

# TensorFlow 2 - Using Convolutional Layers in TensorFlow

March 27, 2018

## 0.0.1 Review

- [TensorFlow 2D convolutions](#) i.e. the use of `tf.nn.conv2d()`
- [Variable](#) i.e. `tf.Variable()`
- How to determine the dimensions of the output based on the input size and the filter size:

```
new_height = (input_height - filter_height + 2 * P)/S + 1
new_width = (input_width - filter_width + 2 * P)/S + 1
```

## 0.0.2 Instruction

Set up the strides, padding, filter weight (F\_w), and filter bias(F\_b) such that the output shape is (1,2,2,3). Note that all of these except strides should be TensorFlow variables.

```
In [1]: '''
        Set up the strides, padding and filter weight/bias such that
        the output shape is (1, 2, 2, 3).
        '''

import tensorflow as tf
import numpy as np

# `tf.nn.conv2d` requires the input be 4D (batch_size, height, width, depth)
# (1, 4, 4, 1)
x = np.array([
    [0, 1, 0.5, 10],
    [2, 2.5, 1, -8],
    [4, 0, 5, 6],
    [15, 1, 2, 3]], dtype = np.float32).reshape((1,4,4,1))
# print(x.shape) # (1, 4, 4, 1)
X = tf.constant(x)

def conv2d(input):
    # Filter (weights and bias)
    # The shape of the filter weight is (height, width, input_depth, output_depth)
    # The shape of the filter bias is (output_dpeth,)
```

```

# TODO: Define the filter weights `F_W` and filter bias `F_b`.
# NOTE: Remember to wrap them in `tf.Variable`, they are trainable parameters after
# The shape of input is (1, 4, 4, 1) --> input_depth = 1
# The output shape is (1, 2, 2, 3) --> output_depth = 3
# The output image size is 2x2 which is changed from 4x4 --> filter size = 3x3
# [filter_size_height, filter_size_width, color_channels, k_output]
F_W = tf.Variable(tf.truncated_normal([3, 3, 1, 3])) # initialise weights to random
F_b = tf.Variable(tf.zeros(3)) # initialise bias to zeros

# TODO: Set the stride for each dimension (batch_size, height, width, depth)
strides = [1,1,1,1]
# TODO: Set the padding, either 'VALID' or 'SAME'
# Because the shape of the output (2x2) differs from that of the input (4x4),
# the padding is set to 'VALID'
padding = 'VALID'
# https://www.tensorflow.org/versions/r0.11/api_docs/python/nn.html#conv2d
# `tf.nn.conv2d` does not include the bias computation so we have to add it ourselves
conv = tf.nn.conv2d(input, F_W, strides, padding)
return tf.nn.bias_add(conv, F_b)

```

```

out = conv2d(X)
print(out.shape)

```

```

/home/supanee/tensorflow/lib/python3.5/site-packages/h5py/__init__.py:36: FutureWarning: Conver
from ._conv import register_converters as _register_converters

```

```

(1, 2, 2, 3)

```