tf.nn.depthwise_conv2d_native_backprop_input

```
depthwise_conv2d_native_backprop_input(
    input_sizes,
    filter,
    out_backprop,
    strides,
    padding,
    data_format='NHWC',
    name=None
)
```

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

See the guide: Neural Network > Convolution

Computes the gradients of depthwise convolution with respect to the input.

Args:

- `input_sizes`: A `Tensor` of type `int32`. An integer vector representing the shape of `input`, based on `data_format`. For example, if `data_format` is 'NHWC' then `input` is a 4-D `[batch, height, width, channels]` tensor.
- `filter`: A `Tensor`. Must be one of the following types: `float32`, `float64`. 4-D with shape `[filter_height, filter_width, in_channels, depthwise_multiplier]`.
- `out_backprop`: A `Tensor`. Must have the same type as `filter`. 4-D with shape based on `data_format`. For example, if `data_format` is 'NHWC' then out_backprop shape is `[batch, out_height, out_width, out_channels]`. Gradients w.r.t. the output of the convolution.
- `strides`: A list of `ints`. The stride of the sliding window for each dimension of the input of the convolution.
- `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 `filter`. 4-D with shape according to `data_format`. For example, if `data_format` is 'NHWC', output shape is `[batch, in_height, in_width, in_channels]`. Gradient w.r.t. the input of the convolution.

GitHub

Twitter

Support

Issue Tracker

Release Notes

Stack Overflow

English

Terms | **Privacy**