Centro Brasileiro de Pesquisas Físicas

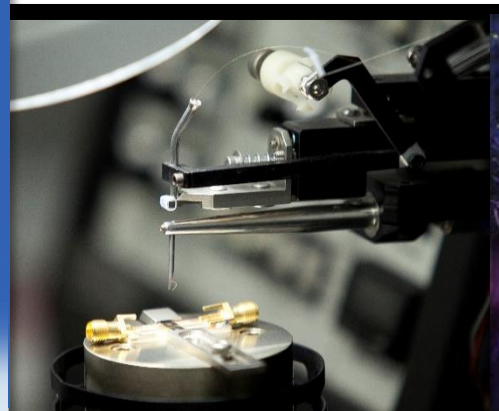
CBPF

Inteligência artificial e aplicações em física

Clécio Roque De Bom – debom@cbpf.br

Elisangela L. Faria
(CBPF)

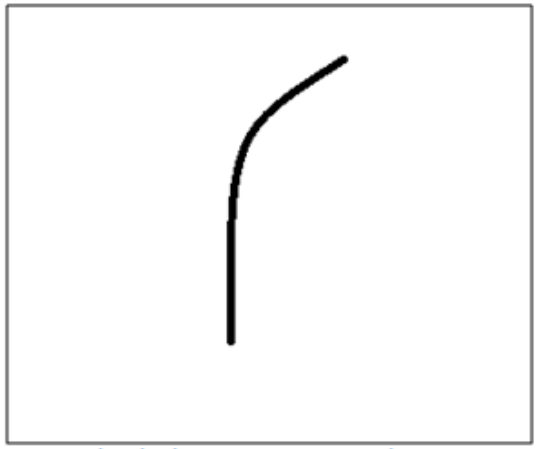
clearnightsrthebest.com



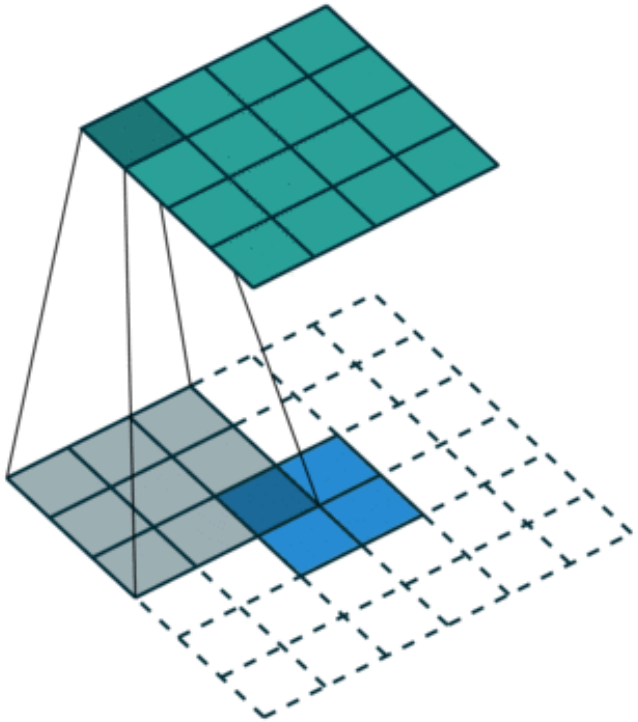
Conv Layers

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter



Visualization of a curve detector filter



https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/

Conv Layers

Visualization of the filter on the image

0	0	0	0	0	0	0
0	40	0	0	0	0	0
40	0	40	0	0	0	0
40	20	0	0	0	0	0
0	50	0	0	0	0	0
0	0	50	0	0	0	0
25	25	0	50	0	0	0

Pixel representation of receptive field

*

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter

Multiplication and Summation = 0

Visualization of the receptive field

0	0	0	0	0	0	30
0	0	0	0	50	50	50
0	0	0	20	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0

Pixel representation of the receptive field

*

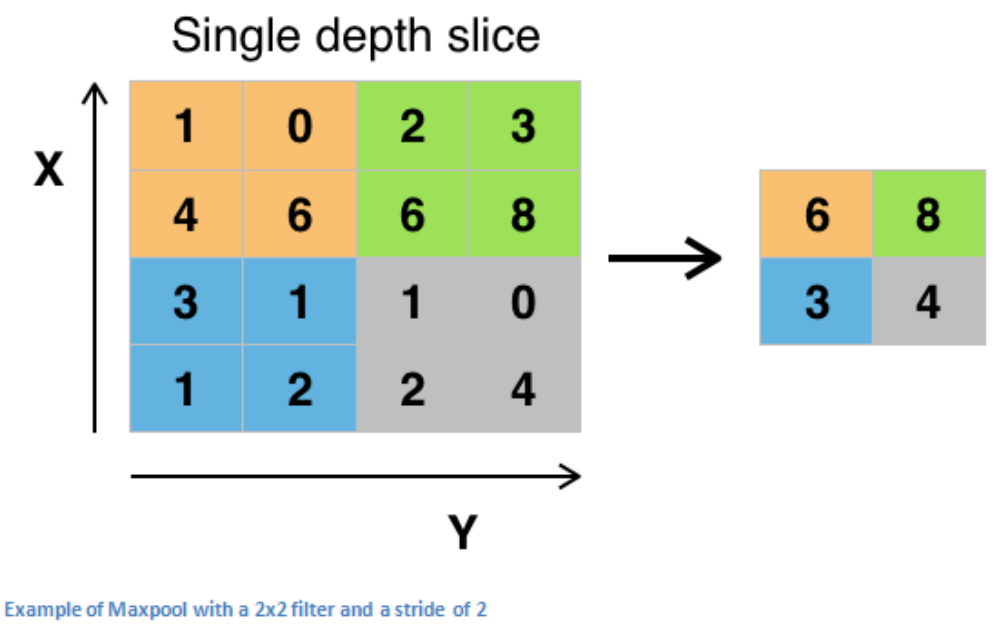
0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter

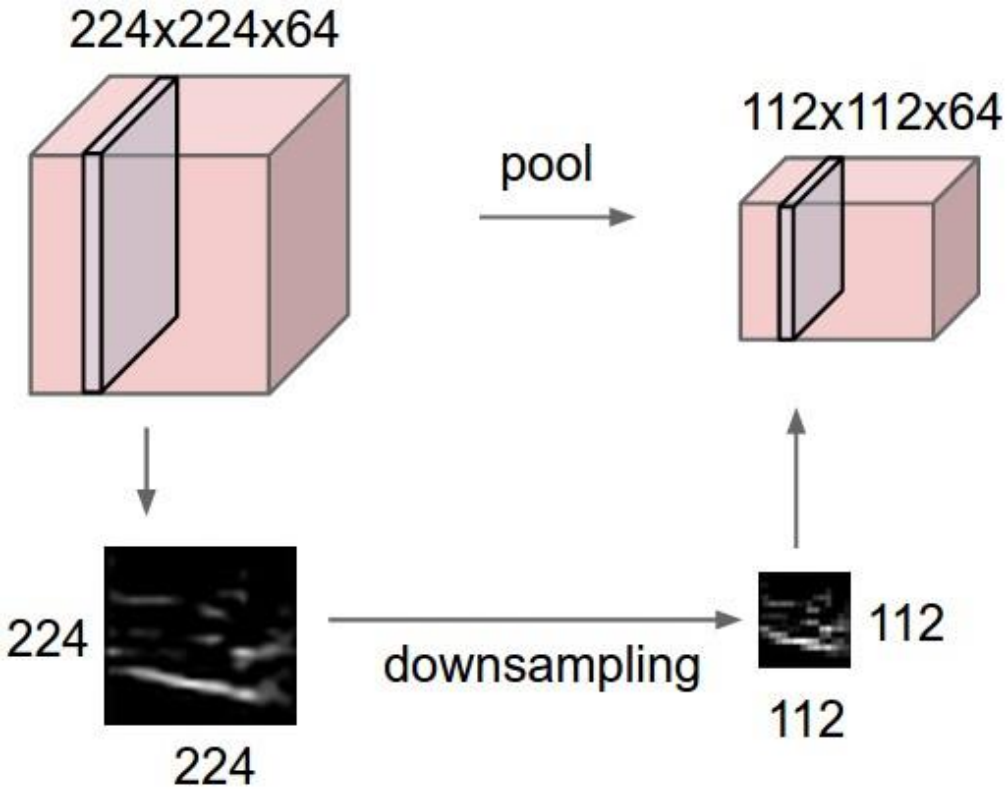
Multiplication and Summation = (50*30)+(50*30)+(50*30)+(20*30)+(50*30) = 6600 (A large number!)

CBPF

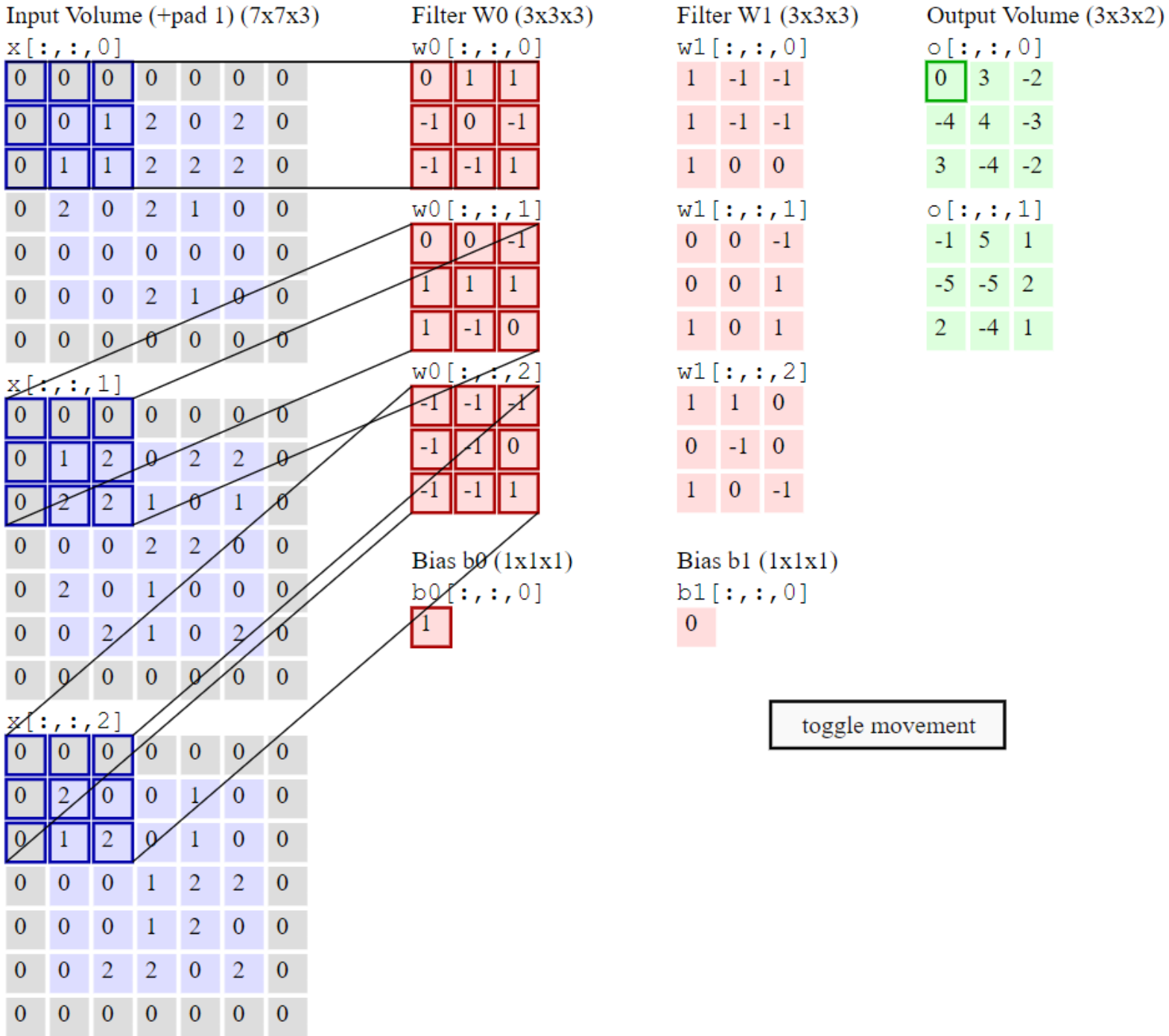
Pooling

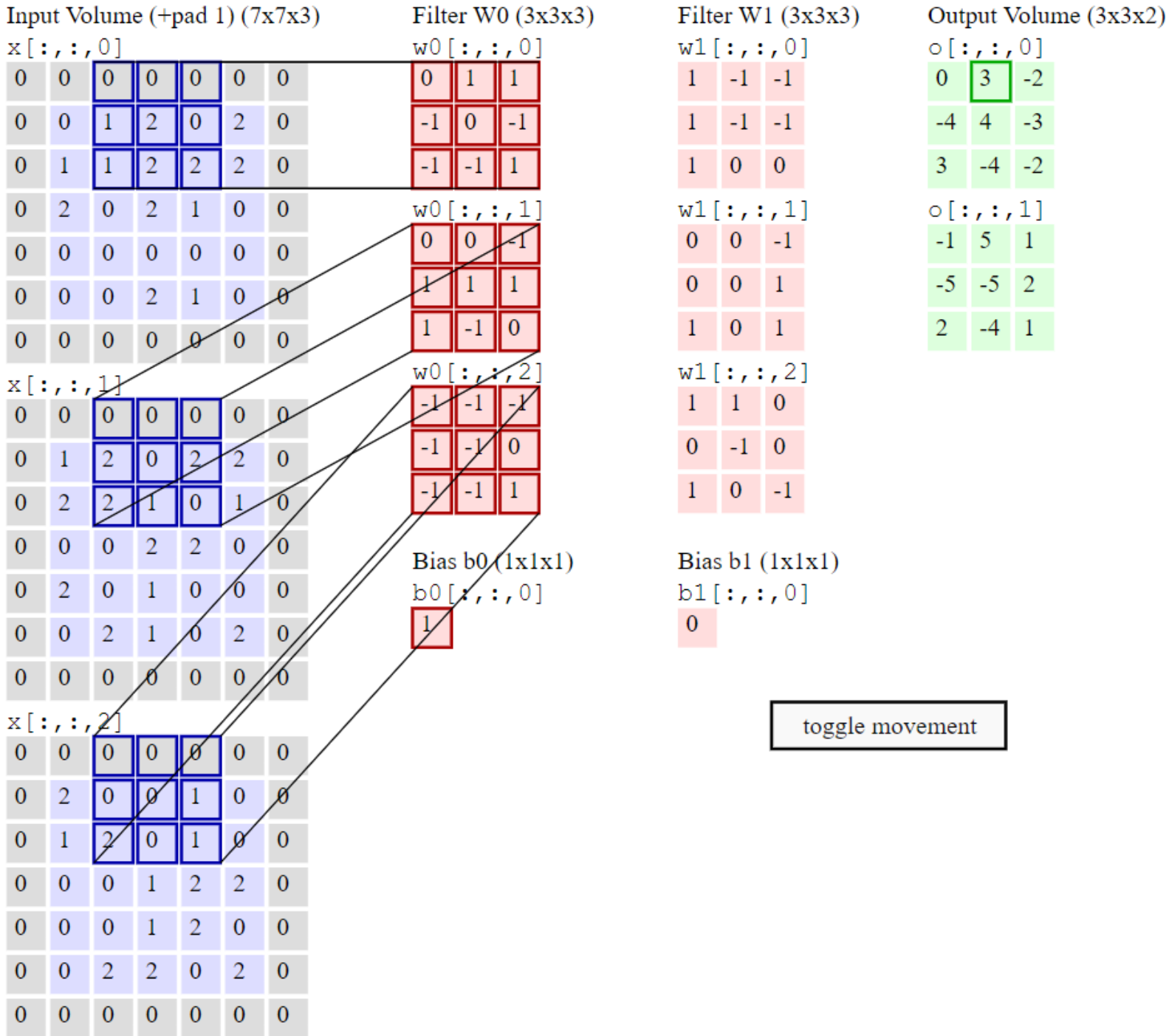


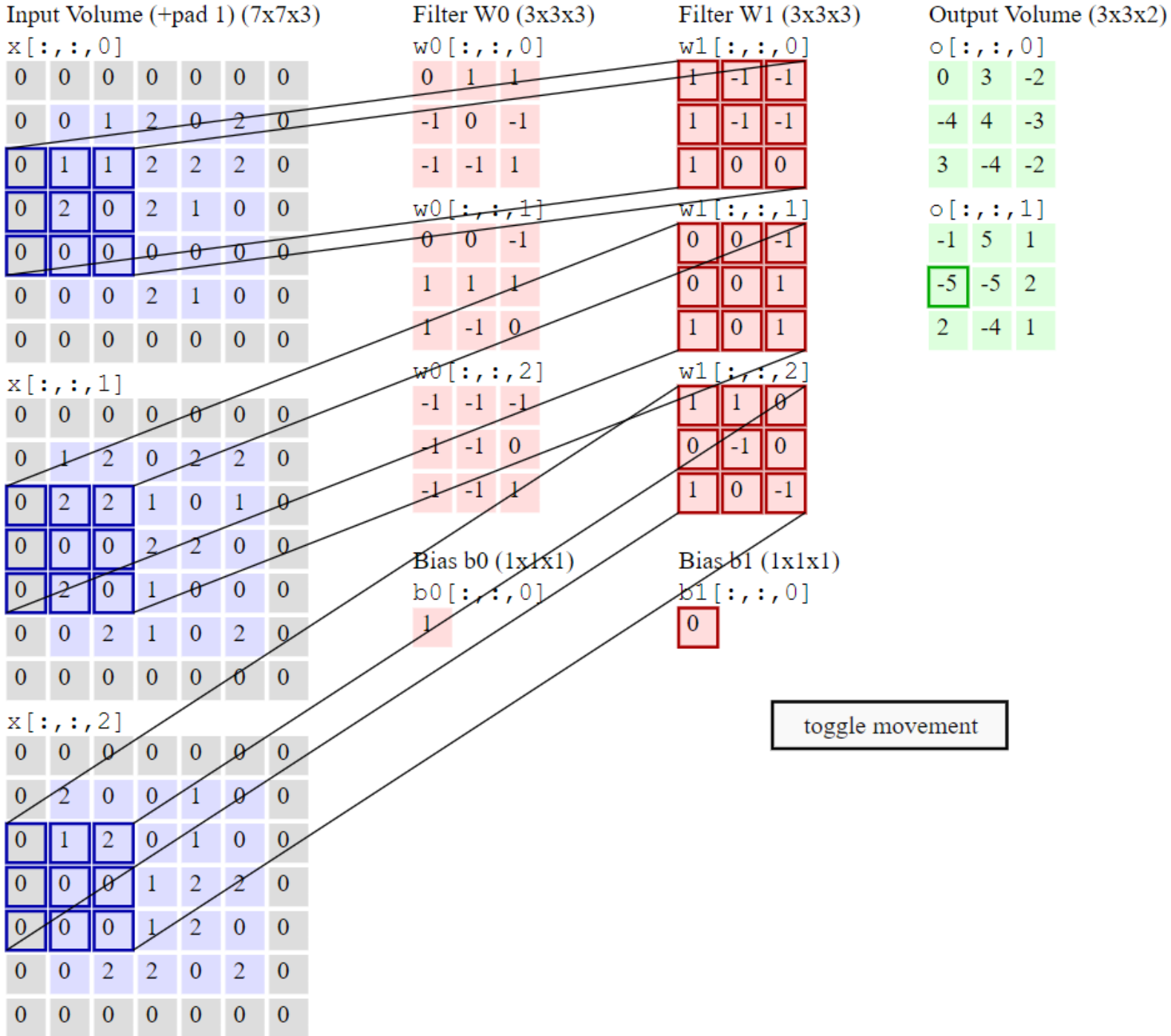
Example of Maxpool with a 2x2 filter and a stride of 2

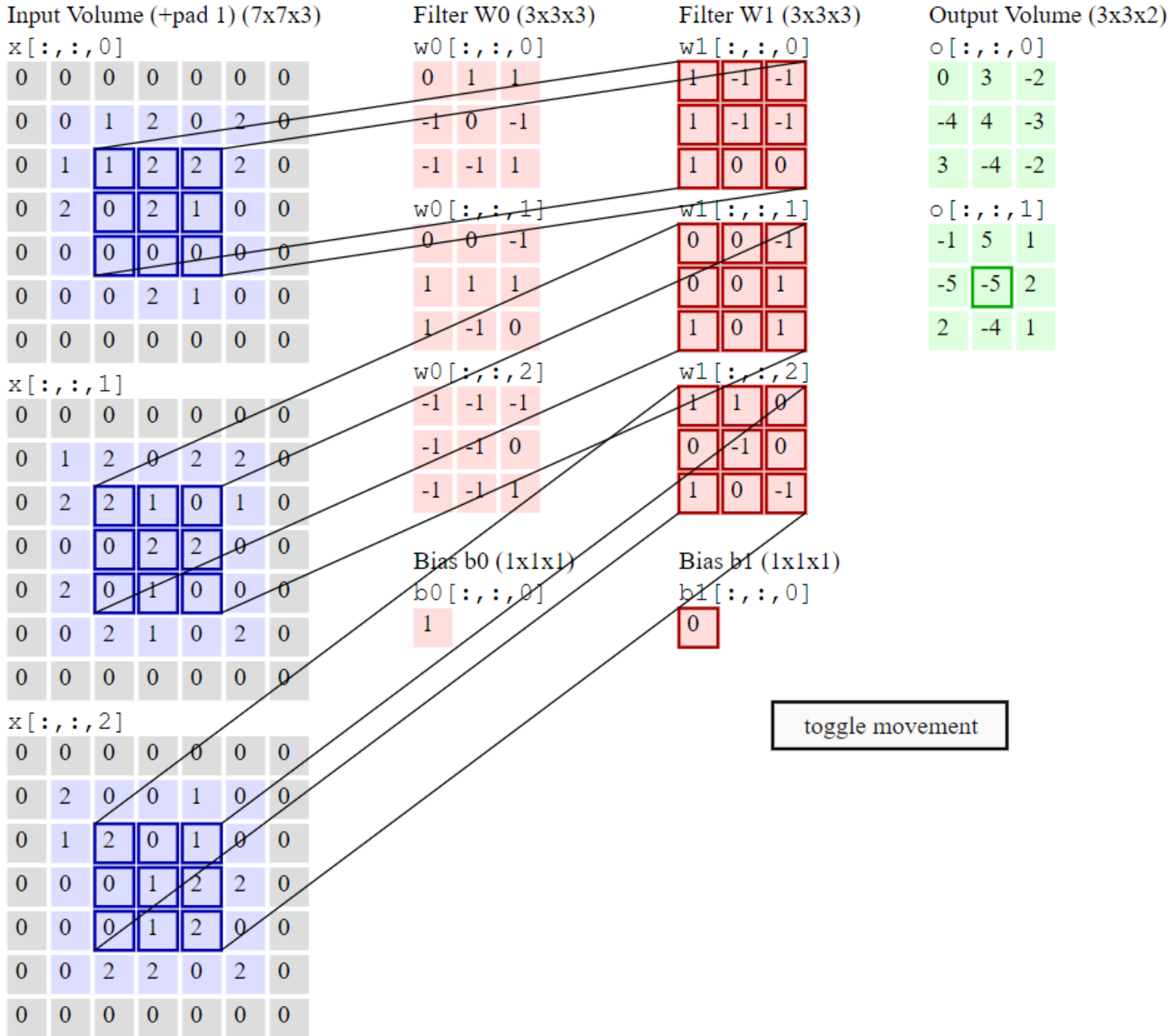


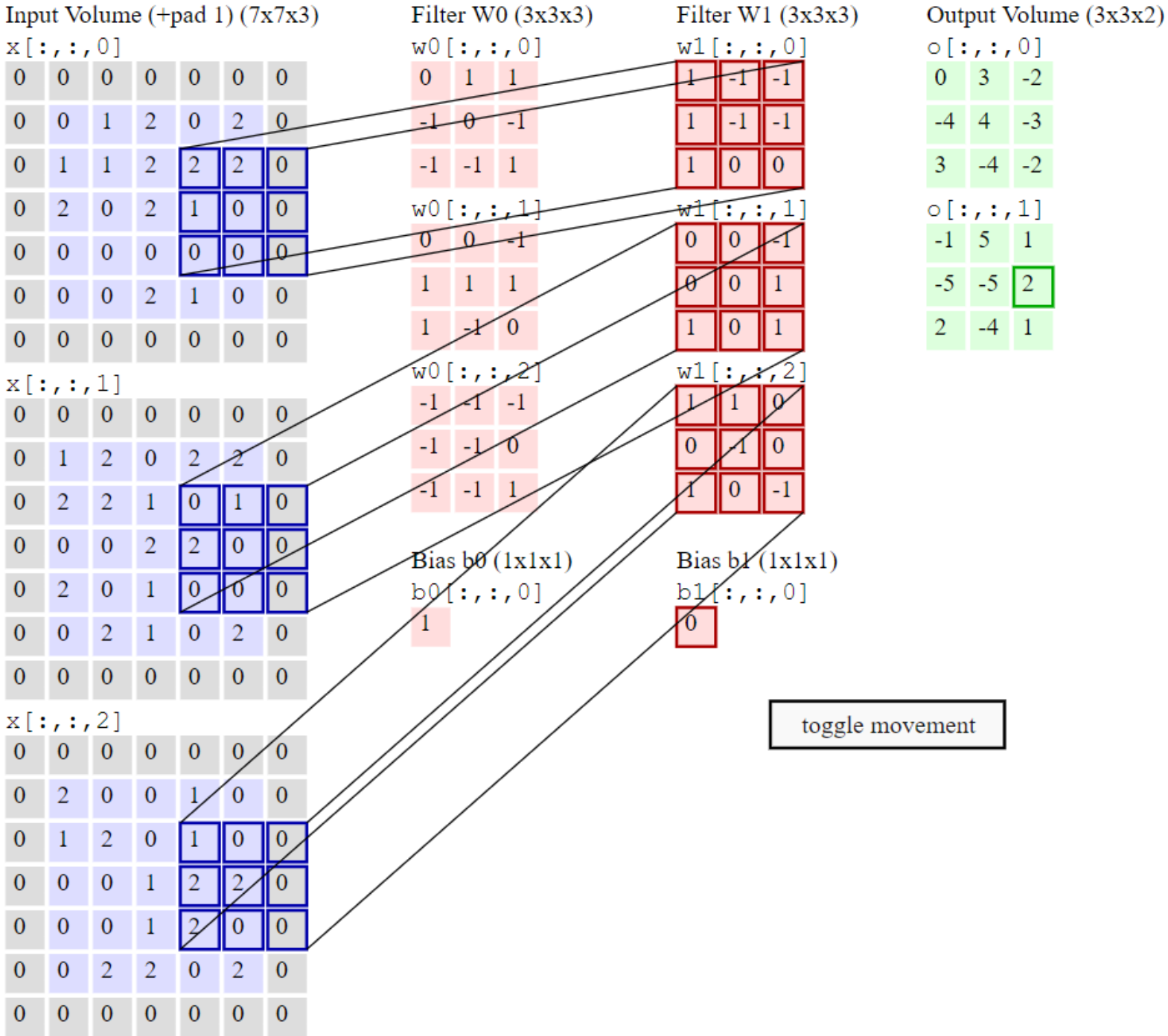
Adapted from : https://ml-cheatsheet.readthedocs.io/en/latest/layers.html


Input Volume (+pad 1) (7x7x3)
x[:,:,0]
0 0 0 0 0 0 0
0 0 1 2 0 2 0
0 1 1 2 2 2 0
0 2 0 2 1 0 0
0 0 0 0 0 0 0
0 0 0 2 1 0 0
0 0 0 0 0 0 0
x[:,:,1]
0 0 0 0 0 0 0
0 1 2 0 2 2 0
0 2 2 1 0 1 0
0 0 0 2 2 0 0
0 2 0 1 0 0 0
0 0 2 1 0 2 0
0 0 0 0 0 0 0
x[:,:,2]
0 0 0 0 0 0 0
0 2 0 0 1 0 0
0 1 2 0 1 0 0
0 0 0 1 2 2 0
0 0 0 1 2 0 0
0 0 2 2 0 2 0
0 0 0 0 0 0 0
Filter W0 (3x3x3)
w0[:,:,0]
0 1 1
-1 0 -1
-1 -1 1
w0[:,:,1]
0 0 -1
1 1 1
1 -1 0
w0[:,:,2]
-1 -1 -1
-1 -1 0
-1 -1 1
Bias b0 (1x1x1)
b0[:,:,0]
1
Filter W1 (3x3x3)
w1[:,:,0]
1 -1 -1
1 -1 -1
1 0 0
w1[:,:,1]
0 0 -1
0 0 1
1 0 1
w1[:,:,2]
1 1 0
0 -1 0
1 0 -1
Bias b1 (1x1x1)
b1[:,:,0]
0
Output Volume (3x3x2)
o[:,:,0]
0 3 -2
-4 4 -3
3 -4 -2
o[:,:,1]
-1 5 1
-5 -5 2
2 -4 1
toggle movement


Input Volume (+pad 1) (7x7x3)
x[:,:,0]
0 0 0 0 0 0 0
0 0 1 2 0 2 0
0 1 1 2 2 2 0
0 2 0 2 1 0 0
0 0 0 0 0 0 0
0 0 0 2 1 0 0
0 0 0 0 0 0 0
x[:,:,1]
0 0 0 0 0 0 0
0 1 2 0 2 2 0
0 2 2 1 0 1 0
0 0 0 2 2 0 0
0 2 0 1 0 0 0
0 0 2 1 0 2 0
0 0 0 0 0 0 0
x[:,:,2]
0 0 0 0 0 0 0
0 2 0 0 1 0 0
0 1 2 0 1 0 0
0 0 0 1 2 2 0
0 0 0 1 2 0 0
0 0 2 2 0 2 0
0 0 0 0 0 0 0
Filter W0 (3x3x3)
w0[:,:,0]
0 1 1
-1 0 -1
-1 -1 1
w0[:,:,1]
0 0 -1
1 1 1
1 -1 0
w0[:,:,2]
-1 -1 -1
-1 -1 0
-1 -1 1
Bias b0 (1x1x1)
b0[:,:,0]
1
Filter W1 (3x3x3)
w1[:,:,0]
1 -1 -1
1 -1 -1
1 0 0
w1[:,:,1]
0 0 -1
0 0 1
1 0 1
w1[:,:,2]
1 1 0
0 -1 0
1 0 -1
Bias b1 (1x1x1)
b1[:,:,0]
0
Output Volume (3x3x2)
o[:,:,0]
0 3 -2
-4 4 -3
3 -4 -2
o[:,:,1]
-1 5 1
-5 -5 2
2 -4 1
toggle movement


Input Volume (+pad 1) (7x7x3)
x[:,:,0]
0 0 0 0 0 0 0
0 0 1 2 0 2 0
0 1 1 2 2 2 0
0 2 0 2 1 0 0
0 0 0 0 0 0 0
0 0 0 2 1 0 0
0 0 0 0 0 0 0
x[:,:,1]
0 0 0 0 0 0 0
0 1 2 0 2 2 0
0 2 2 1 0 1 0
0 0 0 2 2 0 0
0 2 0 1 0 0 0
0 0 2 1 0 2 0
0 0 0 0 0 0 0
x[:,:,2]
0 0 0 0 0 0 0
0 2 0 0 1 0 0
0 1 2 0 1 0 0
0 0 0 1 2 2 0
0 0 0 1 2 0 0
0 0 2 2 0 2 0
0 0 0 0 0 0 0
Filter W0 (3x3x3)
w0[:,:,0]
0 1 1
-1 0 -1
-1 -1 1
w0[:,:,1]
0 0 -1
1 1 1
1 -1 0
w0[:,:,2]
-1 -1 -1
-1 -1 0
-1 -1 1
Filter W1 (3x3x3)
w1[:,:,0]
1 -1 -1
1 -1 -1
1 0 0
w1[:,:,1]
0 0 -1
0 0 1
1 0 1
w1[:,:,2]
1 1 0
0 -1 0
1 0 -1
Output Volume (3x3x2)
o[:,:,0]
0 3 -2
-4 4 -3
3 -4 -2
o[:,:,1]
-1 5 1
-5 -5 2
2 -4 1
Bias b0 (1x1x1)
b0[:,:,0]
1
Bias b1 (1x1x1)
b1[:,:,0]
0
toggle movement


Input Volume (+pad 1) (7x7x3)
x[:,:,0]
0 0 0 0 0 0 0
0 0 1 2 0 2 0
0 1 1 2 2 2 0
0 2 0 2 1 0 0
0 0 0 0 0 0 0
0 0 0 2 1 0 0
0 0 0 0 0 0 0
x[:,:,1]
0 0 0 0 0 0 0
0 1 2 0 2 2 0
0 2 2 1 0 1 0
0 0 0 2 2 0 0
0 2 0 1 0 0 0
0 0 2 1 0 2 0
0 0 0 0 0 0 0
x[:,:,2]
0 0 0 0 0 0 0
0 2 0 0 1 0 0
0 1 2 0 1 0 0
0 0 0 1 2 2 0
0 0 0 1 2 0 0
0 0 2 2 0 2 0
0 0 0 0 0 0 0
Filter W0 (3x3x3)
w0[:,:,0]
0 1 1
-1 0 -1
-1 -1 1
w0[:,:,1]
0 0 -1
1 1 1
1 -1 0
w0[:,:,2]
-1 -1 -1
-1 -1 0
-1 -1 1
Filter W1 (3x3x3)
w1[:,:,0]
1 -1 -1
1 -1 -1
1 0 0
w1[:,:,1]
0 0 -1
0 0 1
1 0 1
w1[:,:,2]
1 1 0
0 -1 0
1 0 -1
Output Volume (3x3x2)
o[:,:,0]
0 3 -2
-4 4 -3
3 -4 -2
o[:,:,1]
-1 5 1
-5 -5 2
2 -4 1
Bias b0 (1x1x1)
b0[:,:,0]
1
Bias b1 (1x1x1)
b1[:,:,0]
0
toggle movement


Input Volume (+pad 1) (7x7x3)
x[:,:,0]
0 0 0 0 0 0 0
0 0 1 2 0 2 0
0 1 1 2 2 2 0
0 2 0 2 1 0 0
0 0 0 0 0 0 0
0 0 0 2 1 0 0
0 0 0 0 0 0 0
x[:,:,1]
0 0 0 0 0 0 0
0 1 2 0 2 2 0
0 2 2 1 0 1 0
0 0 0 2 2 0 0
0 2 0 1 0 0 0
0 0 2 1 0 2 0
0 0 0 0 0 0 0
x[:,:,2]
0 0 0 0 0 0 0
0 2 0 0 1 0 0
0 1 2 0 1 0 0
0 0 0 1 2 2 0
0 0 0 1 2 0 0
0 0 2 2 0 2 0
0 0 0 0 0 0 0
Filter W0 (3x3x3)
w0[:,:,0]
0 1 1
-1 0 -1
-1 -1 1
w0[:,:,1]
0 0 -1
1 1 1
1 -1 0
w0[:,:,2]
-1 -1 -1
-1 -1 0
-1 -1 1
Bias b0 (1x1x1)
b0[:,:,0]
1
Filter W1 (3x3x3)
w1[:,:,0]
1 -1 -1
1 -1 -1
1 0 0
w1[:,:,1]
0 0 -1
0 0 1
1 0 1
w1[:,:,2]
1 1 0
0 -1 0
1 0 -1
Bias b1 (1x1x1)
b1[:,:,0]
0
Output Volume (3x3x2)
o[:,:,0]
0 3 -2
-4 4 -3
3 -4 -2
o[:,:,1]
-1 5 1
-5 -5 2
2 -4 1
toggle movement

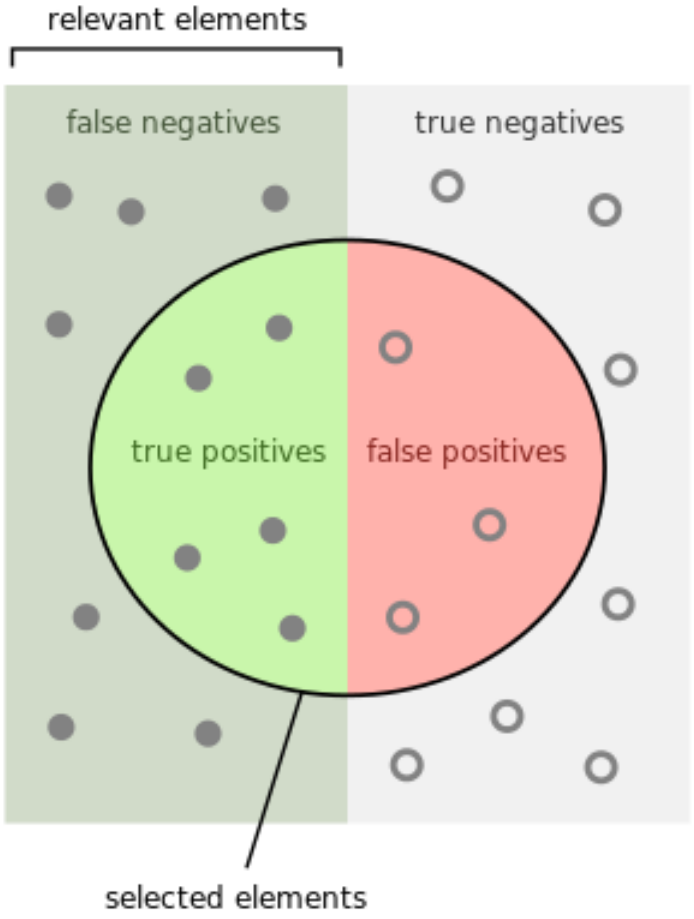
The simplest example I know

```
from tensorflow.keras import models
from tensorflow.keras.models import Sequential
from tensorflow.keras import layers
from tensorflow.keras.layers import Convolution2D
from tensorflow.keras.layers import MaxPooling2D
from tensorflow.keras.layers import Dense, Dropout, Flatten, Activation


model = Sequential()
model.add(Conv2D(32, kernel_size=(5, 5), strides=(1, 1),
                 activation='relu',
                 input_shape=input_shape))
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
model.add(Conv2D(64, (5, 5), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(1000, activation='relu'))
model.add(Dense(num_classes, activation='softmax'))
```

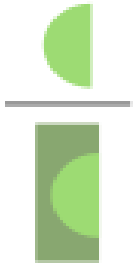
The simplest example I know

```
batch_size = 128
num_classes = 10
epochs = 10

# input image dimensions
img_x, img_y = 28, 28

# load the MNIST data set, which already splits into train and test sets
for us
(x_train, y_train), (x_test, y_test) = mnist.load_data()

model.fit(x_train, y_train,
          batch_size=batch_size,
          epochs=epochs,
          verbose=1,
          validation_data=(x_test, y_test),
          callbacks=[history])
score = model.evaluate(x_test, y_test, verbose=0)
```

Results Metrics



How many relevant items are selected?

Recall = 

Sensitivity, TPR, Recall, Completeness

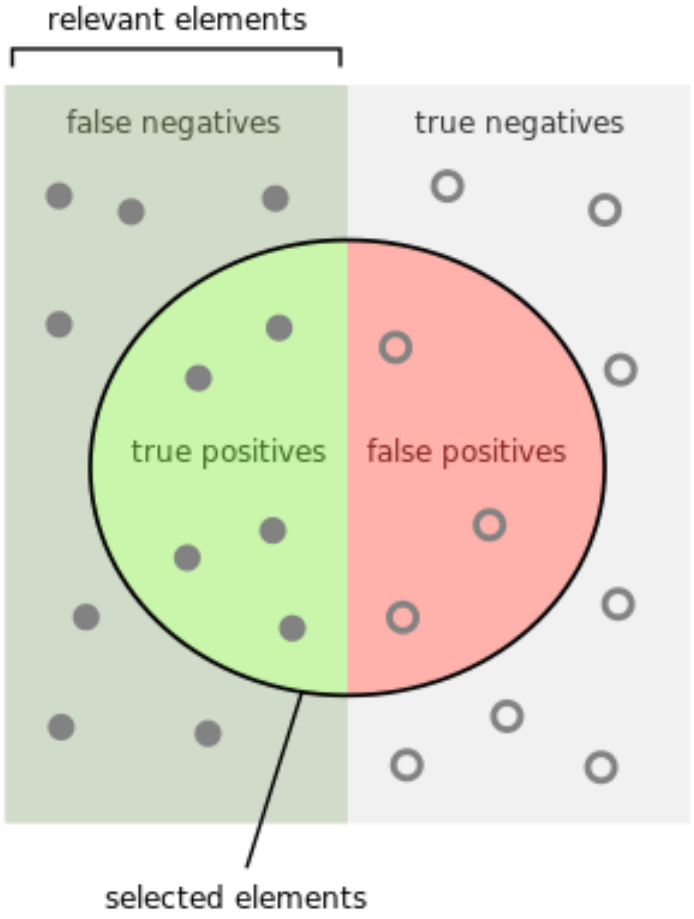
$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{P}} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} = 1 - \mathrm{FNR}$$

False Alarm rate= 

FPR, False Alarm Rate

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{N}} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} = 1 - \mathrm{TNR}$$

Results Metrics



Precision, Purity

How many selected items are relevant?

$$\text{Precision} = \frac{\text{true positives}}{\text{selected elements}}$$

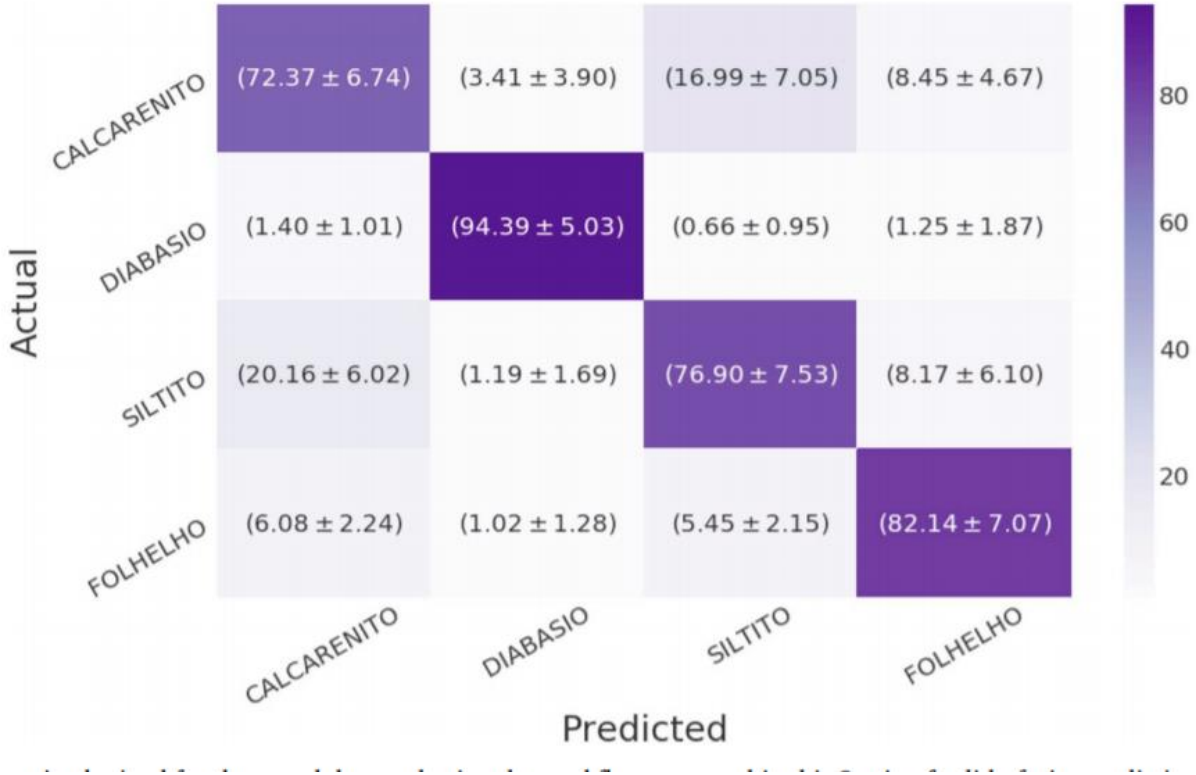
Sensitivity, TPR, Recall, Completeness

How many relevant items are selected?

$$\text{Recall} = \frac{\text{true positives}}{\text{relevant elements}}$$

Confusion Matrix

		Actual class		
		Cat	**Dog**	**Rabbit**
Predicted class	**Cat**	**5**	2	0
	Dog	3	**3**	2
	Rabbit	0	1	**11**



Evaluating your model

True Positive (TP): • Reality: A wolf threatened. • Shepherd said: "Wolf." • Outcome: Shepherd is a hero.	**False Positive (FP):** • Reality: No wolf threatened. • Shepherd said: "Wolf." • Outcome: Villagers are angry at shepherd for waking them up.
False Negative (FN): • Reality: A wolf threatened. • Shepherd said: "No wolf." • Outcome: The wolf ate all the sheep.	**True Negative (TN):** • Reality: No wolf threatened. • Shepherd said: "No wolf." • Outcome: Everyone is fine.

Source: https://developers.google.com/machine-learning/crash-course/classification/true-false-positive-negative

Centro Brasileiro de Pesquisas Físicas

CBPF

Inteligência artificial e aplicações em física

Clécio Roque De Bom – debom@cbpf.br

Elisangela L. Faria
(CBPF)

clearnightsrthebest.com