In [1]:
```
from tensorflow import keras
from tensorflow.keras import layers
from keras import models
from keras.models import Sequential
# from sklearn.metrics import confusion_matrix, cohen_kappa_score
import numpy as np
```

In [2]:
```
w = list(range(2))
w[0] = np.vstack((np.ones((1,2,1,1)),np.zeros((1,2,1,1)))) # perhaps this matches 
w[1] = np.zeros(1,) # bias. not (1,1,1) it's just as though squeezed, 1 each channe
```

In [3]:
```
inputs = keras.Input(shape=(10,10,1))
p1 = layers.ZeroPadding2D(padding=((1, 0),(1, 0)),input_shape=(10, 10, 1), data_for
c1 = layers.Conv2D(1, kernel_size=(2, 2), weights=w, padding="valid",name="C1_Layer
p2 = layers.ZeroPadding2D(padding=((1, 0),(1, 0)),input_shape=(10, 10, 1), data_for
c2 = layers.Conv2D(1, kernel_size=(2, 2), weights=w, padding="valid",name="C2_Layer

p3 = layers.ZeroPadding2D(padding=((1, 0),(1, 0)),input_shape=(10, 10, 1), data_for
c3 = layers.Conv2D(1, kernel_size=(2, 2), weights=w, padding="valid",name="C3_Layer
p4 = layers.ZeroPadding2D(padding=((1, 0),(1, 0)),input_shape=(10, 10, 1), data_for
c4 = layers.Conv2D(1, kernel_size=(2, 2), weights=w, padding="valid",name="C4_Layer

p5 = layers.ZeroPadding2D(padding=((1, 0),(1, 0)),input_shape=(10, 10, 1), data_for
c5 = layers.Conv2D(1, kernel_size=(2, 2), weights=w, padding="valid",name="C5_Layer

p6 = layers.ZeroPadding2D(padding=((1, 0),(1, 0)),input_shape=(10, 10, 1), data_for
c6 = layers.Conv2D(1, kernel_size=(2, 2), weights=w, padding="valid",name="C6_Layer

p7 = layers.ZeroPadding2D(padding=((1, 0),(1, 0)),input_shape=(10, 10, 1), data_for
c7 = layers.Conv2D(1, kernel_size=(2, 2), weights=w, padding="valid",name="C7_Layer

p8 = layers.ZeroPadding2D(padding=((1, 0),(1, 0)),input_shape=(10, 10, 1), data_for
c8 = layers.Conv2D(1, kernel_size=(2, 2), weights=w, padding="valid",name="C8_Layer

p9 = layers.ZeroPadding2D(padding=((1, 0),(1, 0)),input_shape=(10, 10, 1), data_for
c9 = layers.Conv2D(1, kernel_size=(2, 2), weights=w, padding="valid",name="C9_Layer

p10 = layers.ZeroPadding2D(padding=((1, 0),(1, 0)),input_shape=(10, 10, 1), data_fo
c10 = layers.Conv2D(1, kernel_size=(2, 2), weights=w, padding="valid",name="C10_Lay
```

In [4]:
```
model = keras.Model(inputs=inputs, outputs=c10, name="pascal_conv_DAG_model")
model.summary()
```

```
Model: "pascal_conv_DAG_model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 input_1 (InputLayer)        [(None, 10, 10, 1)]       0         
                                                                 
 P1_Layer (ZeroPadding2D)    (None, 11, 11, 1)         0         
                                                                 
 C1_Layer (Conv2D)           (None, 10, 10, 1)         5         
                                                                 
 P2_Layer (ZeroPadding2D)    (None, 11, 11, 1)         0         
                                                                 
 C2_Layer (Conv2D)           (None, 10, 10, 1)         5         
                                                                 
 P3_Layer (ZeroPadding2D)    (None, 11, 11, 1)         0         
                                                                 
 C3_Layer (Conv2D)           (None, 10, 10, 1)         5         
                                                                 
 P4_Layer (ZeroPadding2D)    (None, 11, 11, 1)         0         
                                                                 
 C4_Layer (Conv2D)           (None, 10, 10, 1)         5         
                                                                 
 P5_Layer (ZeroPadding2D)    (None, 11, 11, 1)         0         
                                                                 
 C5_Layer (Conv2D)           (None, 10, 10, 1)         5         
                                                                 
 P6_Layer (ZeroPadding2D)    (None, 11, 11, 1)         0         
                                                                 
 C6_Layer (Conv2D)           (None, 10, 10, 1)         5         
                                                                 
 P7_Layer (ZeroPadding2D)    (None, 11, 11, 1)         0         
                                                                 
 C7_Layer (Conv2D)           (None, 10, 10, 1)         5         
                                                                 
 P8_Layer (ZeroPadding2D)    (None, 11, 11, 1)         0         
                                                                 
 C8_Layer (Conv2D)           (None, 10, 10, 1)         5         
                                                                 
 P9_Layer (ZeroPadding2D)    (None, 11, 11, 1)         0         
                                                                 
 C9_Layer (Conv2D)           (None, 10, 10, 1)         5         
                                                                 
 P10_Layer (ZeroPadding2D)   (None, 11, 11, 1)         0         
                                                                 
 C10_Layer (Conv2D)          (None, 10, 10, 1)         5         
                                                                 
=================================================================
Total params: 50
Trainable params: 50
Non-trainable params: 0
_________________________________________________________________
```

In [5]:
```
keras.utils.plot_model(model, "my_first_model_with_shape_info.png", show_shapes=Tru
```

```
You must install pydot (`pip install pydot`) and install graphviz (see instruction
s at https://graphviz.gitlab.io/download/) for plot_model/model_to_dot to work.
```

In [6]:
```
# Merge all available features into a single large vector via concatenation
# Mux = layers.Concatenate()([p1, p2, p3, p4, p5, p6, p7, p8, p9, p10])
Mux = layers.Add()([c1, c2, c3, c4, c5, c6, c7, c8, c9, c10])
model = keras.Model(inputs=inputs, outputs=Mux, name="pascal_conv_DAG_model")
model.summary()
```

```
Model: "pascal_conv_DAG_model"
__________________________________________________________________________________
________________
 Layer (type)                  Output Shape         Param #     Connected to
==================================================================================
================
 input_1 (InputLayer)          [(None, 10, 10, 1)]  0           []

 P1_Layer (ZeroPadding2D)      (None, 11, 11, 1)    0           ['input_1[0][0]']

 C1_Layer (Conv2D)             (None, 10, 10, 1)    5           ['P1_Layer[0]
[0]']

 P2_Layer (ZeroPadding2D)      (None, 11, 11, 1)    0           ['C1_Layer[0]
[0]']

 C2_Layer (Conv2D)             (None, 10, 10, 1)    5           ['P2_Layer[0]
[0]']

 P3_Layer (ZeroPadding2D)      (None, 11, 11, 1)    0           ['C2_Layer[0]
[0]']

 C3_Layer (Conv2D)             (None, 10, 10, 1)    5           ['P3_Layer[0]
[0]']

 P4_Layer (ZeroPadding2D)      (None, 11, 11, 1)    0           ['C3_Layer[0]
[0]']

 C4_Layer (Conv2D)             (None, 10, 10, 1)    5           ['P4_Layer[0]
[0]']

 P5_Layer (ZeroPadding2D)      (None, 11, 11, 1)    0           ['C4_Layer[0]
[0]']

 C5_Layer (Conv2D)             (None, 10, 10, 1)    5           ['P5_Layer[0]
[0]']

 P6_Layer (ZeroPadding2D)      (None, 11, 11, 1)    0           ['C5_Layer[0]
[0]']

 C6_Layer (Conv2D)             (None, 10, 10, 1)    5           ['P6_Layer[0]
[0]']

 P7_Layer (ZeroPadding2D)      (None, 11, 11, 1)    0           ['C6_Layer[0]
[0]']

 C7_Layer (Conv2D)             (None, 10, 10, 1)    5           ['P7_Layer[0]
[0]']

 P8_Layer (ZeroPadding2D)      (None, 11, 11, 1)    0           ['C7_Layer[0]
[0]']

 C8_Layer (Conv2D)             (None, 10, 10, 1)    5           ['P8_Layer[0]
[0]']

 P9_Layer (ZeroPadding2D)      (None, 11, 11, 1)    0           ['C8_Layer[0]
[0]']

 C9_Layer (Conv2D)             (None, 10, 10, 1)    5           ['P9_Layer[0]
[0]']

 P10_Layer (ZeroPadding2D)     (None, 11, 11, 1)    0           ['C9_Layer[0]
[0]']
```

```
 C10_Layer (Conv2D)            (None, 10, 10, 1)    5           ['P10_Layer[0]
[0]']

 add (Add)                     (None, 10, 10, 1)    0           ['C1_Layer[0]
[0]',
                                                                 'C2_Layer[0]
[0]',
                                                                 'C3_Layer[0]
[0]',
                                                                 'C4_Layer[0]
[0]',
                                                                 'C5_Layer[0]
[0]',
                                                                 'C6_Layer[0]
[0]',
                                                                 'C7_Layer[0]
[0]',
                                                                 'C8_Layer[0]
[0]',
                                                                 'C9_Layer[0]
[0]',
                                                                 'C10_Layer[0]
[0]']

==================================================================================
================
Total params: 50
Trainable params: 50
Non-trainable params: 0
__________________________________________________________________________________
________________
```

In [7]:
```python
inputData = np.zeros((10,10,1))
inputData[0,0,0] = 1
inputData = inputData[np.newaxis,:,:,:]
p = model.predict(inputData)
```

```
1/1 [==============================] - 0s 169ms/step
```

In [8]:
```python
# print(p) # commented to avoid paper waste for today!
```

In [9]:
```python
np.shape(p)
```

Out[9]:
```
(1, 10, 10, 1)
```

In [10]:
```python
print(np.squeeze(p))
```

```
[[  0.   0.   0.   0.   0.   0.   0.   0.   0.   0.]
 [  1.   1.   0.   0.   0.   0.   0.   0.   0.   0.]
 [  1.   2.   1.   0.   0.   0.   0.   0.   0.   0.]
 [  1.   3.   3.   1.   0.   0.   0.   0.   0.   0.]
 [  1.   4.   6.   4.   1.   0.   0.   0.   0.   0.]
 [  1.   5.  10.  10.   5.   1.   0.   0.   0.   0.]
 [  1.   6.  15.  20.  15.   6.   1.   0.   0.   0.]
 [  1.   7.  21.  35.  35.  21.   7.   1.   0.   0.]
 [  1.   8.  28.  56.  70.  56.  28.   8.   1.   0.]
 [  1.   9.  36.  84. 126. 126.  84.  36.   9.   1.]]
```

In [ ]: