AGroup

分享

阅读: **11** 分享: **0**

conv升级相关情况

背景

padding支持

- Padding目前涉及到Padding操作的OP，如conv, pool都只支持对称的Padding方式，不支持两侧Padding size不同。模型转换时无法对齐TensorFlow的SAME模式（在做average pooling时，如果调用pad2d层，会导致pool层无法识别哪些为填充值）

channel last格式支持

- PaddlePaddle目前仅部分OP支持NHWC/NDHWC，大部分不支持，导致整体上NHWC/NDHWC无法使用，所有接口的体验不统一。

升级概况

1.padding功能：

升级前	升级后
只支持对称的Padding方式	1.1支持不对称padding，即两侧Padding size不同; 1.2 支持padding算法：“SAME”和“VALID”;

2. channel last功能

升级前	升级后

升级前	升级后
默认使用NCHW格式的数据	支持NHWC(2d)/NDHWC(3d)格式的输入输出。

参数

1. python参数

以conv2d为例（conv3d同）

```
def conv2d(input,
           num_filters,
           filter_size,
           stride=1,
           padding=0,
           dilation=1,
           groups=None,
           param_attr=None,
           bias_attr=None,
           use_cudnn=True,
           act=None,
           name=None,
           data_format="NCHW")··
```

属性	升级前	升级后

属性	升级前	升级后
padding	只支持对称padding。 padding参数可以为1个整数值或者含2个元素的list或tuple [padding_hight, padding_width]	支持非对称padding，padding参数新增以下三种情况： 1.支持是一个字符串类型，可以是”VALID”或者”SAME”，表示填充算法。 2.包含4个二元组： 当 `data_format` 为”NCHW”时为 [[0,0], [0,0], [padding_height_top, padding_height_bottom], [padding_width_left, padding_width_right]]； 当 `data_format` 为”NHWC”时为[[0,0], [padding_height_top, padding_height_bottom], [padding_width_left, padding_width_right], [0,0]]； 3.包含4个整数值：[padding_height_top, padding_height_bottom, padding_width_left, padding_width_right]；
data_format	没有该参数，默认格式是NCHW	新增该参数，表示输入输出的格式。 conv2d默认NCHW，同时可支持NHWC conv3d默认NCDHW，同时可支持NDHWC

2. c++属性

属性	type	升级前	升级后
`paddings`	`vector<int>`	只支持传2个值	可以传2或4个值
`padding_algorithm`	`string`	无该属性	表示padding算法 可以为”EXPLICIT”,”“SAME”,”VALID” 若为”EXPLICIT”，将使用 `paddings` 的值
`data_format`	`string`	默认值是“AnyLayout”	默认值是“NCHW”，也支持“NHWC”

注意：

在对”data_format”进行判断的时候，用字符串”NHWC” / “NDHWC”做判断：

因为旧版本中的”AnyLayout”和新版本中的”NCHW”/”NCDHW”均指 旧的数据格式。

eg:

```
// 判断是否是channel_last的情况
// ！！不正确的用法
data_format != "NCHW" // data_format还可能是"AnyLayout",这是旧版本中遗留的，旧版本的"AnyLayout"与新版本的"NCHW"(2d) "NCDHW"(3d)说的是一回事。

// 建议用法：
data_format == "NHWC" // 若是channel_last格式 则为true
data_format != "NHWC" // 若是channel_last格式 则为false
```

实现细节c++

1. 输出维度推断 `ConvOp::InferShape`

- padding和data_format对输出维度推断有直接影响。需根据padding和data_format及其他参数计算输出维度。
- 函数 `UpdatePaddingAndDilation` 计算属性paddings和dilations的最终值。
- 用 `bool channel_last` 表示是否是channel_last格式，对不同格式进行不同的处理。

2. kernel 8个

本次升级共涉及8个kernel：

- 可分为CPU,CUDA, CUDNN 3类，以及 前向、反向grad、double grad 3类
- conv2d conv3d 共用的kernel

- -	正向 kernel	Grad Kernel	Double Grad Kernel
CPU	GemmConvKernel	GemmConvGradKernel	GemmConvDoubleGradKernel
CUDA	GemmConvKernel	GemmConvGradKernel	无
cudnn	CUDNNConvOpKernel	CUDNNConvGradOpKernel	CUDNNConvDoubleGradOpKernel

- depthwise_conv2d （没有depthwise_conv3d）

- -	正向 kernel	Grad Kernel
CPU	GemmConvKernel	GemmConvGradKernel
CUDA	DepthwiseConvKernel	DepthwiseConvGradKernel

2.1 GemmConvKernel 细节

- 根据输入参数更新计算padding和dilation的值；
- // todo

2.2 GemmConvGradKernel 和 double grad

2.3 cudnn kernel

注意:

相关代码或文档

PR：

- pool2d/pool3d： https://github.com/PaddlePaddle/Paddle/pull/19739
- conv2d/conv3d: https://github.com/PaddlePaddle/Paddle/pull/20042
- conv_cudnn_op.cu文件diff可查看 https://github.com/liym27/Paddle/commit/acaf2c16fb2fe8559a1d723d727fbd109c033765