

TensorFlow Parser Scope 融合规则参考 CANN

6.3.RC2

文档版本 01
发布日期 2023-07-26



版权所有 © 华为技术有限公司 2023。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

目 录

1 简介.....	1
2 融合规则说明.....	3
2.1 ScopeLayerNormPass.....	3
2.2 ScopeLayerNormGradPass.....	5
2.3 ScopeBasicLSTMCellPass.....	7
2.4 ScopeDynamicLSTMPass.....	8
2.5 ScopeClipBoxesPass.....	10
2.6 ScopeROIAlignPass.....	11
2.7 ScopeRpnProposalsPass.....	12
2.8 ScopeFastrcnnPredictionsPass.....	15
2.9 ScopeDecodeBboxPass.....	17
2.10 ScopeToAbsoluteBBoxPass.....	20
2.11 ScopeNormalizeBBoxPass.....	22
2.12 ScopeDecodeBboxV2Pass.....	25
2.13 ScopeBatchMultiClassNMSPass.....	27
2.14 ScopeKeepRatioResizeBilinearPass.....	29
2.15 ScopeBatchMultiClassNonMaxSuppressionPass.....	31
2.16 ScopeDynamicGRUPass.....	34
2.17 ScopeDynamicRNNPass.....	35

1 简介

概述

Scope融合是一种基于Scope来进行融合的能力，把Scope内的多个小算子替换为一个
大算子或多个算子组合，以实现效率的提升。

本文主要介绍内置的Scope融合规则，同时开放Scope融合规则开发接口供用户自定义，具体请参考《[TensorFlow Parser Scope融合规则开发指南](#)》。

通用和定制化融合规则

融合规则通常分为通用和定制化融合规则两类：

- 通用融合规则（General）：各网络通用的scope融合规则；默认生效，不支持用户指定失效。
- 定制化融合规则（Non-General）：特定网络适用的scope融合规则；默认不生效，用户可以指定需要生效的融合规则，定制化融合规则生效方式可以参考如下生效方式。

表 1-1 定制化融合规则生效方式

场景	生效方式
离线推理场景下，使用 离线模型转换工具编译 TensorFlow原始模型	通过模型转换命令行参数 enable_scope_fusion_passes 指定需要生效的融合 规则，多个用“,”分隔： <code>--enable_scope_fusion_passes = DecodeBboxV2ScopeFusionPass</code>
离线推理场景下，解析 TensorFlow原始模型	通过 aclgrphParseTensorFlow 接口解析TensorFlow 原始模型时，通过 ENABLE_SCOPE_FUSION_PASSES 参数指定需要生 效的融合规则，多个用“,”分隔： <code>{ge::AscendString(ge::ir_option::ENABLE_SCOPE_FUSION_PASSES) , ge::AscendString("DecodeBboxV2ScopeFusionPass")},</code>

场景	生效方式
训练或在线推理场景下，在TensorFlow框架内执行	<p>通过TensorFlow框架运行配置参数 enable_scope_fusion_passes指定需要生效的融合规则，多个用“,”分隔：</p> <pre>import tensorflow as tf from npu_bridge.estimator import npu_ops from tensorflow.core.protobuf.rewriter_config_pb2 import RewriterConfig config = tf.ConfigProto() custom_op = config.graph_options.rewrite_options.custom_optimizers.add() custom_op.name = "NpuOptimizer" custom_op.parameter_map["use_off_line"].b = True custom_op.parameter_map["enable_scope_fusion_passes"].s = tf.compat.as_bytes("DecodeBboxV2ScopeFusionPass") config.graph_options.rewrite_options.remapping = RewriterConfig.OFF with tf.Session(config=config) as sess: sess.run()</pre>

2 融合规则说明

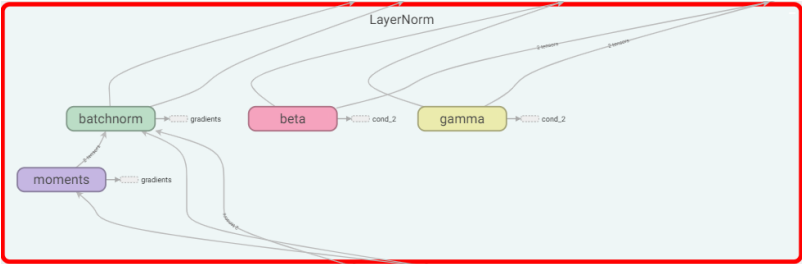
[ScopeLayerNormPass](#)
[ScopeLayerNormGradPass](#)
[ScopeBasicLSTMCellPass](#)
[ScopeDynamicLSTMPass](#)
[ScopeClipBoxesPass](#)
[ScopeROIAlignPass](#)
[ScopeRpnProposalsPass](#)
[ScopeFastrcnnPredictionsPass](#)
[ScopeDecodeBboxPass](#)
[ScopeToAbsoluteBBoxPass](#)
[ScopeNormalizeBBoxPass](#)
[ScopeDecodeBboxV2Pass](#)
[ScopeBatchMultiClassNMSPass](#)
[ScopeKeepRatioResizeBilinearPass](#)
[ScopeBatchMultiClassNonMaxSuppressionPass](#)
[ScopeDynamicGRUPass](#)
[ScopeDynamicRNNPass](#)

2.1 ScopeLayerNormPass

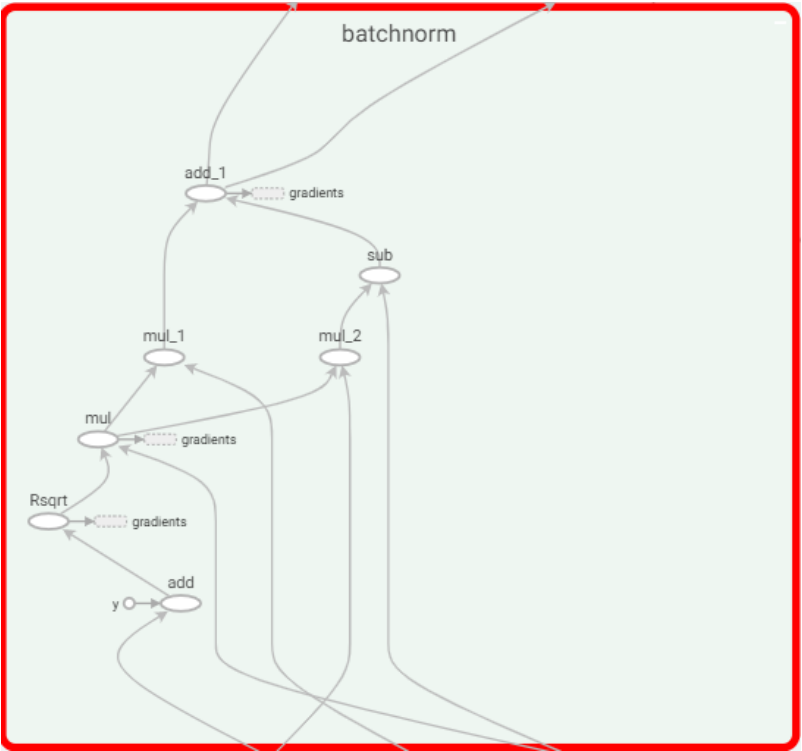
功能说明

将tf.layernorm生成的layernorm/batchnorm和layernorm/moments这两个Scope，融合为LayerNorm算子。

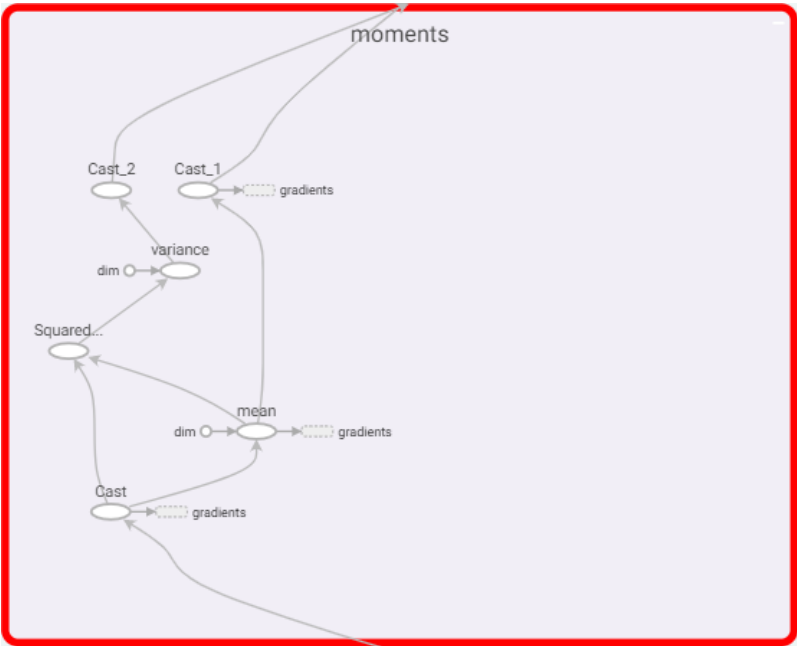
Scope 详情



batchnorm展开:



moments展开:



融合后的算子原型

LayerNorm，具体请参见《[算子清单](#)》。

融合对应关系

当有Cast节点时，首个cast的输入，作为融合后的第一个输入x。

Mul节点的gamma输入作为融合后后的第2个输入gamma。

最后一个Add节点beta输入作为融合后的第3个输入beta。

第4个begin_norm_axis使用默认值1。

第5个begin_param_axis使用默认值-1。

适用网络

Bert

融合规则类型

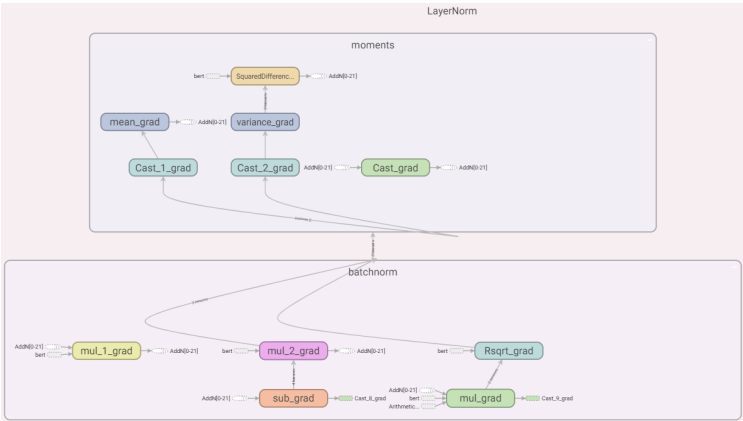
通用融合规则

2.2 ScopeLayerNormGradPass

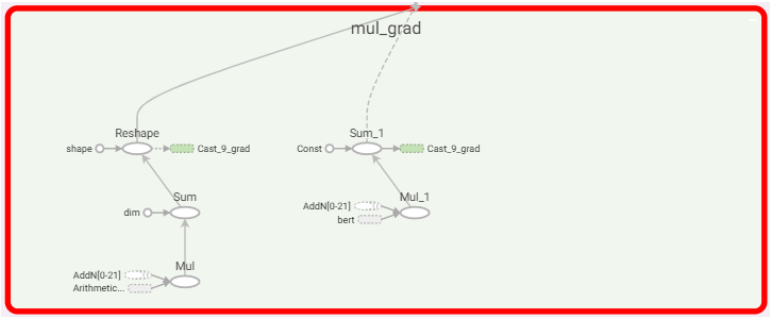
功能说明

将tf.layernorm的反向Scope融合为LayerNormGrad算子。

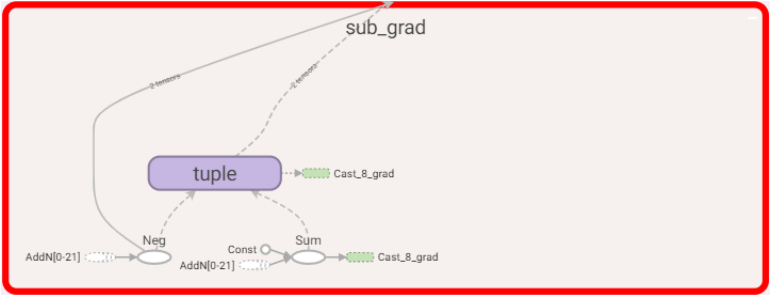
Scope 详情



`mul_grad`展开:



`sub_grad`展开:



融合后的算子原型

`LayerNormGrad`，具体请参见《[算子清单](#)》。

融合对应关系

- LayerNorm反向的输入作为融合后的第一个输入dy。
- LayerNorm正向的输入作为融合后的第2个输入x。
- 正向的第3个输出variance作为反向的第3个输入variance。
- 正向的第2个输出mean作为反向的第3个输入mean。
- 正向的第2个输入gamma作为反向的第4个输入gamma。

反向的第1个输出连接到反向图最后的addN节点的输出上。

反向的第2个输出gamma_backprop连接到mul_grad中的到cast节点的Mul的输出上。

反向的第3个输出beta_backprop连接到sub_grad中的到cast节点的Sum的输出上。

适用网络

Bert

融合规则类型

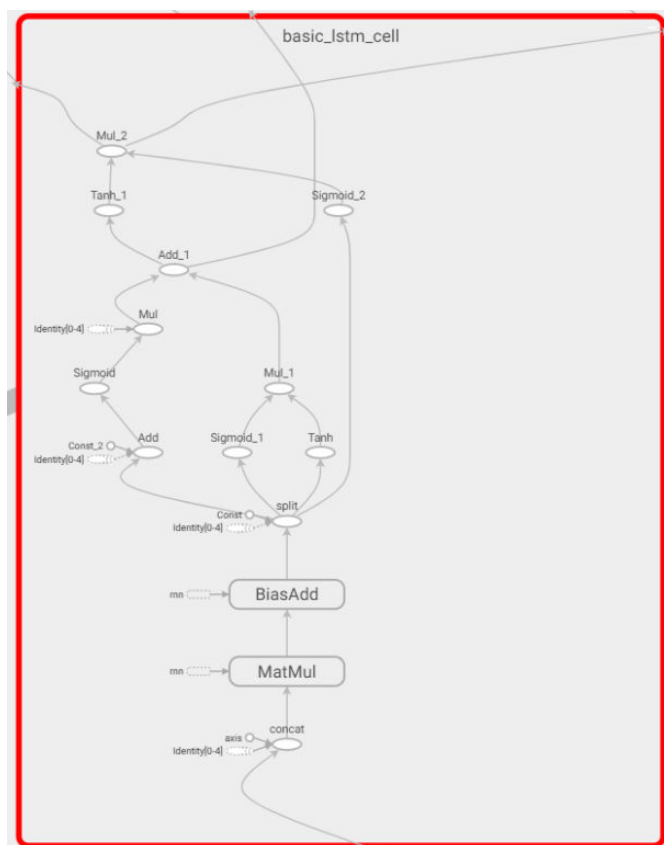
通用融合规则

2.3 ScopeBasicLSTMCellPass

功能说明

将tf.nn.rnn_cell.BasicLSTMCell生成的Scope内的小算子组合融合为BasicLSTMCell算子。

Scope 详情



融合后的算子原型

BasicLSTMCell，具体请参见《[算子清单](#)》。

融合对应关系

- Concat算子的第1个输入作为融合后的第1个输入x。
- Concat算子的第2个输入作为融合后的第2个输入h。
- Mul算子的第1个输入作为融合后的第3个输入c。
- MatMul算子的第2个输入作为融合后的第4个输入w。
- BiasAdd算子的第2个输入作为融合后的第5个输入b。
- Add_1的输出作为融合后的第0个输出ct。
- Mul_2的输出作为融合后的第1个输出ht。

适用网络

使用BasicLSTMCell单cell的非循环推理网络，例如NMT推理网络。

融合规则类型

定制化融合规则

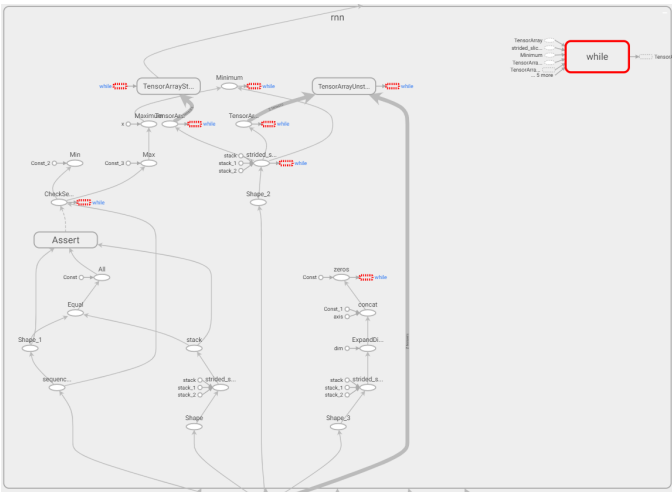
2.4 ScopeDynamicLSTMPass

功能说明

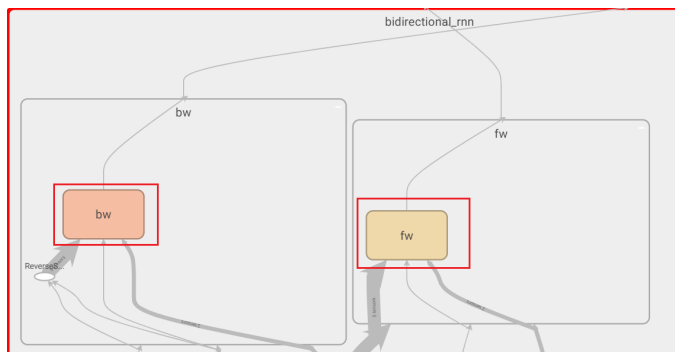
将tf.nn.dynamic_rnn或tf.nn.bidirectional_dynamic_rnn生成的Scope内的小算子组合融合为DynamicLSTM算子。当前仅支持Cell结果为BasicLSTMCell的循环场景，且仅支持个别shape。

Scope 详情

dynamic_rnn对应的Scope结构：



或者bidirectional_dynamic_rnn中对应的两个dynamic_rnn，分别为FW和BW：



融合后的算子原型

DynamicLSTM，具体请参见《[算子清单](#)》。

融合对应关系

当time_major为False时：

rnn/transpose节点的第1个输入作为融合后的第1个输入x。

rnn/while/basic_lstm_cell/MatMul/Enter节点的输入作为融合后的第2个输入w。

rnn/while/basic_lstm_cell/BiasAdd/Enter节点的输入作为融合后的第3个输入b。

rnn/transpose_1节点的输出作为融合后的输出output_h。

当time_major为True时：

rnn/TensorArrayUnstack/TensorArrayScatter/TensorArrayScatterV3节点的第3个输入作为融合后的第1个输入x。

rnn/while/basic_lstm_cell/MatMul/Enter节点的输入作为融合后的第2个输入w。

rnn/while/basic_lstm_cell/BiasAdd/Enter节点的输入作为融合后的第3个输入b。

rnn/TensorArrayStack/TensorArrayGatherV3节点的输出作为融合后的输出output_h。

说明

上图的Scope是以time_major为True举例的。

适用网络

使用dynamic_rnn且单cell为BasicLSTMCell的推理网络。

融合规则类型

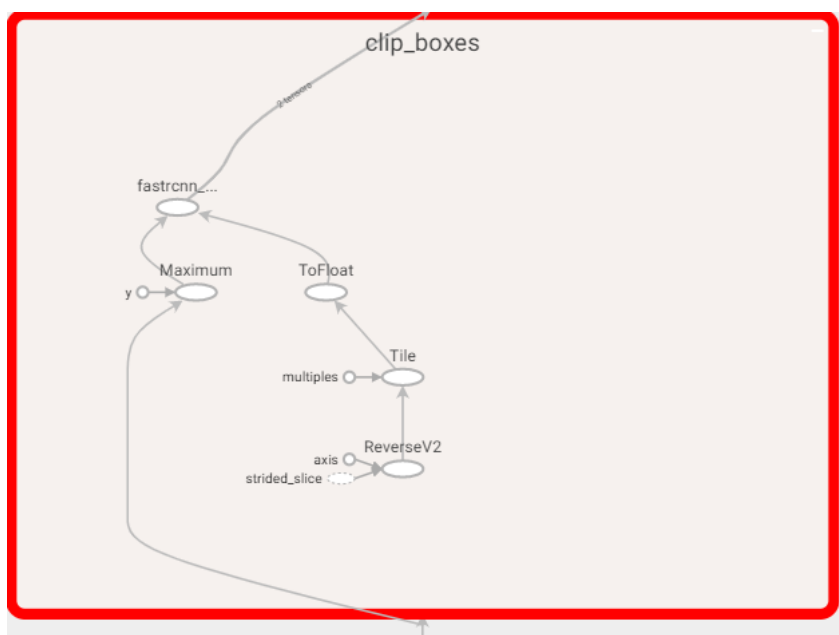
定制化融合规则

2.5 ScopeClipBoxesPass

功能说明

将clip_boxes Scope融合为ClipBoxes算子。Scope内包括tf.Maximum、tf.ReverseV2、tf.Tile和tf.Minimum这四个算子，不包含Gather_2、TopKV2、Reshape_2、Split、Greater、Squeeze、Gather、boolean_mask、decode_bbox_target等算子。

Scope 详情



融合后的算子原型

ClipBoxes，具体请参见《[算子清单](#)》。

融合对应关系

- clip_boxes/Maximum输入作为融合后的第一个输入boxes_input。
- clip_boxes/ReverseV2输入作为融合后的第二个输入im_info。
- clip_boxes/fastrcnn_all_boxes（Minimum）输出作为融合后的输出boxes_output。
- clip_boxes/ReverseV2的输出作为clip_boxes/Tile的输入。
- clip_boxes/Tile的输出作为clip_boxes/ToFloat的输入。
- clip_boxes/ToFloat的输入作为clip_boxes/fastrcnn_all_boxes的第二个输入。
- clip_boxes/Maximum的输出作为clip_boxes/fastrcnn_all_boxes的第一个输入。

适用网络

2D-H1

融合规则类型

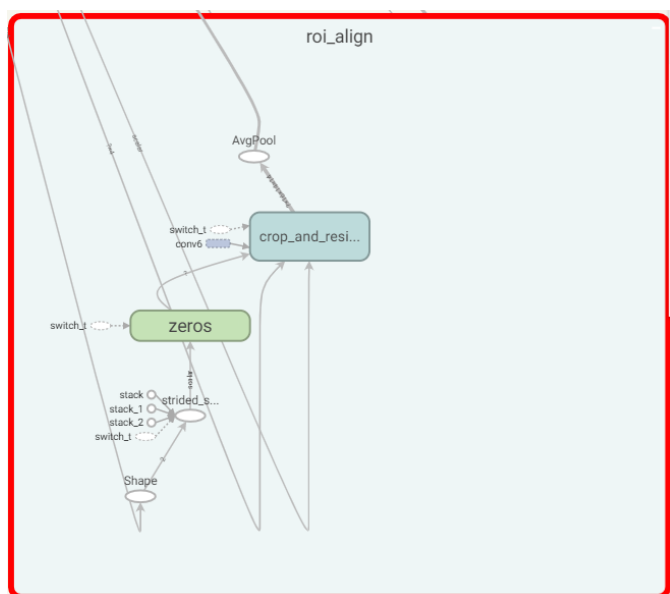
定制化融合规则

2.6 ScopeROIAlignPass

功能说明

将tf.AvgPool和tf.image.CropAndResize这两个Scope融合为ROIAlign算子，不包含Merge算子。

Scope 详情



融合后的算子原型

ROIAlign，具体请参见《[算子清单](#)》。

融合对应关系

crop_and_resize/Shape/Switch输入作为融合后的第一个输入features。

Shape输入作为融合后的第二个输入rois。

Avgpool输出作为融合后的输出y。

Shape的输出作为StridedSlice的输入。

StridedSlice的输出作为zeros的输入。

crop_and_resize/Shape/Switch的输出作为 crop_and_resize/Shape/Shape和 crop_and_resize/transpose的输入。

crop_and_resize/Shape/Shape的输出作为 crop_and_resize/strided_slice的输入。

crop_and_resize/strided_slice的输出作为crop_and_resize/transform_fpcoor_for_tf的输入。

crop_and_resize/transform_fpcoor_for_tf和crop_and_resize/transpose的输出作为CropAndResize的输入。

crop_and_resize/CropAndResize的输出作为crop_and_resize/transpose_1的输入。

crop_and_resize/transpose_1的输出作为AvgPool的输入。

适用网络

2D-H1

融合规则类型

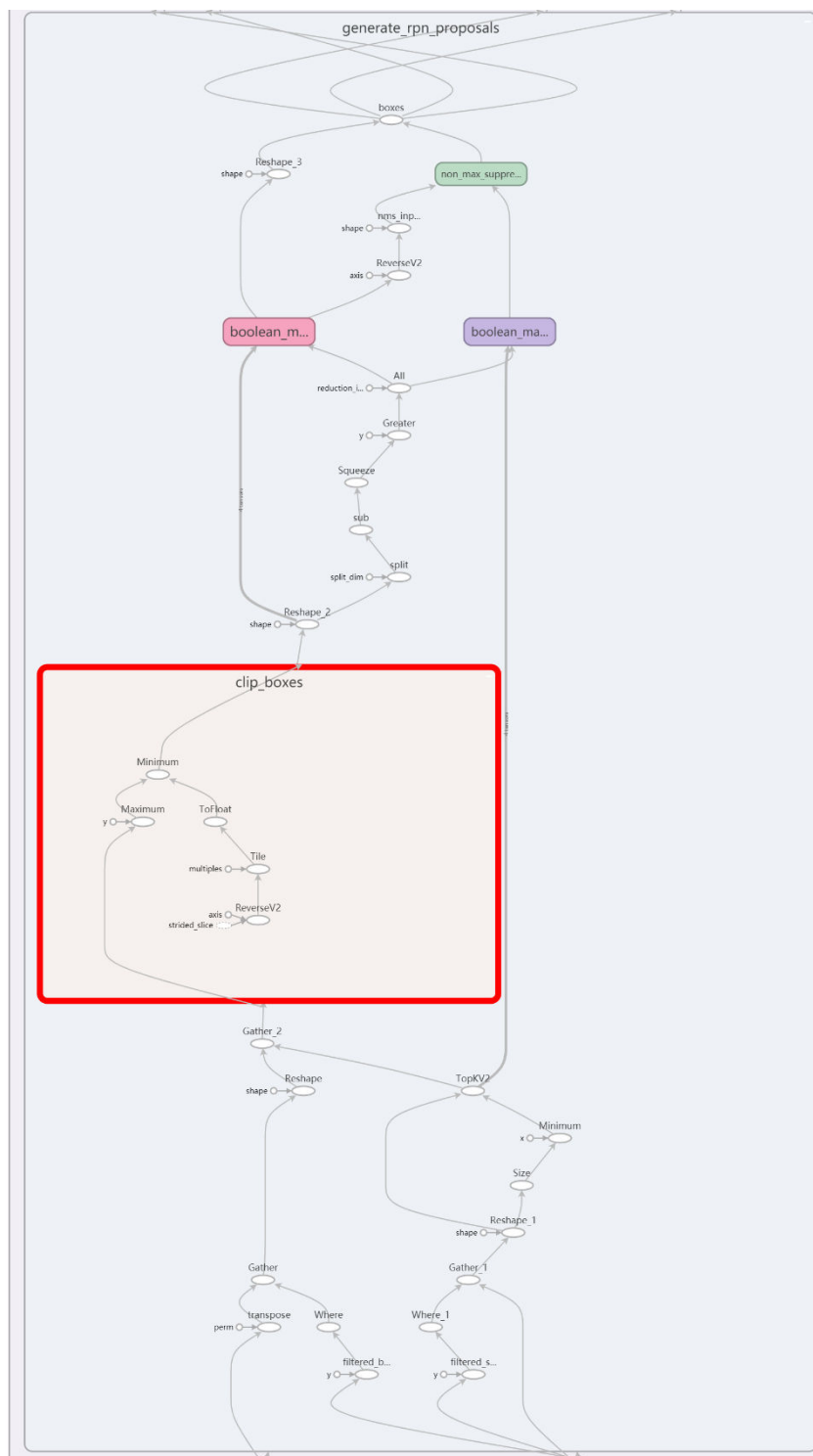
定制化融合规则

2.7 ScopeRpnProposalsPass

功能说明

将generate_rpn_proposals Scope融合为RpnProposals算子。Scope内包括tf.NonMaxSuppressionV2算子、tf.TopKV2算子、4的倍数个tf.Where算子、6的倍数个tf.Gather算子，不包含ExpandDims、Switch和transpose算子。

Scope 详情



融合后的算子原型

RpnProposals, 具体请参见《[算子清单](#)》。

融合对应关系

Transpose的输入作为融合后的第一个输入rois。

filtered_boxes、filtered_scores和Gather_1的输入作为融合后的第二个输入cls_bg_prob。

clip_boxes/ReverseV2输入作为融合后的第三个输入img_info。

boxes的输出作为融合后的输出sorted_box。

filtered_boxes的输出作为Where的输入。

transpose和Where的输出作为Gather的输入。

Gather的输出作为Reshape的输入。

filtered_scores的输出作为Where_1的输入。

Where_1的输出作为Gather_1的输入。

Gather_1的输出作为 Reshape_1的输入。

Reshape_1的输出 作为TopK V2和 size的输入。

Size的输出作为Minimum的输入。

Minimum的输出作为TopKV2的输入。

TopKV2的输出作为Gather_2和boolean_mask_1的输入。

Gather_2的输出作为clip_boxes/Maximum的输入。

clip_boxes/ReverseV2的输出作为clip_boxes/Tile的输入。

clip_boxes/Tile的输出作为clip_boxes/ToFloat的输入。

clip_boxes/Maximum和clip_boxes/ToFloat的输出作为clip_boxes/Minimum的输入。

clip_boxes/Minimum的输出作为Reshape_2的输入。

Reshape_2的输出作为boolean_mask和split的输入。

Split的输出作为sub的输入。

Sub的输出作为Squeeze的输入。

Squeeze的输出作为Greater的输入。

Greater的输出作为All的输入。

All的输出作为boolean_mask和boolean_mask_1的输入。

boolean_mask的输出作为Reshape_3和ReverseV2的输入。

ReverseV2的输出作为nms_input_boxes的输入。

nms_input_boxes的输出作为non_max_suppression的输入。

boolean_mask_1的输出作为non_max_suppression的输入。

non_max_suppression和Reshape_3的输出作为Boxes的输入。

适用网络

2D-H1

融合规则类型

定制化融合规则

2.8 ScopeFastrcnnPredictionsPass

功能说明

将fastrcnn_predictions Scope融合为FastrcnnPredictions算子。Scope内包括2的倍数个tf.TopKV2算子、3的倍数个tf.Where算子、tf. NonMaxSuppressionV2算子、tf.Less算子、tf.LoopCond算子等，不包含ExpandDims、clip_boxes和decode_bbox_target。

Scope 详情

融合后的算子原型

FastrcnnPredictions, 具体请参见《[算子清单](#)》。

融合对应关系

fastrcnn_predictions/transpose和fastrcnn_predictions/GatherNd的输入作为融合后的输入rois。

fastrcnn_predictions/strided_slice的输入作为融合后的输入score。

fastrcnn_predictions/TopKV2的输出作为融合的输出sorted_rois。

fastrcnn_predictions/GatherNd的输出作为融合后的输出sorted_scores。

fastrcnn_predictions/Add的输出作为融合后的输出sorted_classes。

fastrcnn_predictions/strided_slice的输出作为fastrcnn_predictions/transpose_1的输入。

fastrcnn_predictions/transpose_1的输出作为fastrcnn_predictions/map和fastrcnn_predictions/boolean_mask的输入。

astrcnn_predictions/map的输出作为fastrcnn_predictions/Where的输入。

fastrcnn_predictions/Where的输出作为fastrcnn_predictions/Gather的输入。

fastrcnn_predictions/boolean_mask的输出作为fastrcnn_predictions/Size和fastrcnn_predictions/TopKV2的输入。

fastrcnn_predictions/Size的输出作为fastrcnn_predictions/Minimum的输入。

fastrcnn_predictions/Minimum的输出作为fastrcnn_predictions/TopKV2的输入。

fastrcnn_predictions/TopKV2作为融合后的输出（sorted_rois）和fastrcnn_predictions/Gather的输入。

fastrcnn_predictions/Gather的输出作为fastrcnn_predictions/filtered_indices的输入。

fastrcnn_predictions/filtered_indices的输出作为fastrcnn_predictions/GatherNd和fastrcnn_predictions/ToFloat的输入。

fastrcnn_predictions/ToFloat的输出作为fastrcnn_predictions/strided_slice_1的输入。

fastrcnn_predictions/strided_slice_1的输出作为fastrcnn_predictions/Add的输入。

适用网络

2D-H1

融合规则类型

定制化融合规则

2.9 ScopeDecodeBboxPass

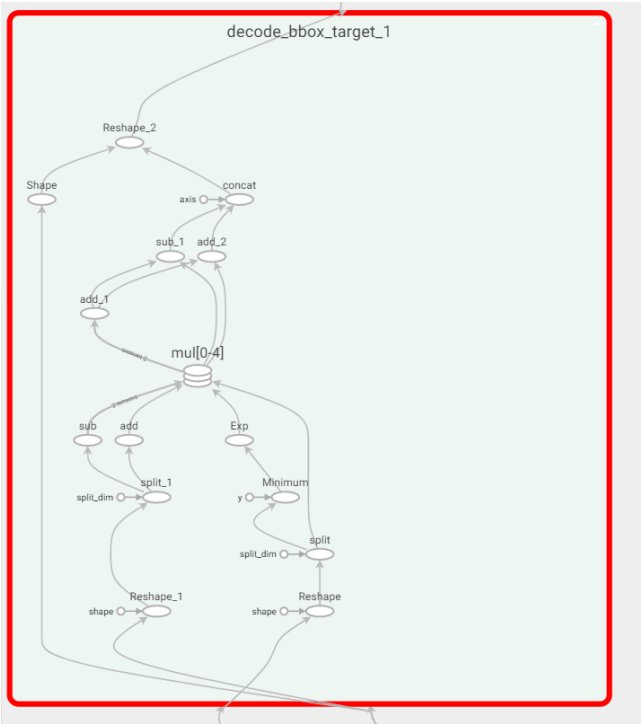
功能说明

将含有如下算子的Scope融合为DecodeBbox算子。Scope内包括：3的倍数个tf.Reshape算子、2的倍数个tf.Split算子、tf.Minimum算子、3的倍数个tf.Add算子、tf.ConcatV2算子、2的倍数个tf.Sub算子，不包含Greater、Squeeze、Gather_2、TopKV2、boolean_mask等算子。

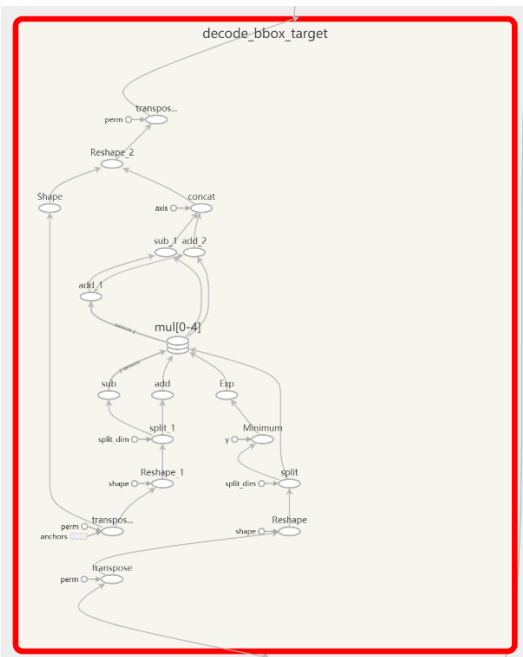
Scope 详情

根据是否包含Transpose算子，有两种Scope，如下所示：

不含Transpose算子:



含Transpose算子:



融合后的算子原型

DecodeBbox, 具体请参见《[算子清单](#)》。

融合对应关系

- 不含Transpose算子:

Reshape的输入作为融合后的输入box_predictions。
Shape和Reshape_1的输入作为融合后的输入anchors。
Reshape_2的输出为融合后的输出decoded_boxes。
Reshape的输出作为Split的输入。
Split的输出作为Minimum和Mul的输入。
Minimum的输出作为Exp的输入。
Exp的输出作为mul的输入。
Reshape_1的输出作为split_1的输入。
split_1的输出作为Sub和Add输入。
Sub和Add的输出作为Mul的输入。
Mul的输出作为Add_1、Sub_1、Add_2的输入。
Add_1的输出作为Sub_1和Add_2的输入。
Sub_1和Add_2的输出作为Concat的输入。
Shape和Concat的输出作为Reshape_2的输入。
融合算子排除Greater、Squeeze、Gather_2、TopKV2、boolean_mask等算子。

- 含Transpose算子：
transpose的输入作为融合后的输入box_predictions。
transpose_1的输入作为融合后的输入anchors。
transpose_2的输出为融合后的输出decoded_boxes。
transpose的输出作为Reshape的输入。
Reshape的输出作为Split的输入。
Split的输出作为Minimum和Mul的输入。
Minimum的输出作为Exp的输入。
Exp的输出作为mul的输入。
transpose_1的输出作为Reshape_1和Shape的输入。
Reshape_1的输出作为split_1的输入。
split_1的输出作为Sub和Add输入。
Sub和Add的输出作为Mul的输入。
Mul的输出作为Add_1、Sub_1、Add_2的输入。
Add_1的输出作为Sub_1和Add_2的输入。
Sub_1和Add_2的输出作为Concat的输入。
Shape和Concat的输出作为Reshape_2的输入。
Reshape_2的输出作为transpose_2的输入。

适用网络

2D-H1

融合规则类型

定制化融合规则

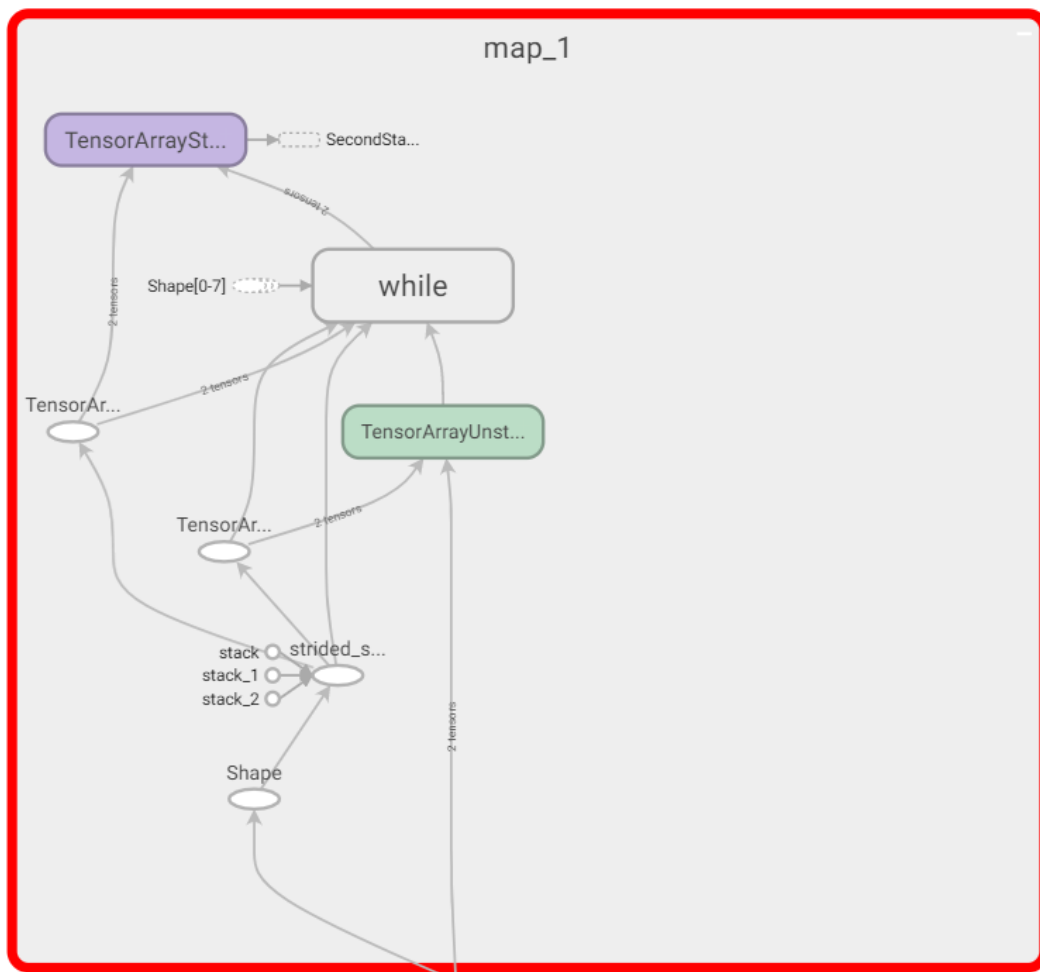
2.10 ScopeToAbsoluteBBoxPass

功能说明

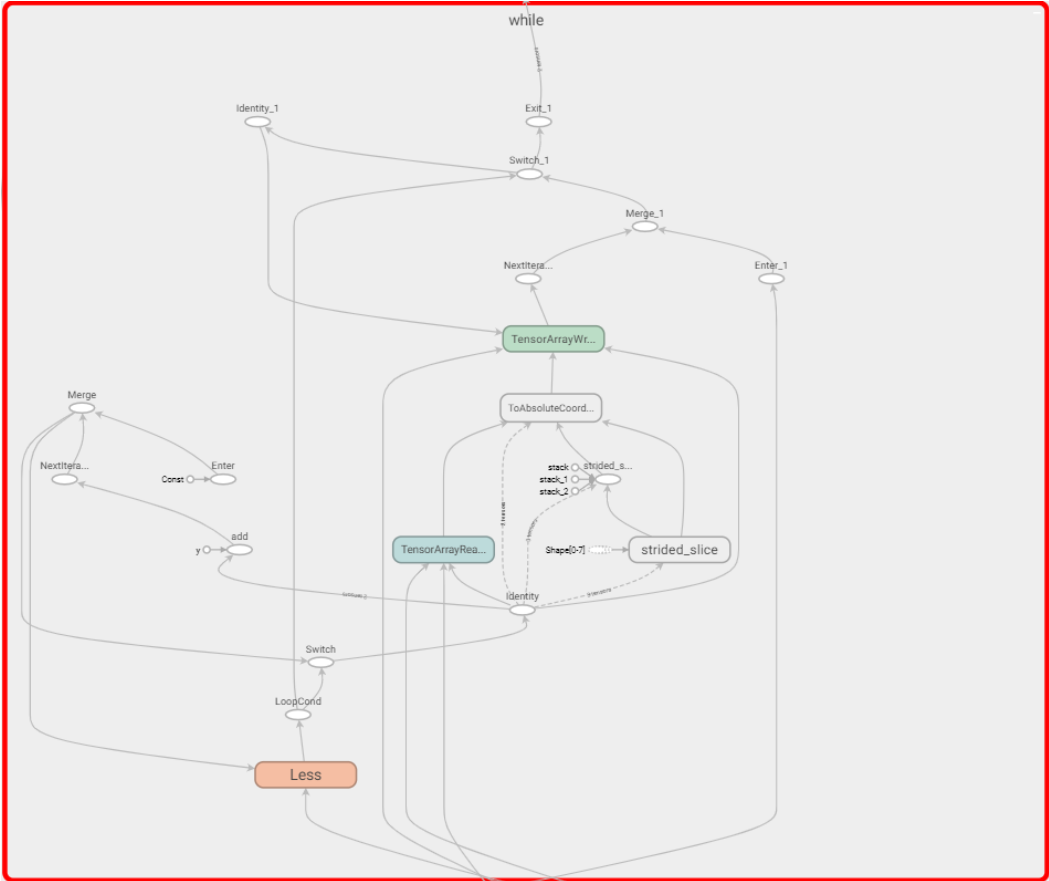
融合的scope为Map, Map下需要有while, while下需要有ToAbsoluteCoordinates, ToAbsoluteCoordinates下有scale, 同时scope下有4个Mul算子, 融合成ToAbsoluteBBox算子。

Scope 详情

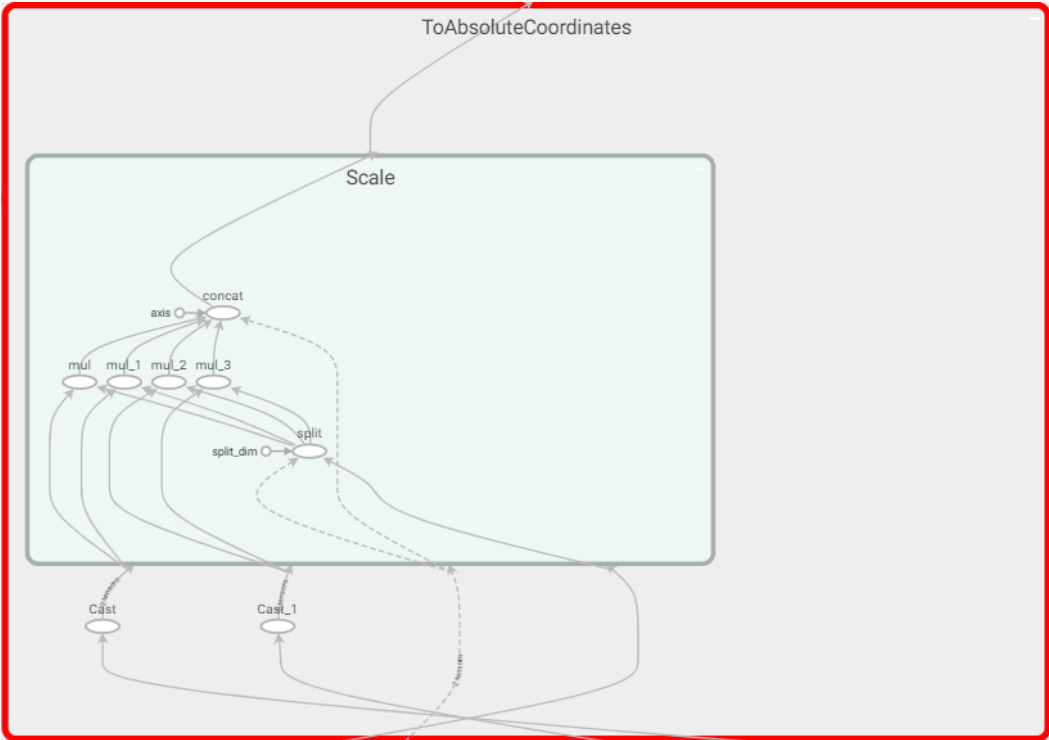
map_1下有while:



while下有ToAbsoluteCoordinates:



ToAbsoluteCoordinates下有scale:



融合后的算子原型

ToAbsoluteBBox，具体请参见《[算子清单](#)》。

融合对应关系

Shape或者TensorArrayUnstack/Shape 输入作为融合后第一个输入

while/strided_slice/Enter的输入作为融合后第二个输入

TensorArrayStack/TensorArrayGatherV3的输出作为融合后第一个输出

融合的scope中存在while/ToAbsoluteCoordinates/scale 子图结构

适用网络

Fast R-CNN

融合规则类型

定制化融合规则

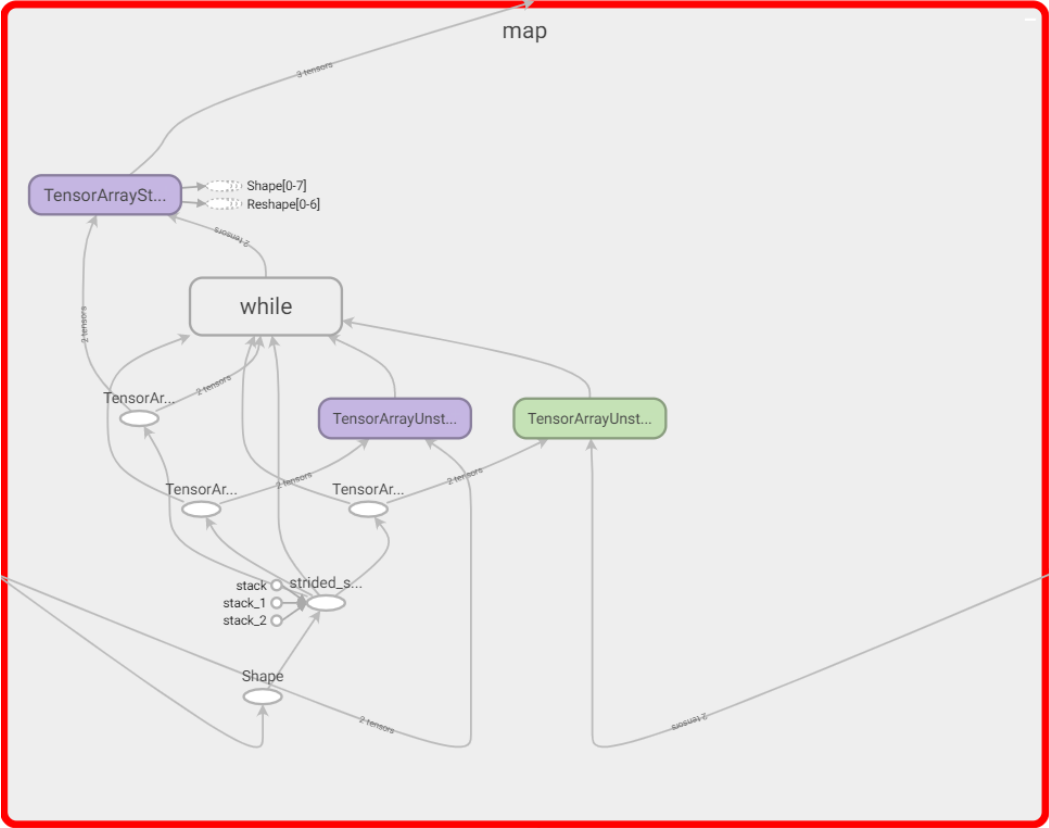
2.11 ScopeNormalizeBBoxPass

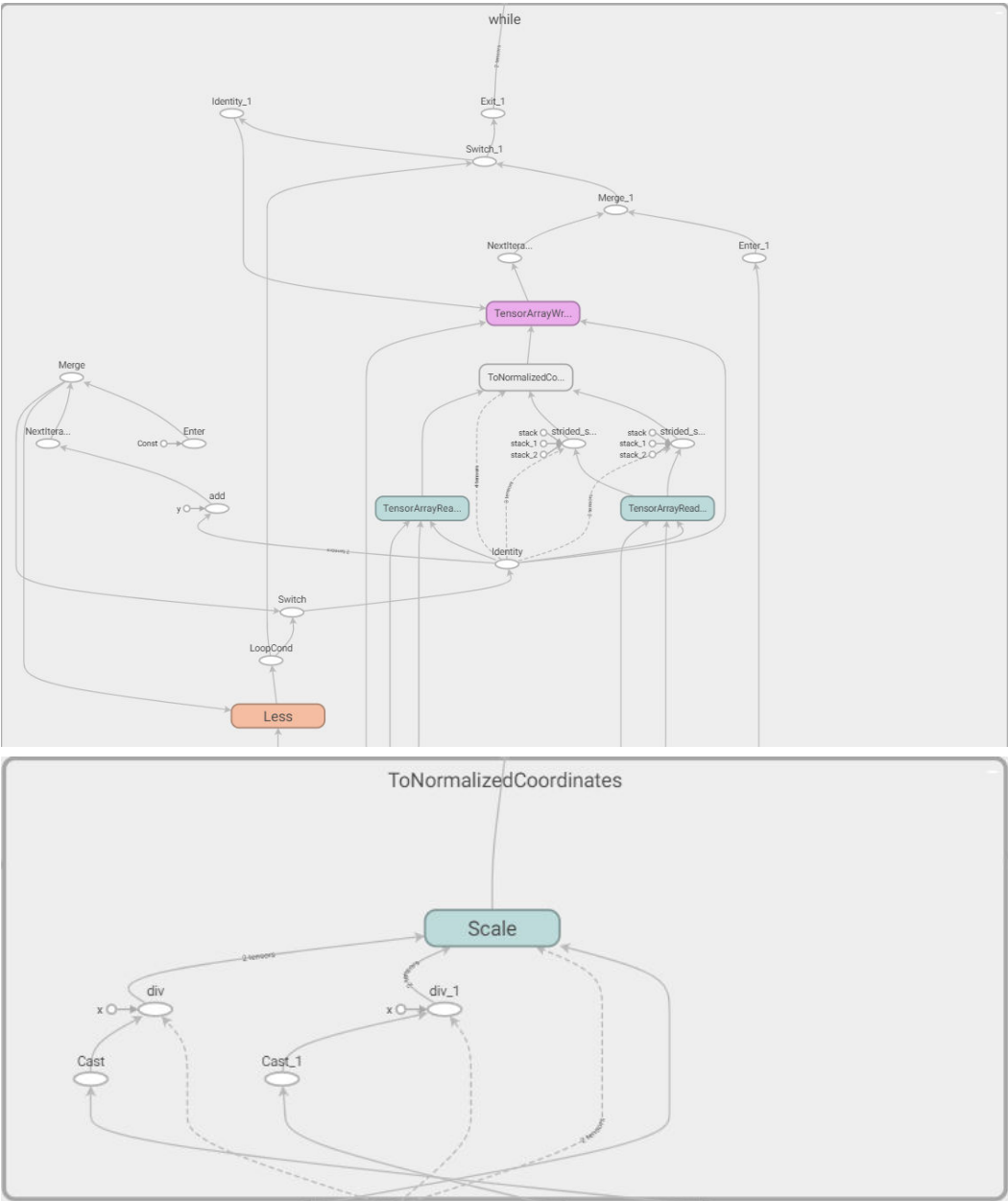
功能说明

融合scope为Map, Map下需要有while,while下需要有ToNormalizedCoordinates,ToNormalizedCoordinates下有scale,同时scope下有4个Mul算子，融合成NormalizeBBox算子。

Scope 详情

Map下有while:





融合后的算子原型

NormalizeBBox，具体请参见《[算子清单](#)》。

融合对应关系

- map/Shape 输入作为融合后第一个输入。
- map/TensorArrayUnstack_1/Shape 输入作为融合后第二个输入。
- TensorArrayStack/TensorArrayGatherV3 输入作为融合后第一个输出。
- scope存在图结构 "while/ToNormalizedCoordinates/Scale/"。

适用网络

Fast R-CNN

融合规则类型

定制化融合规则

2.12 ScopeDecodeBboxV2Pass

功能说明

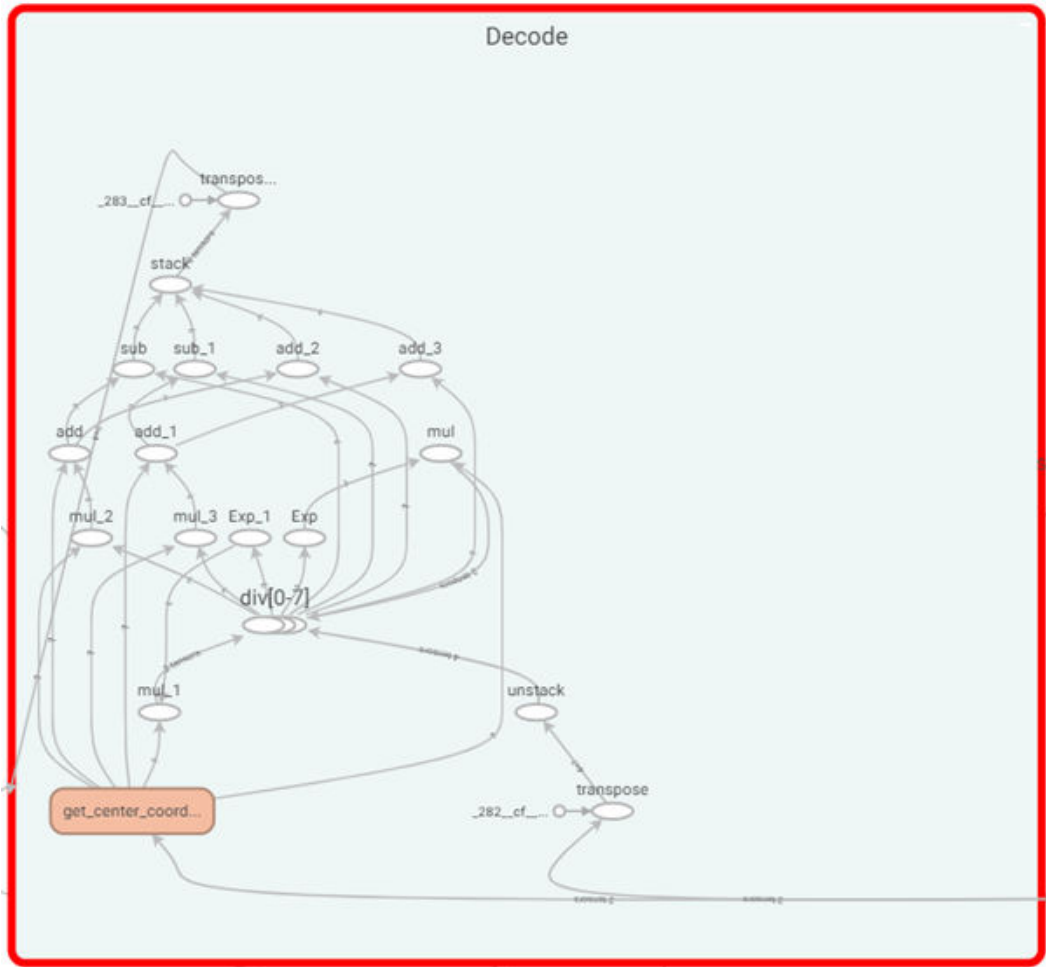
将含有如下两种Scope融合为DecodeBboxV2算子。

Scope1至少包括2个Exp, 4个Mul, 4个Sub, 2的倍数个RealDiv, 2个Unpack, 1个Pack, 3个Transpose算子, 不包括Softmax算子。

Scope2至少包括2个Exp, 4个Mul, 10个Sub, 2的倍数个RealDiv, 2个Unpack, 1个Pack, 3个Transpose, 3个Rank, 3个Range算子, 不包括Sigmoid算子。

Scope 详情

scope1:

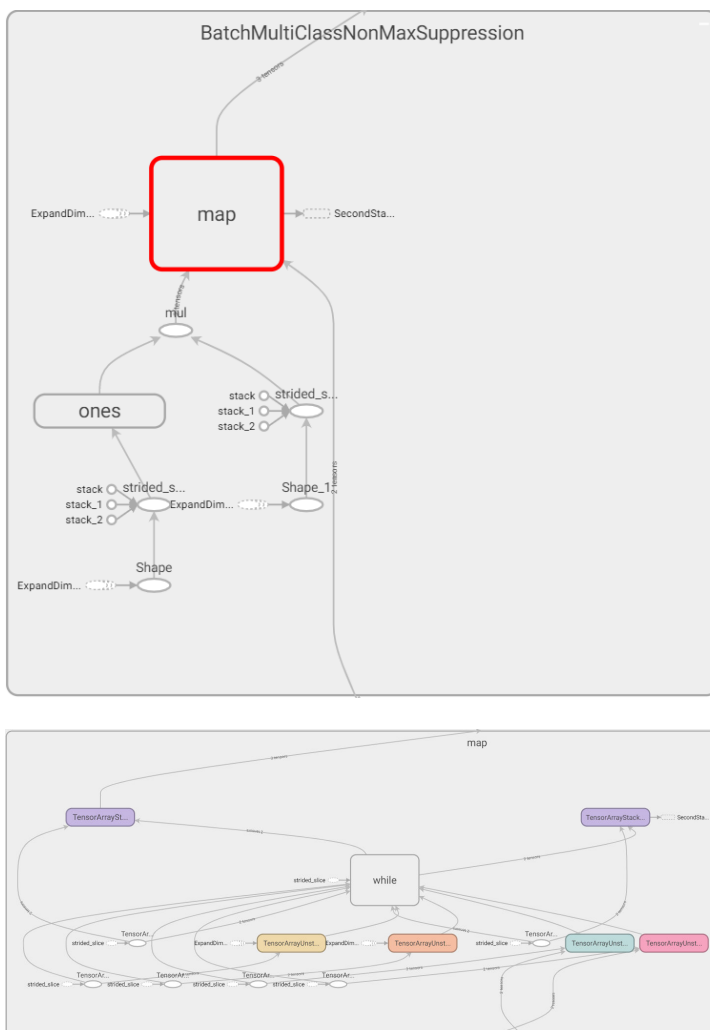


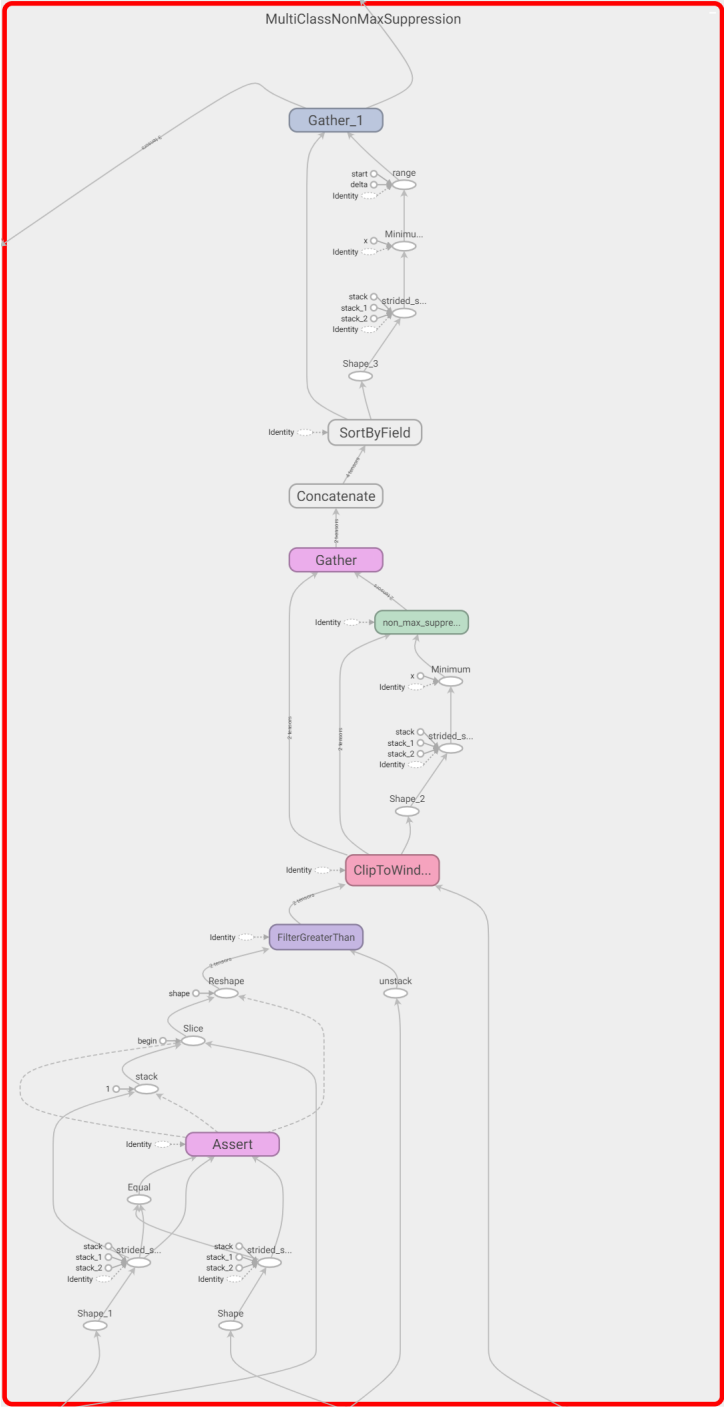
2.13 ScopeBatchMultiClassNMSPass

功能说明

如下图结构进行融合成BatchMultiClassNonMaxSuppression算子，融合scope下有scope路径：map/while/MultiClassNonMaxSuppression/

Scope 详情





融合后的算子原型

BatchMultiClassNonMaxSuppression，具体请参见《[算子清单](#)》。

融合对应关系

map/TensorArrayUnstack/Shape，做为融合后算子第一个输入。

map/TensorArrayUnstack_1/Shape，做为融合后算子第二个输入。

map/TensorArrayUnstack_3/Shape, 如果存在这个节点, 做为融合后算子第三个输入。

map/TensorArrayUnstack_4/Shape, 如果存在这个节点, 做为融合后算子第四个输入。

map/TensorArrayStack/TensorArrayGatherV3, 如果存在, 做为融合后算子第一个输出。

map/TensorArrayStack_1/TensorArrayGatherV3, 如果存在, 做为融合后算子第二个输出。

map/TensorArrayStack_2/TensorArrayGatherV3, 如果存在, 做为融合后算子第三个输出。

map/TensorArrayStack_4/TensorArrayGatherV3, 如果存在, 做为融合后算子第四个输出。

适用网络

Fast R-CNN

SSD-Resnet50V1-FPN

Mask R-cnn

融合规则类型

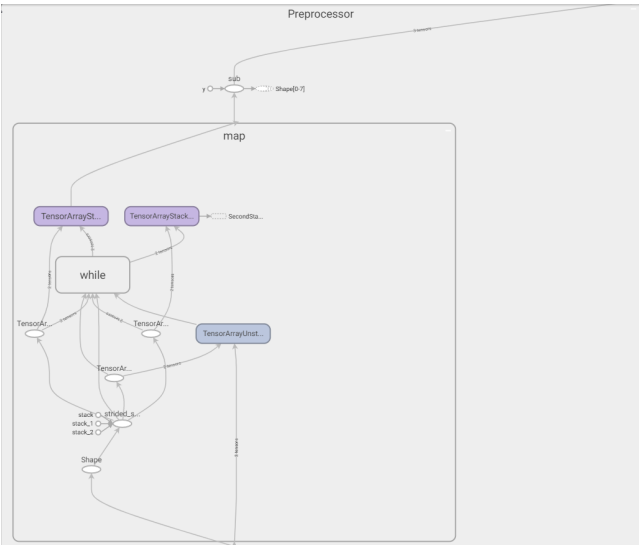
定制化融合规则

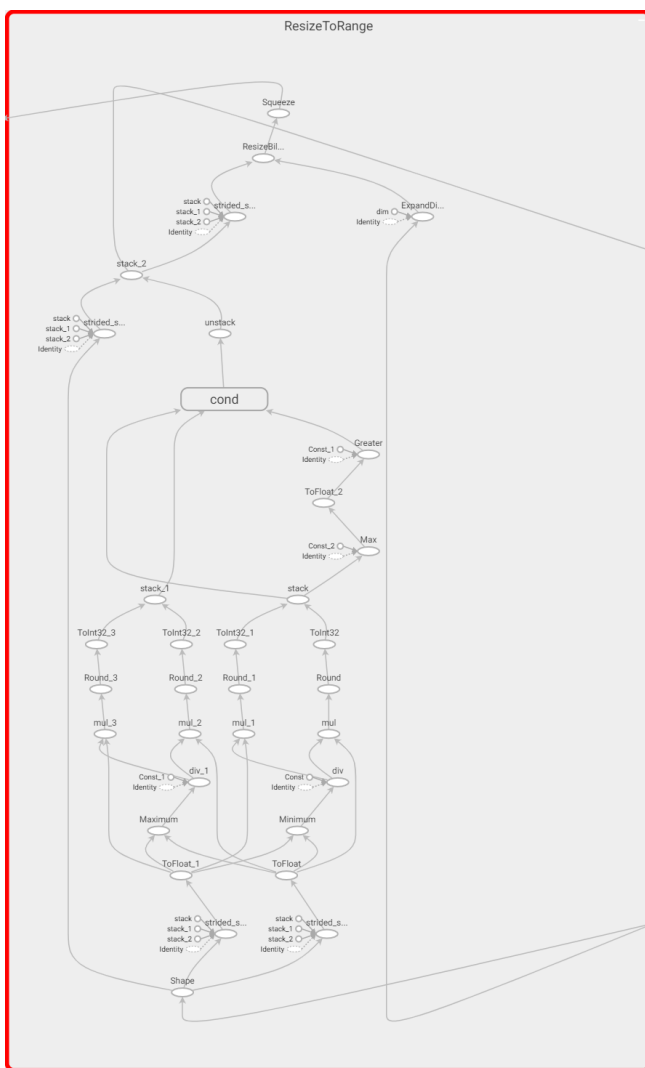
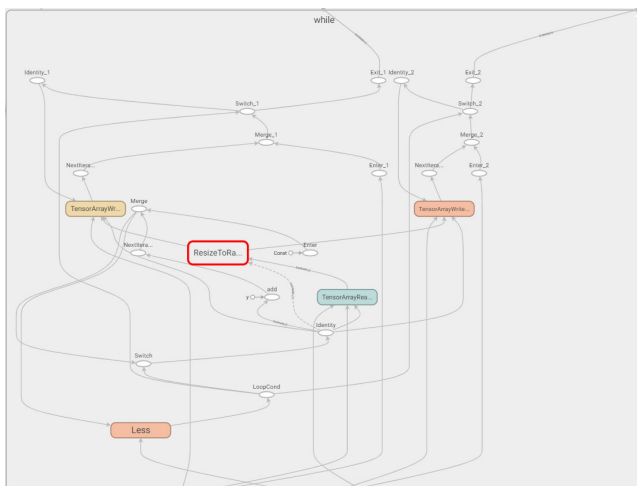
2.14 ScopeKeepRatioResizeBilinearPass

功能说明

将特定Scope进行融合成KeepRationResizeBilinear + Shape + Slice*2 + Expanddims + ConcatV2 + Tile + Const*4, Scope中包括图结构: map/while/ResizeToRange/, 包括算子Maximum, Minimum, Round, ResizeBilinear。

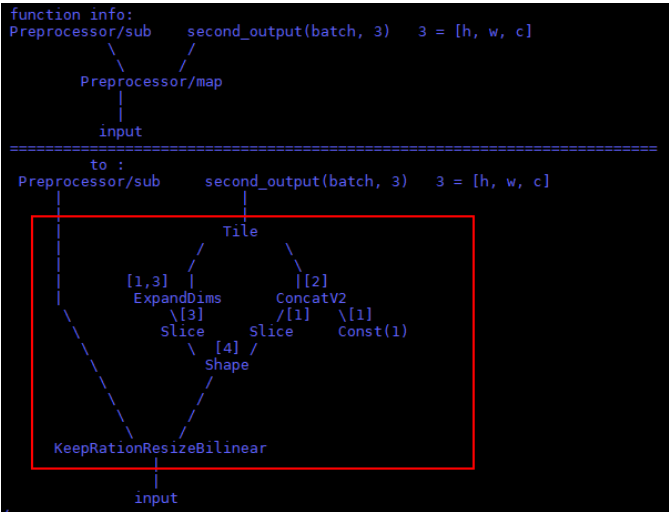
Scope 详情





融合后的算子原型

KeepRationResizeBilinear + Shape + Slice*2 + ExpandDims + ConcatV2 + Tile + Const*4，具体请参见《[算子清单](#)》。



融合对应关系

"scope名字+/Shape" 的节点 输入作为融合后子图的输入。

TensorArrayStack/TensorArrayStack/TensorArrayGatherV3 输出作为融合后子图的第一输出。

TensorArrayStack_1/TensorArrayStack_1/TensorArrayGatherV3 输出作为融合后子图的第二个输出。

融合后子图 KeepRationResizeBilinear算子 的输出连接到scope 第一个输出。

融合后子图 Tile 的输出连接到scope 第二个输出。

适用网络

Fast R-CNN

融合规则类型

定制化融合规则

2.15 ScopeBatchMultiClassNonMaxSuppressionPass

功能说明

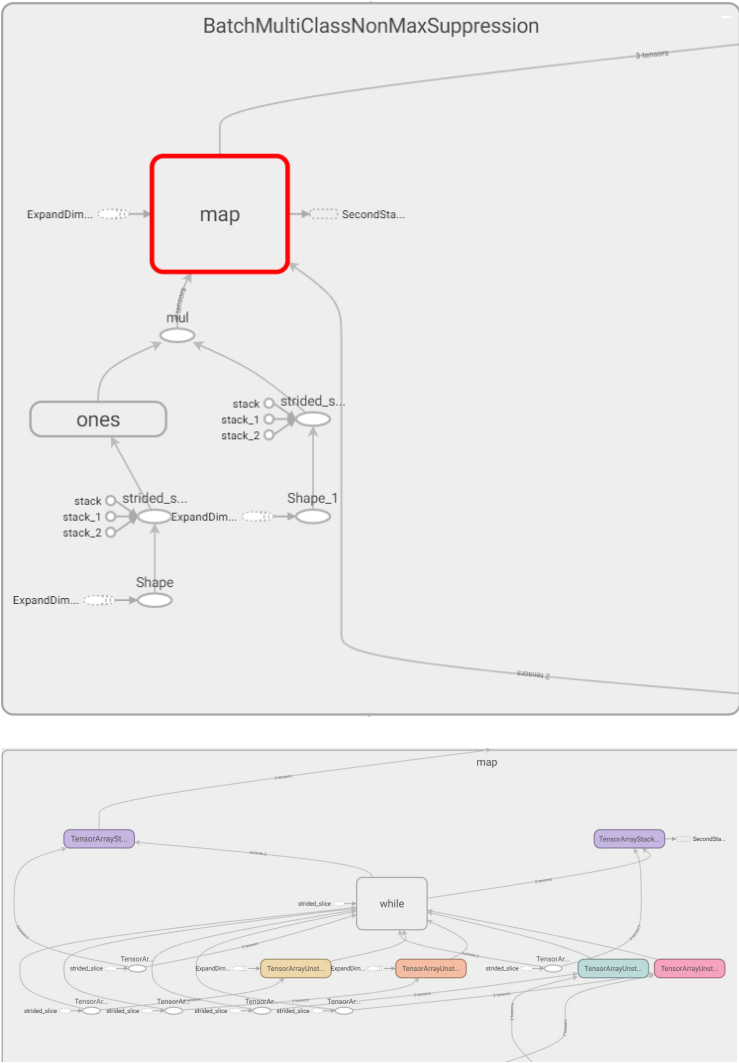
如下图结构进行融合成BatchMultiClassNonMaxSuppression算子，融合scope下有scope路径：

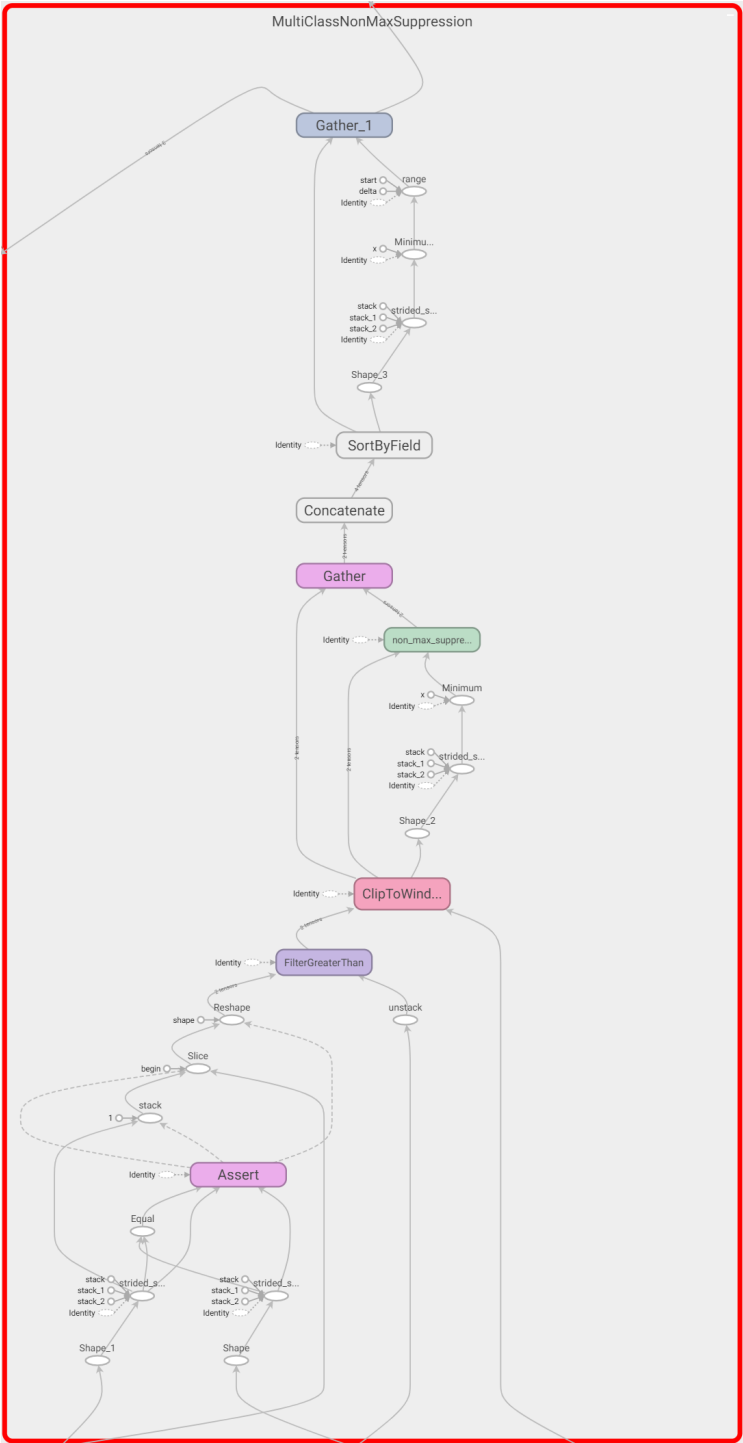
该融合规则包含了两个匹配规则：

ScopeFaceBoxesBatchMultiClassNMSPattern：包括1个 NonMaxSuppressionV3，并且不包含Transpose

ScopeFilteredBatchMultiClassNMSPattern：包括1个NonMaxSuppressionV3，5个Range，1个ConcatV2和80个Fill

Scope 详情





融合后的算子原型

BatchMultiClassNonMaxSuppression，具体请参见《[算子清单](#)》。

融合对应关系

map/TensorArrayUnstack/Shape，做为融合后算子第一个输入。

map/TensorArrayUnstack_1/Shape，做为融合后算子第二个输入。

- map/TensorArrayUnstack_3/Shape, 如果存在这个节点, 做为融合后算子第三个输入。
- map/TensorArrayUnstack_4/Shape, 如果存在这个节点, 做为融合后算子第四个输入。
- map/TensorArrayStack/TensorArrayGatherV3, 如果存在, 做为融合后算子第一个输出。
- map/TensorArrayStack_1/TensorArrayGatherV3, 如果存在, 做为融合后算子第二个输出。
- map/TensorArrayStack_2/TensorArrayGatherV3, 如果存在, 做为融合后算子第三个输出。
- map/TensorArrayStack_4/TensorArrayGatherV3, 如果存在, 做为融合后算子第四个输出。

适用网络

- FaceBox
- Retinanet

融合规则类型

- 定制化融合规则

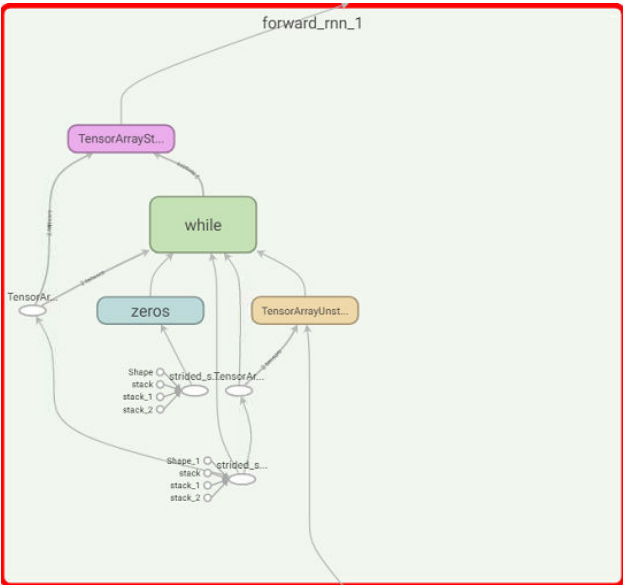
2.16 ScopeDynamicGRUPass

功能说明

将含有如下算子的Scope融合为DynamicGRU算子。Scope内包括：5个AddV2算子、3个Mul算子、1个Tanh算子、不包含Transpose算子。

Scope 详情

例如，待融合的Scope为：



其中while内包含5个AddV2算子、3个Mul算子、1个Tanh算子、不包含Transpose算子，最终把上面红框中的所有算子融合为DynamicGRU算子。

融合后的算子原型

DynamicGRUV2，具体请参见《[算子清单](#)》。

融合对应关系

TensorArrayUnstack/TensorArrayScatter/TensorArrayScatterV3的输入作为融合后算子的第一个输入。

while/ReadVariableOp_1/Enter的输入作为融合后算子的第二个输入。

while/ReadVariableOp_4/Enter的输入作为融合后算子的第三个输入。

while/ReadVariableOp_00/Enter的输入作为融合后算子的第四个输入。

while/ReadVariableOp_01/Enter的输入作为融合后算子的第五个输入。

TensorArrayStack/TensorArrayGatherV3的输出作为融合后算子的第一个输出。

适用网络

DeepSpeech2

融合规则类型

定制化融合规则

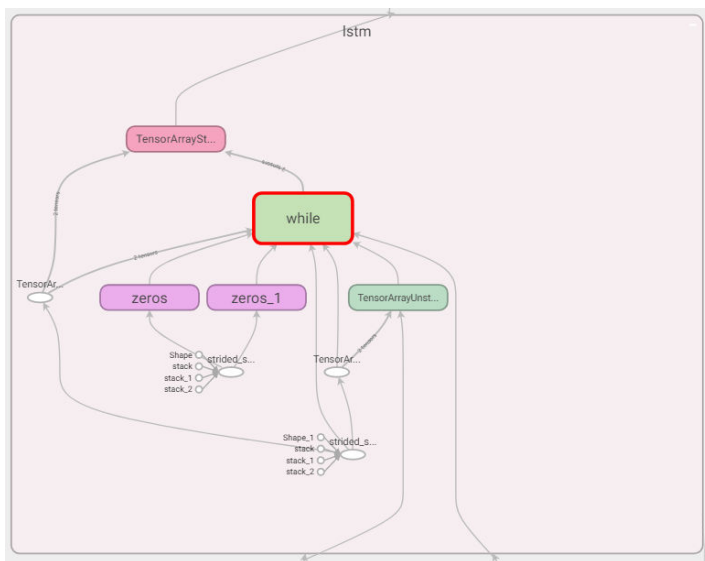
2.17 ScopeDynamicRNPass

功能说明

将含有如下算子的Scope融合为DynamicRNN算子。Scope内包括：1个名为while的子Scope，不包含Transpose算子；while子Scope内包括：4的倍数个BiasAdd算子、2的倍数个Tanh算子、8个MatMul算子，1个Split算子。

Scope 详情

待融合的Scope：



其中while子Scope内包括：4的倍数个BiasAdd算子、2的倍数个Tanh算子、8个MatMul算子，1个Split算子。

融合后的算子原型

DynamicRNN，具体请参见《[算子清单](#)》。

融合对应关系

TensorArrayUnstack/TensorArrayScatter/TensorArrayScatterV3的输入作为融合后算子的第一个输入。

while/split/ReadVariableOp/Enter的输入作为融合后算子的第二个输入。

while/split_1/ReadVariableOp/Enter的输入作为融合后算子的第三个输入。

TensorArrayStack/TensorArrayGatherV3的输出作为融合后算子的第一个输出。

适用网络

TacoTron

融合规则类型

定制化融合规则