東南大學
SOUTHEAST UNIVERSITY

TensorRT Hackathon2022 中期检查

😊 SmilingFaces 🙂

2022-6-13

前期工作

在转换初期遭遇reflect padding问题

- 暂时跳过这一数据预处理过程，使用手工定义的适宜大小图像进行网络主体的trt转换
- 后期可以参照godv的方案融入进主体网络做转换

前期工作

在转换初期遭遇reflect padding问题

- 暂时跳过这一数据预处理过程，使用手工定义的适宜大小图像进行网络主体的trt转换
- 后期可以参照godv的方案融入进主体网络做转换

FP32一路通顺，不需要任何surgeon工作

FP16精度下降较为严重，存在溢出问题

前期工作

在转换初期遭遇reflect padding问题

- 暂时跳过这一数据预处理过程，使用手工定义的适宜大小图像进行网络主体的trt转换
- 后期可以参照godv的方案融入进主体网络做转换

FP32一路通顺，不需要任何surgeon工作

FP16精度下降较为严重，存在溢出问题

编写了一个test_perf.py脚本便于查看性能表现

```
==== pytorch ====
use cuda & cudnn for acceleration!
the gpu id is: [4, 5, 6, 7]
load pretrained model: /root/trt-elan/weights/model_x4_4
torch.Size([1, 3, 80, 80])
torch.Size([1, 3, 320, 320])
Pytorch time:  40.87517023086548
Pytorch throughout:  24.464729916767034
```

```
==== ONNX ====
****Average diff between onnx and pytorch****

bs: Batch Size
width: Image width
height: Image height
lt: Latency (ms)
tp: throughput (image/s)
max-a0: maximum of absolute difference of output 0
med-a0: median of absolute difference of output 0
mea-a0: mean of absolute difference of output 0
max-r0: maximum of absolute difference of output 0
med-r0: median of relative difference of output 0
mea-r0: mean of relative difference of output 0
----+-----+------+--------+---------+---------+---------+---------+---------+---------+---------+-------------
  bs|width|height|      lt|       tp|   max-a0|   med-a0|   mea-a0|   max-r0|   med-r0|   mea-r0| output check
----+-----+------+--------+---------+---------+---------+---------+---------+---------+---------+-------------

   1,    80,    80,  36.682,2.726e+01,1.294e-02,7.782e-04,9.571e-04,2.255e+01,8.776e-04,5.548e-03
```

前期工作

在转换初期遭遇reflect padding问题

- 暂时跳过这一数据预处理过程，使用手工定义的适宜大小图像进行网络主体的trt转换
- 后期可以参照godv的方案融入进主体网络做转换

FP32一路通顺，不需要任何surgeon工作

FP16精度下降较为严重，存在溢出问题

编写了一个test_perf.py脚本便于查看性能表现

```
==== pytorch ====
use cuda & cudnn for acceleration!
the gpu id is: [4, 5, 6, 7]
load pretrained model: /root/trt-elan/weights/model_x4_4
torch.Size([1, 3, 80, 80])
torch.Size([1, 3, 320, 320])
Pytorch time:  40.87517023086548
Pytorch throughout:  24.464729916767034
```

```
==== TensorRT fp32 ====
Succeeded loading engine!
EngineBinding0-> (1, 3, 80, 80) DataType.FLOAT
EngineBinding1-> (1, 3, 320, 320) DataType.FLOAT
TRT time:  34.738571643829346
****Average diff between trt fp32 and pytorch****

bs: Batch Size
width: Image width
height: Image height
lt: Latency (ms)
tp: throughput (image/s)
max-a0: maximum of absolute difference of output 0
med-a0: median of absolute difference of output 0
mea-a0: mean of absolute difference of output 0
max-r0: maximum of absolute difference of output 0
med-r0: median of relative difference of output 0
mea-r0: mean of relative difference of output 0
----+-----+------+--------+---------+---------+---------+---------+---------+---------+---------+-------------
  bs|width|height|      lt|       tp|   max-a0|   med-a0|   mea-a0|   max-r0|   med-r0|   mea-r0| output check
----+-----+------+--------+---------+---------+---------+---------+---------+---------+---------+-------------

   1,    80,    80,  34.739,2.879e+01,3.286e-02,4.463e-03,5.385e-03,1.301e+02,4.942e-03,3.130e-02
```

前期工作

在转换初期遭遇reflect padding问题

- 暂时跳过这一数据预处理过程，使用手工定义的适宜大小图像进行网络主体的trt转换
- 后期可以参照godv的方案融入进主体网络做转换

FP32一路通顺，不需要任何surgeon工作

FP16精度下降较为严重，存在溢出问题

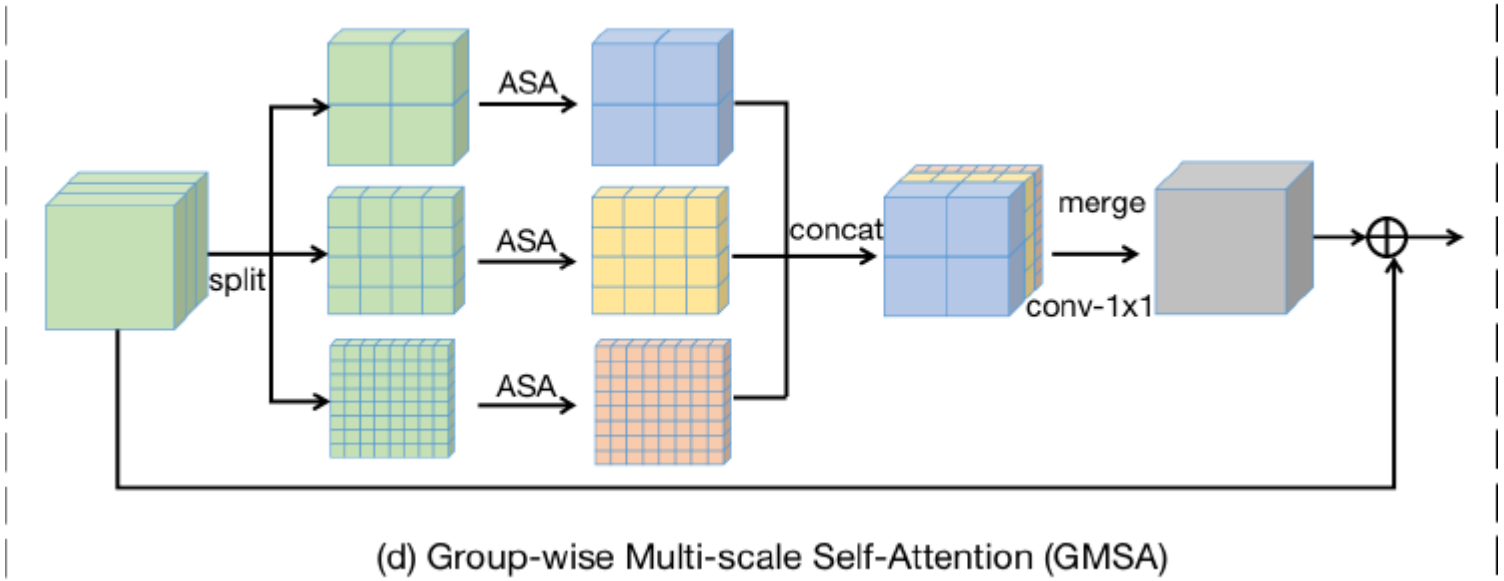
编写了一个test_perf.py脚本便于查看性能表现

```
==== pytorch ====
use cuda & cudnn for acceleration!
the gpu id is: [4, 5, 6, 7]
load pretrained model: /root/trt-elan/weights/model_x4_4
torch.Size([1, 3, 80, 80])
torch.Size([1, 3, 320, 320])
Pytorch time:  40.87517023086548
Pytorch throughout:  24.464729916767034
```

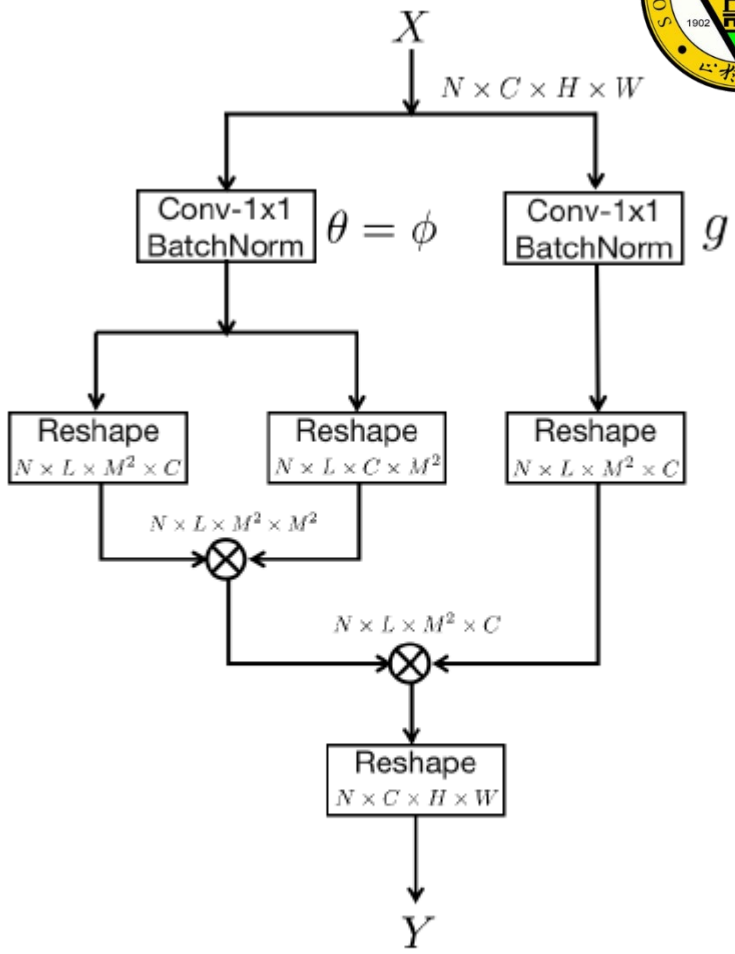
```
==== TensorRT fp16 ====
Succeeded loading engine!
EngineBinding0-> (1, 3, 80, 80) DataType.FLOAT
EngineBinding1-> (1, 3, 320, 320) DataType.FLOAT
TRT time:  20.738341808319092
****Average diff between trt fp16 and pytorch****

bs: Batch Size
width: Image width
height: Image height
lt: Latency (ms)
tp: throughput (image/s)
max-a0: maximum of absolute difference of output 0
med-a0: median of absolute difference of output 0
mea-a0: mean of absolute difference of output 0
max-r0: maximum of absolute difference of output 0
med-r0: median of relative difference of output 0
mea-r0: mean of relative difference of output 0
----+-----+------+--------+---------+---------+---------+---------+---------+---------+---------+------------
  bs|width|height|      lt|       tp|   max-a0|   med-a0|   mea-a0|   max-r0|   med-r0|   mea-r0| output check
----+-----+------+--------+---------+---------+---------+---------+---------+---------+---------+------------

   1,    80,    80,  20.738,4.822e+01,8.088e-01,1.154e-01,1.363e-01,3.931e+03,1.270e-01,7.861e-01
```

未来工作

写出ASA或者GMSA的Plugin



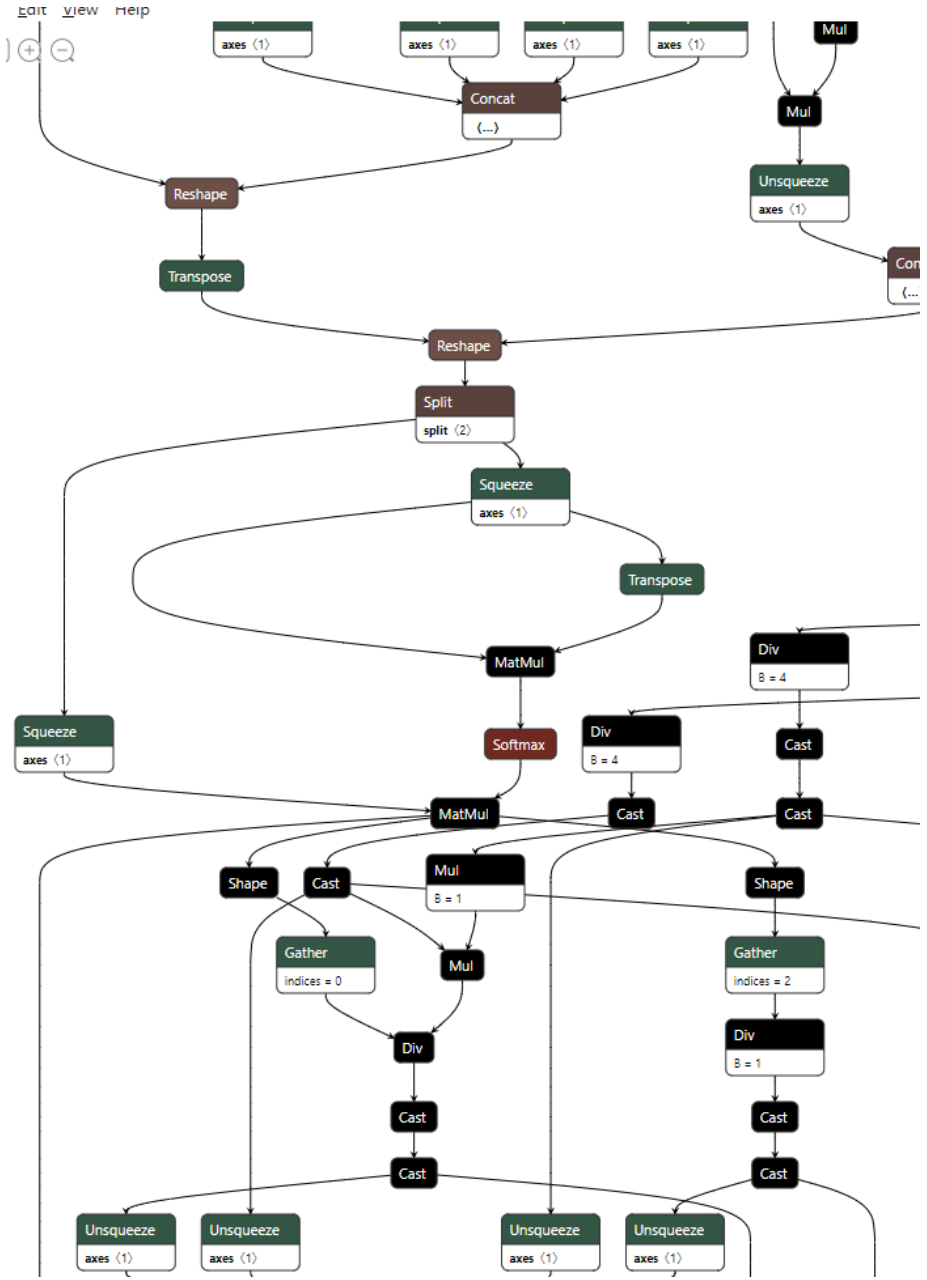
(d) Group-wise Multi-scale Self-Attention (GMSA)



(e) Accelerated Self-Attention (ASA)

未来工作

写出ASA或者GMSA的Plugin



未来工作

API-based Constructor

【去年未成功 – 权重维度对齐问题】

```python
class TRT_Constructor:
    def __init__(self, network: trt.tensorrt.INetworkDefinition, cuda=False):
        self.network = network
        self.cuda = cuda

    def MaxPool2d(self, pool: nn.MaxPool2d, x: trt.tensorrt.ITensor):
        stride, padding, window_size = pool.stride, pool.padding, pool.kernel_size
        tlist = [stride, padding, window_size]
        tlist = [[a, a] if type(a) is int else list(a) for a in tlist]
        tlist = [trt.tensorrt.DimsHW(a) for a in tlist]
        stride, padding, window_size = tlist
        if type(window_size) is int:
            window_size = [window_size, window_size]
        else:
            window_size = list(window_size)
        y = self.network.add_pooling(
            input=x,
            type=trt.PoolingType.MAX,
            window_size=window_size
        )
        y.stride = stride
        y.padding = padding
        return y.get_output(0)

    def Conv2d(self, conv: nn.Conv2d, x: trt.tensorrt.ITensor):
        if self.cuda:
            y = self.network.add_convolution(
                input=x,
                num_output_maps=conv.out_channels,
                kernel_shape=conv.kernel_size,
                kernel=conv.weight.cpu().numpy(),
                bias=conv.bias.cpu().numpy() if conv.bias is not None else None
            )
        else:
            y = self.network.add_convolution(
```

未来工作

量化

1. 搜索出FP16精度问题的位置，使用strict type优化这一问题

2. 实现INT8量化
【去年未实现】

未来工作

Bug报告？关于reflect padding不能良好转换的问题

阅读TensorRT-ONNXparser相关代码，找出问题。

未来工作

工作时间线安排

6.13 ~ 6.19:　完成ASA plugin的书写

完成API搭建工作

6.20 ~ 6.25:　解决fp16精度问题

实现int8量化（6.23），并解决精度问题（6.25）

6.25 ~ 6.26:　完成性能测试，并撰写性能测试报告

6.27　　:　书写完整报告并公开repo

東南大學
SOUTHEAST UNIVERSITY

Thank you!

2022.6.13

心於至善