Solution

Here's how I did it. **NOTE**: there's more than 1 way to get the correct output shape. Your answer might differ from mine.

```
def maxpool(input):
    ksize = [1, 2, 2, 1]
    strides = [1, 2, 2, 1]
    padding = 'VALID'
    return tf.nn.max_pool(input, ksize, strides, padding)
```

I want to transform the input shape `(1, 4, 4, 1)` to `(1, 2, 2, 1)`. I choose `'VALID'` for the padding algorithm. I find it simpler to understand and it achieves the result I'm looking for.

```
out_height = ceil(float(in_height - filter_height + 1) / float(strides[1]))
out_width  = ceil(float(in_width - filter_width + 1) / float(strides[2]))
```

Plugging in the values:

```
out_height = ceil(float(4 - 2 + 1) / float(2)) = ceil(1.5) = 2
out_width  = ceil(float(4 - 2 + 1) / float(2)) = ceil(1.5) = 2
```

The depth doesn't change during a pooling operation so I don't have to worry about that.

NEXT