tf.nn.depthwise_conv2d_native

```
depthwise_conv2d_native(
    input,
    filter,
    strides,
    padding,
    data_format='NHWC',
    name=None
)
```

Defined in `tensorflow/python/ops/gen_nn_ops.py`.

See the guide: Neural Network > Convolution

Computes a 2-D depthwise convolution given 4-D `input` and `filter` tensors.

Given an input tensor of shape `[batch, in_height, in_width, in_channels]` and a filter / kernel tensor of shape `[filter_height, filter_width, in_channels, channel_multiplier]`, containing `in_channels` convolutional filters of depth 1, `depthwise_conv2d` applies a different filter to each input channel (expanding from 1 channel to `channel_multiplier` channels for each), then concatenates the results together. Thus, the output has `in_channels * channel_multiplier` channels.

```
for k in 0..in_channels-1
  for q in 0..channel_multiplier-1
    output[b, i, j, k * channel_multiplier + q] =
      sum_{di, dj} input[b, strides[1] * i + di, strides[2] * j + dj, k] *
                        filter[di, dj, k, q]
```

Must have `strides[0] = strides[3] = 1`. For the most common case of the same horizontal and vertices strides, `strides = [1, stride, stride, 1]`.

Args:

- `input`: A `Tensor`. Must be one of the following types: `float32`, `float64`.
- `filter`: A `Tensor`. Must have the same type as `input`.
- `strides`: A list of `ints`. 1-D of length 4. The stride of the sliding window for each dimension of `input`.
- `padding`: A `string` from: `"SAME", "VALID"`. The type of padding algorithm to use.
- `data_format`: An optional `string` from: `"NHWC", "NCHW"`. Defaults to `"NHWC"`. Specify the data format of the input and output data. With the default format "NHWC", the data is stored in the order of: [batch, height, width, channels]. Alternatively, the format could be "NCHW", the data storage order of: [batch, channels, height, width].
- `name`: A name for the operation (optional).

Returns:

A `Tensor`. Has the same type as `input`.

Last updated November 2, 2017.