is invalid.
```

Особенности конвертации из PyTorch

PyTorch при конвертации использует собственное внутреннее представление через trace:

- Операции должны выполняться над объектами torch. Tensor.
- In-place операции скорее всего создадут неверный граф (о чём предупреждает конвертатор).
- Control flow посчитается один раз.
- Проверять корректность модели лучше на других данных, что использовались при конвертации.

MxNet B TensorFlow



Конвертатор из MxNet поддерживает мало

Официальная документация

Prerequisites

To run the tutorial you will need to have installed the following python modules: - MXNet >= 1.3.0 - onnx v1.2.1 (follow the install guide)

Note: MXNet-ONNX importer and exporter follows version 7 of ONNX operator set which comes with ONNX v1.2.1.

В последней стабильной версии ONNX v1.7.0 версия операторов 12-я, и не все имеют forward-совместимость.

Можно обойтись тем, что есть

- Нам нужна на самом деле 10-я версия операторов.
- Почти все операторы имеют forward-совместимость.
- Слой для якорей и декодинга нужно переписать.
- Остаётся дело за малым нужно выкрутиться с NonMaximumSuppression.

Можно обойтись тем, что есть

- Нам нужна на самом деле 10-я версия операторов.
- Почти все операторы имеют forward-совместимость.
- Слой для якорей и декодинга нужно переписать.
- Остаётся дело за малым нужно выкрутиться с NonMaximumSuppression.
- Сконвертируем модель без него, а затем добавим в граф TensorFlow.

Что хотелось показать?

- Задача конвертации возникает достаточно часто.
- Неожиданные моменты могут потребовать много работы.
- Обсуждайте итоговые особенности продакшена заранее.
- ONNX хорошо использовать для сериализации моделей.
- Иногда проще менять граф в нужном фреймворке.

Что за рамками

- Редкие специфичные модели (Grid Sample).
- Рекурентные модели.
- Не нейронки.
- Отладка и проверка корректности моделей.
- Мобильные и другие платформо-специфичные фреймворки.

Спасибо за внимание!

