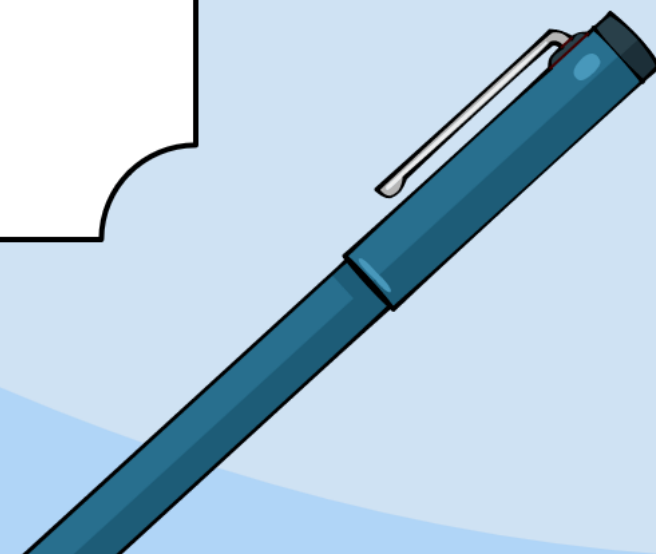




# آموزش کاهش

## نویز تصاویر

محمد دهقانی





# معرفی

۱. مدیر دیتاهاب

۲. لیسانس نرم افزار از دانشگاه اصفهان و ارشد IT تربیت مدرس

۳. کارشناس سابق پردازش متن شرکت لایف وب

۴. سابقه همکاری با شرکت های داده پردازی آرون و توانمند

۵. سابقه تدریس در مرکز علوم شناختی (IPM) و دانشگاه های شریف، اصفهان، امیرکبیر،

شهرکرد، علوم پزشکی تهران و کنفرانس وب پژوهی

۶. دارای بیش از ۳ مقاله ژورنالی

۷. مترجم کتاب یادگیری ماشین

۸. نویسنده کتاب تحلیل عواطف با استفاده از تکنیک های یادگیری ماشین

# این دوره مناسب چه افرادی هست؟

۱. دانشمندان داده

۲. مهندسين داده

۳. برنامه نویسان



# پیش نیازهای یادگیری این دوره

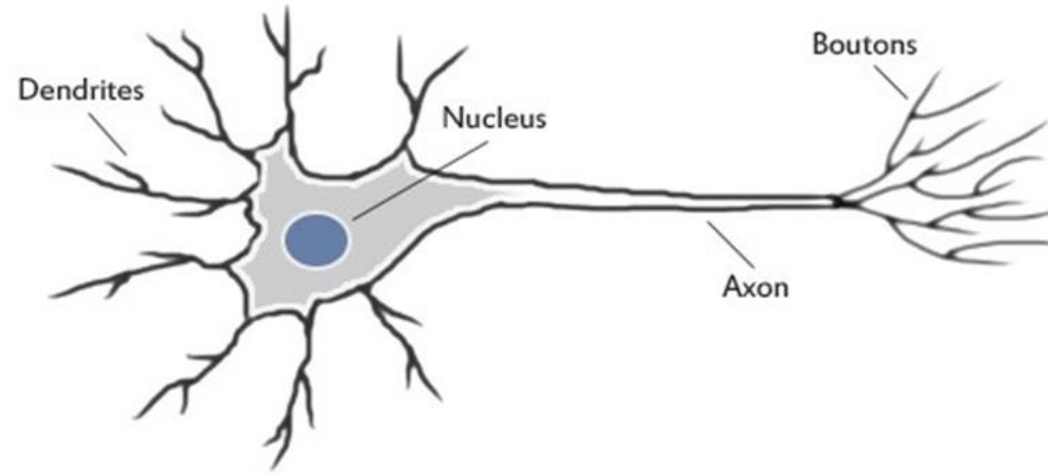
۱. آشنایی با پایتون

۲. آشنایی با برنامه نویسی شی گرا

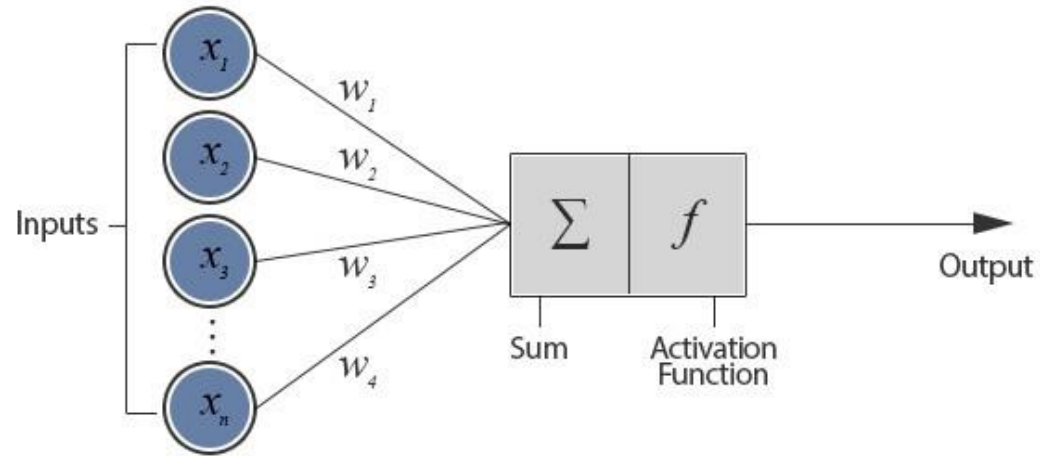
۳. آشنایی با شبکه های عصبی



# A neuron in our brain

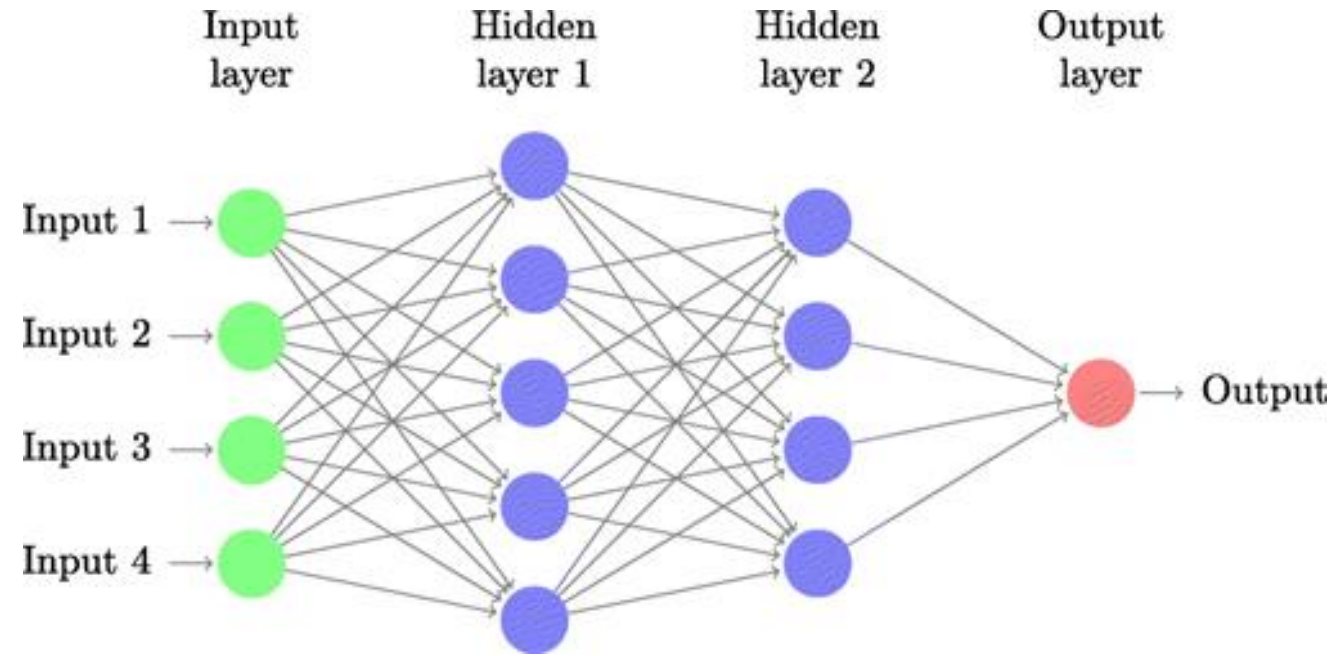


# Perceptron



$$f \left( b + \sum_{i=1}^n x_i w_i \right)$$

# Multi Layer Perceptron

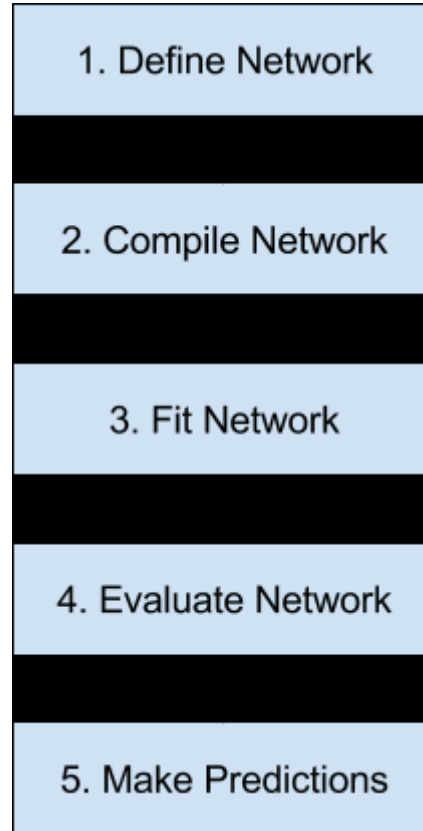


# Keras

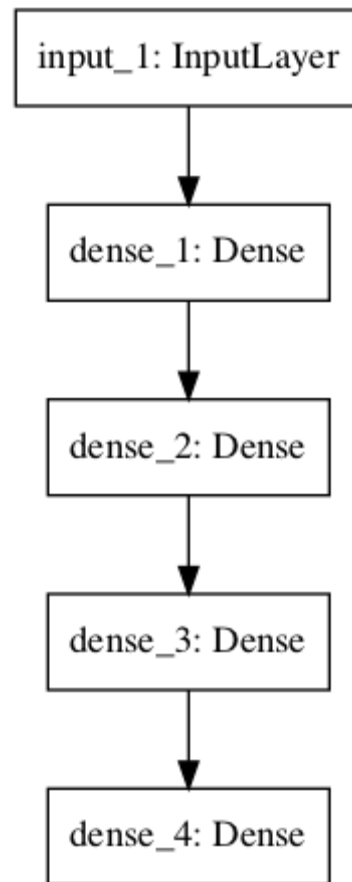




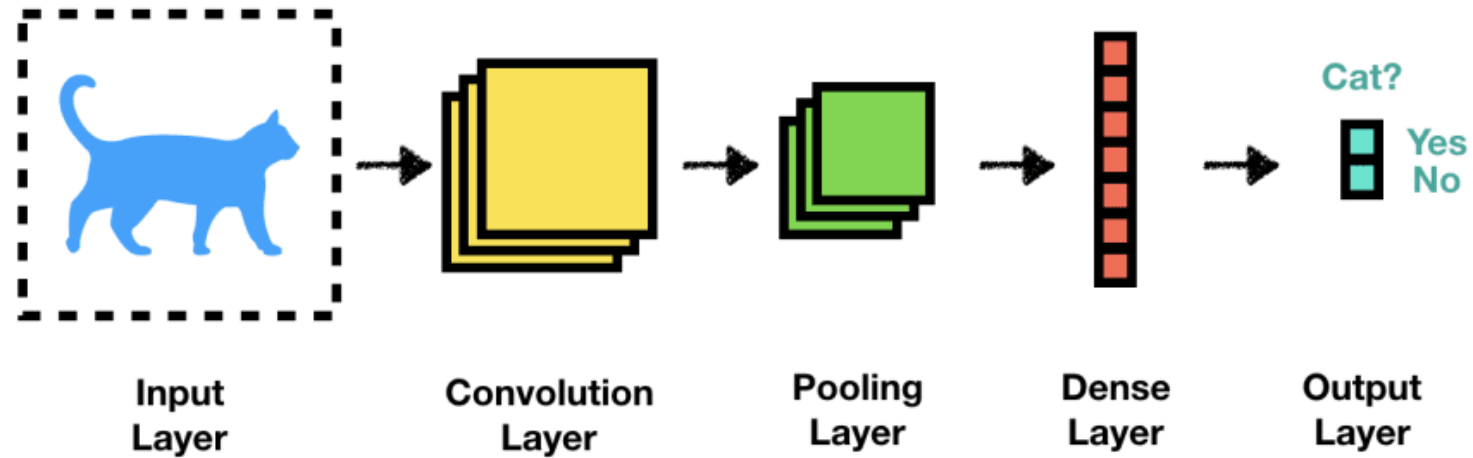
# Keras



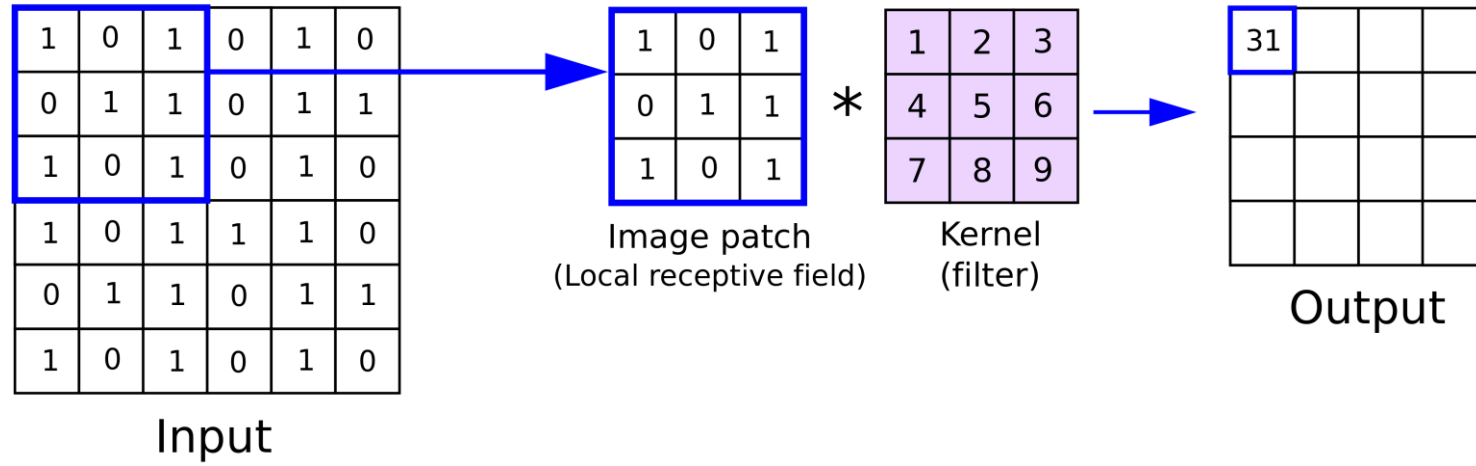
# Keras



# Convolutional Neural Networks



# Filters



# Zero Padding

"VALID" = without padding:

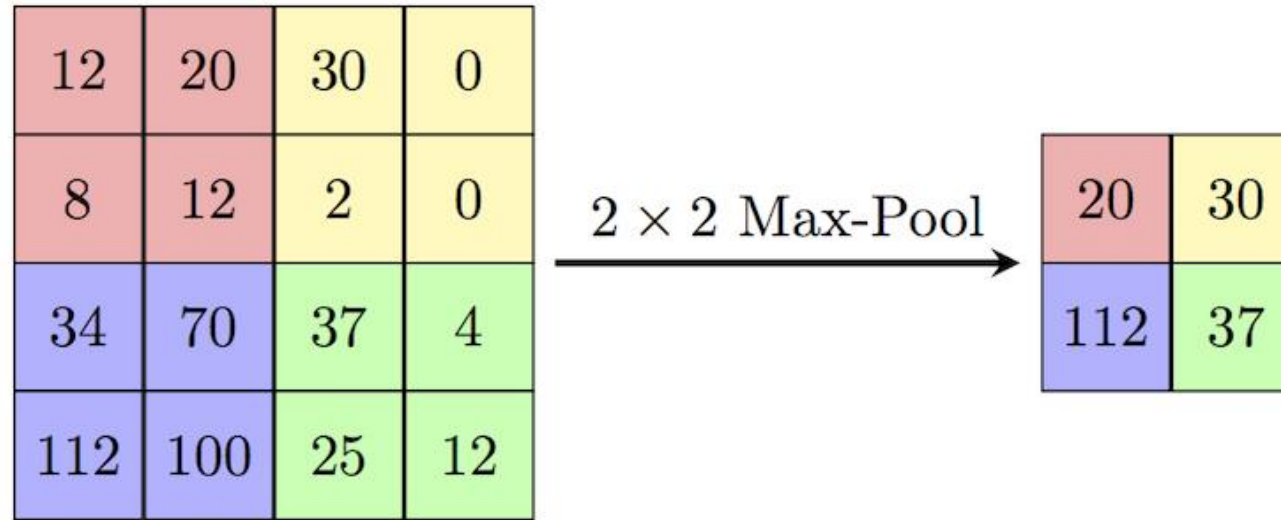
inputs:        1 2 3 4 5 6 7 8 9 10 11 (12 13)  
                 |\_\_\_\_\_|                        dropped  
                 |\_\_\_\_\_|

"SAME" = with zero padding:

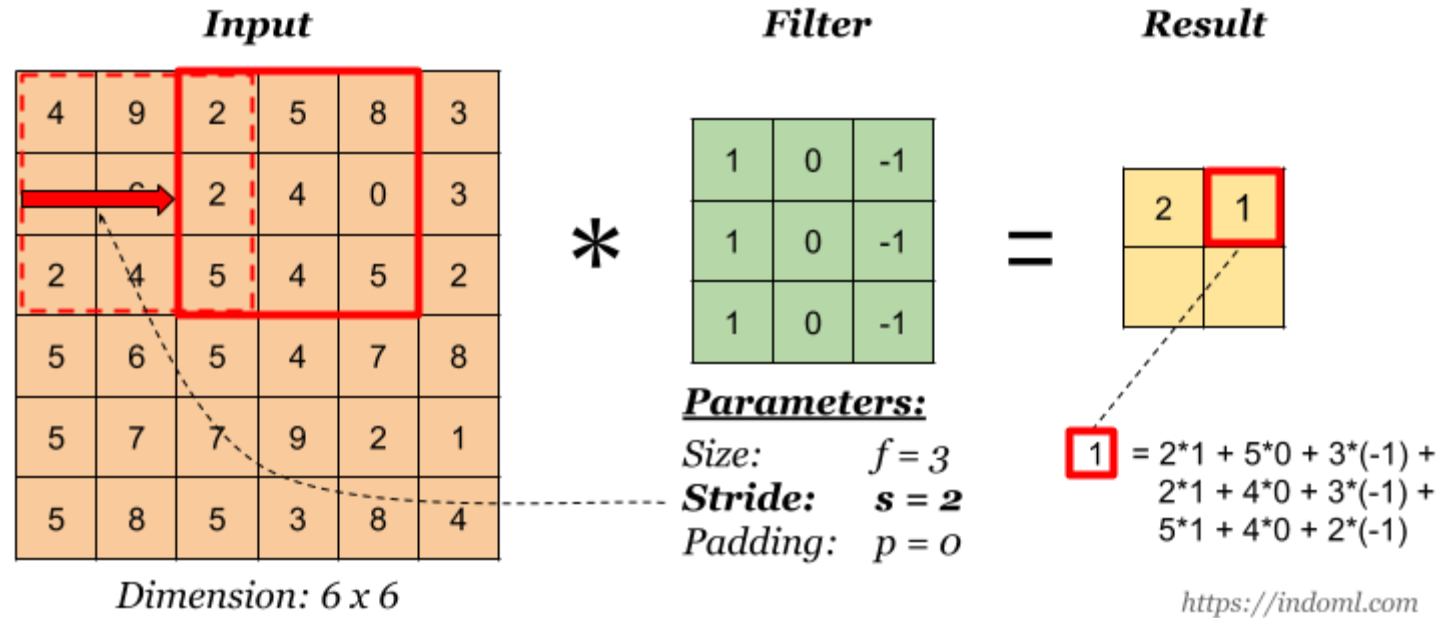
              pad|                                        |pad  
inputs:       0 | 1 2 3 4 5 6 7 8 9 10 11 12 13 | 0 0  
              |\_\_\_\_\_|                                        |\_\_\_\_\_|

0	0	0	0	0	0
0	35	19	25	6	0
0	13	22	16	53	0
0	4	3	7	10	0
0	9	8	1	3	0
0	0	0	0	0	0

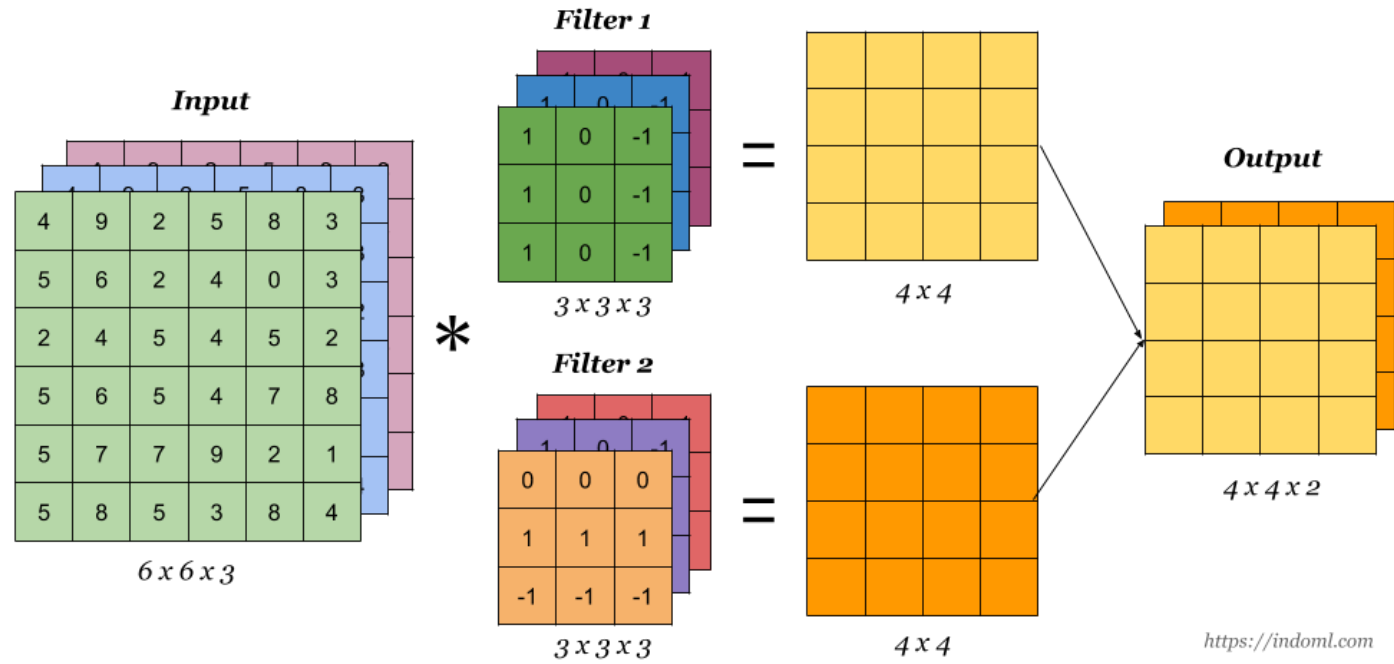
# Pooling Layers



# Stride

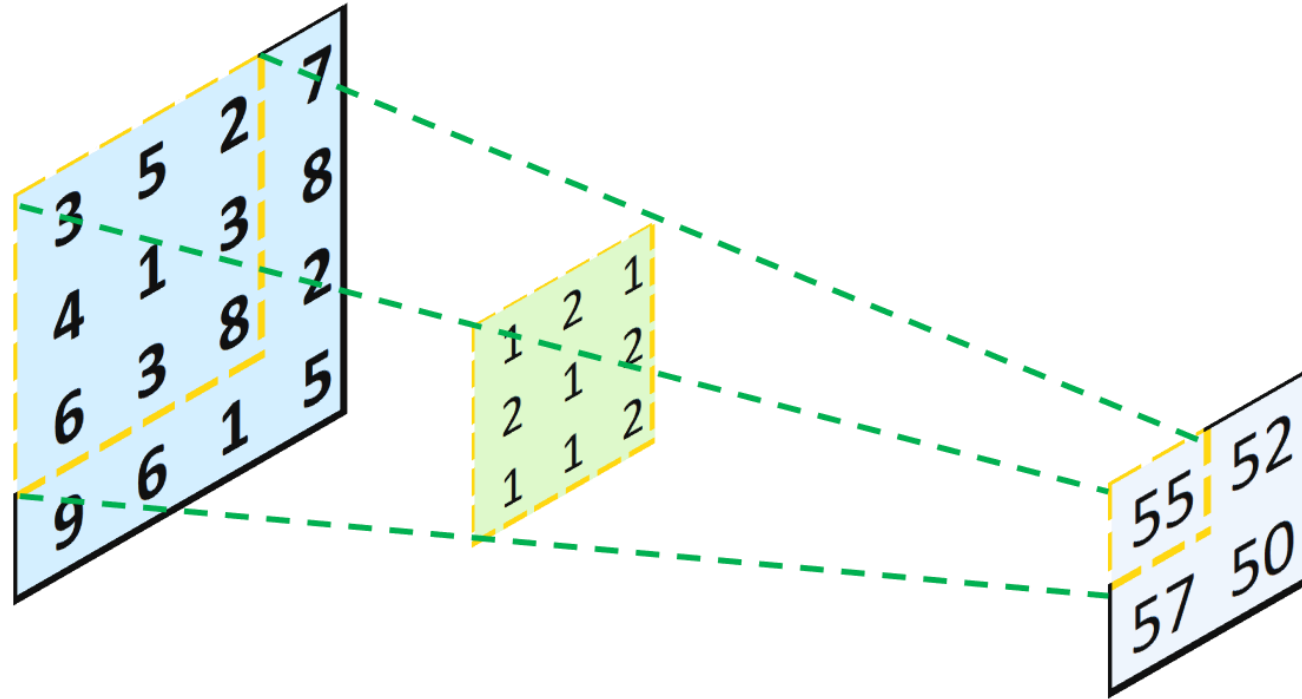


# Filters

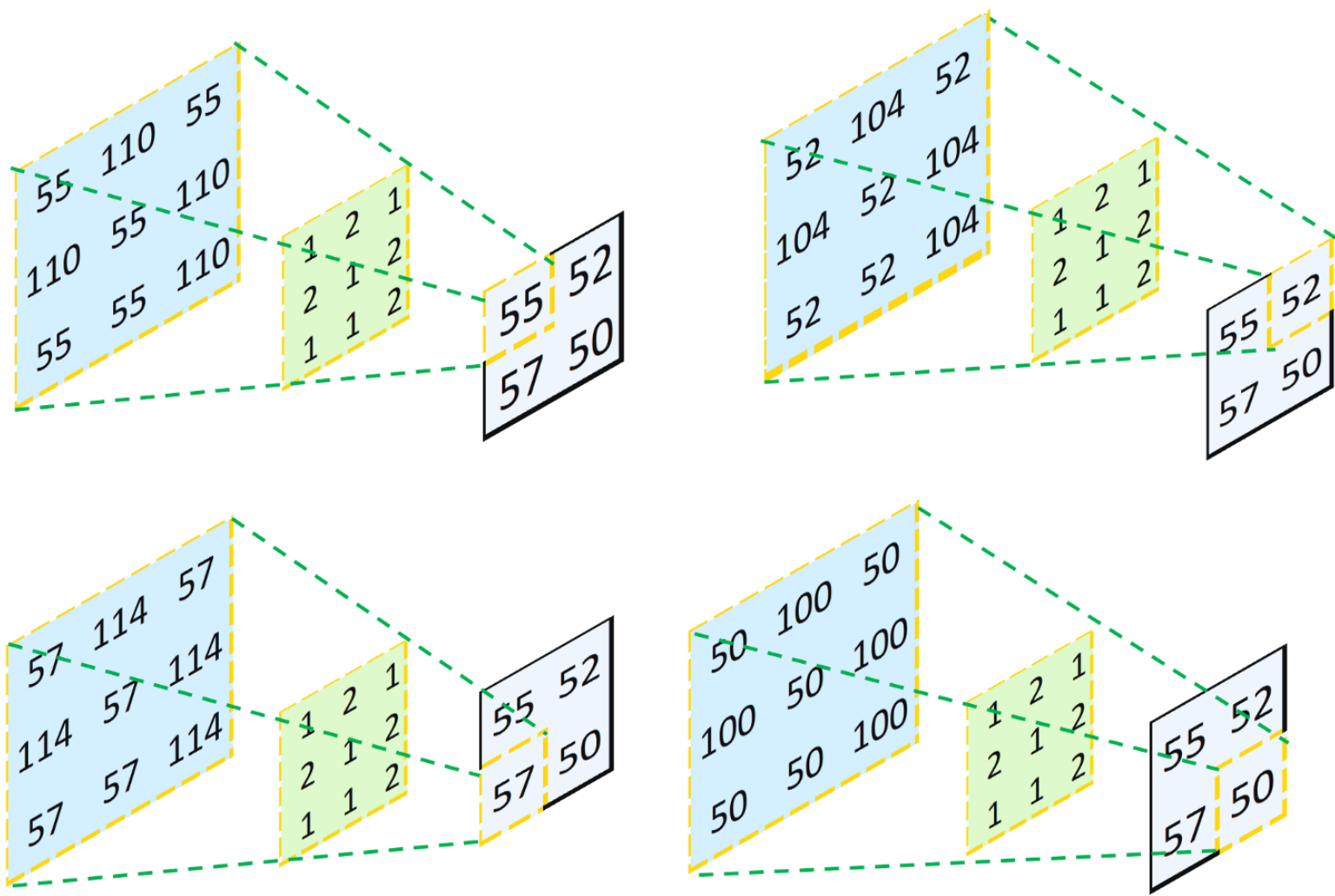




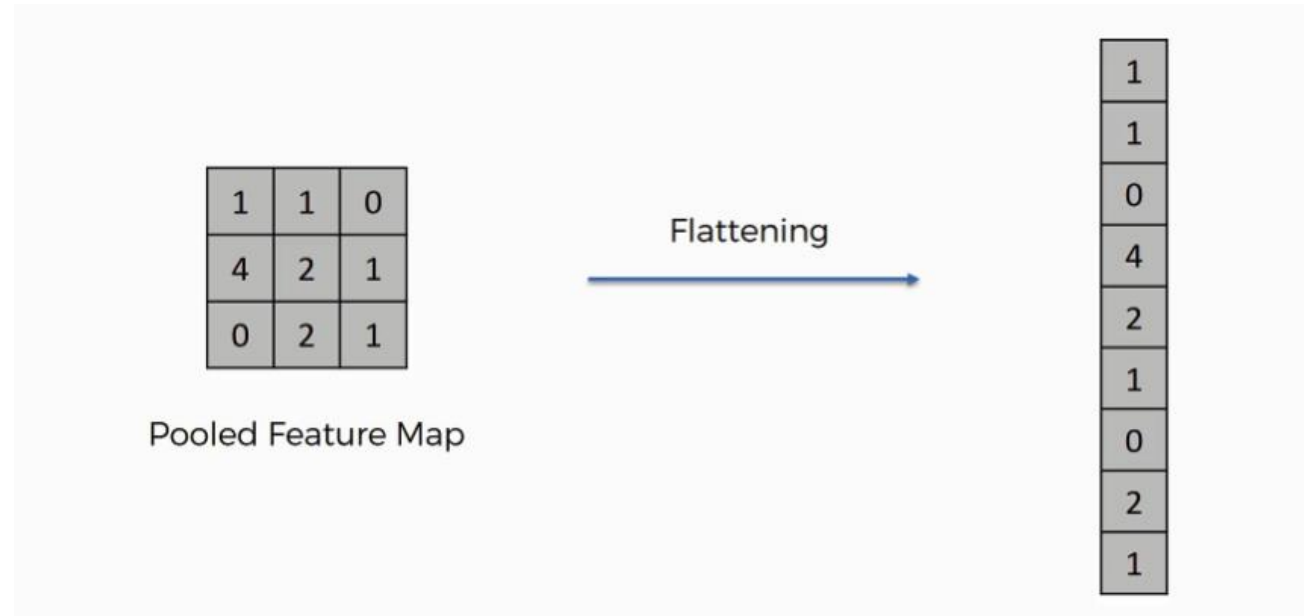
# Filters



# Filters



# Flatten Layers



# Input\_shape

**Dense layers require inputs as  
(batch\_size, input\_size)**

**2D convolutional layers need inputs as:**

**if using channels\_last: (batch\_size, imageside1, imageside2, channels)**

**if using channels\_first: (batch\_size, channels, imageside1, imageside2)**

**1D convolutions and recurrent layers use  
(batch\_size, sequence\_length, features)**

# Conv2D Layers

```
Conv2D(  
    filters,  
    kernel_size,  
    strides=(1, 1),  
    padding="valid",  
)  
    Input_shape=[samples, rows, columns, channels]  
    [1, 4, 4, 1]  
  
    model = Sequential()  
    model.add(Conv2D(1, (3, 3), input_shape=(4, 4, 1)))
```

# Conv1D Layers

`[0, 0, 0, 1, 1, 0, 0, 0]`

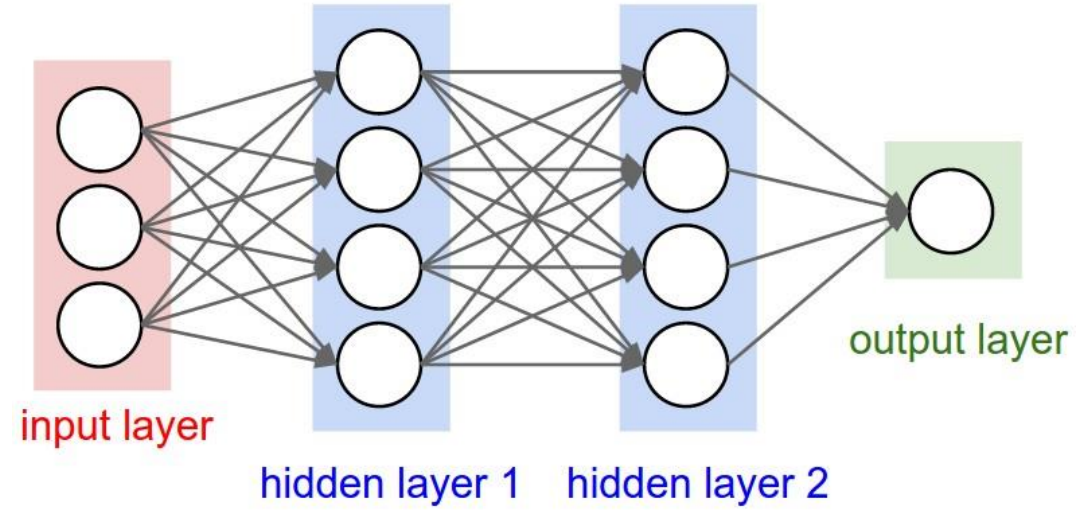
`[1, 8, 1]`

```
model = Sequential()  
model.add(Conv1D(1, 3, input_shape=(8, 1)))
```

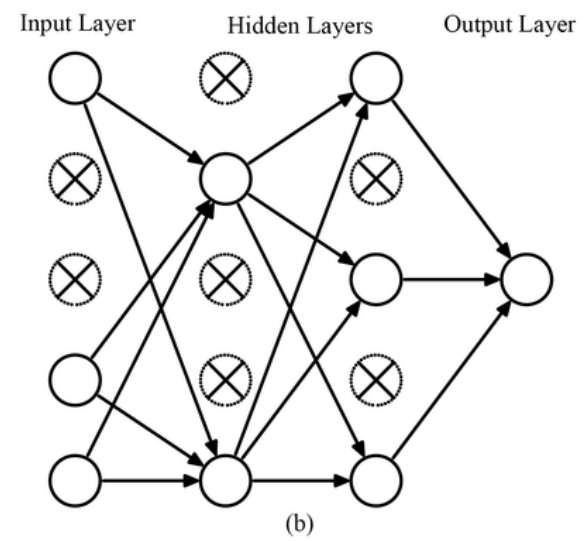
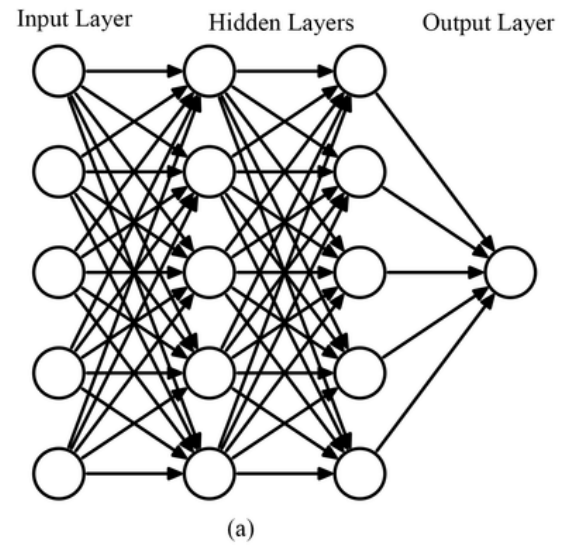
`[0, 1, 0] . [0, 0, 0] = 0`

`[0, 0, 1, 1, 0, 0]`

# Input & Output

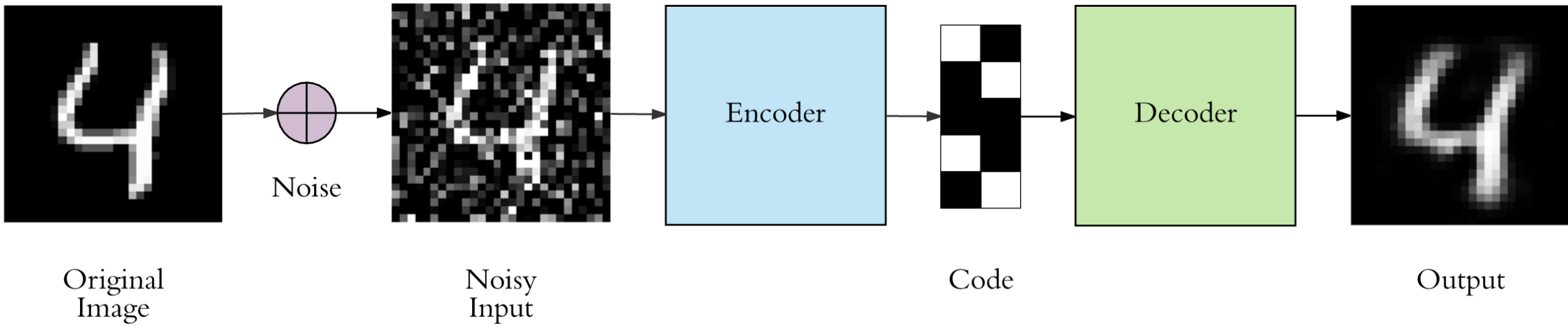


# Dropout

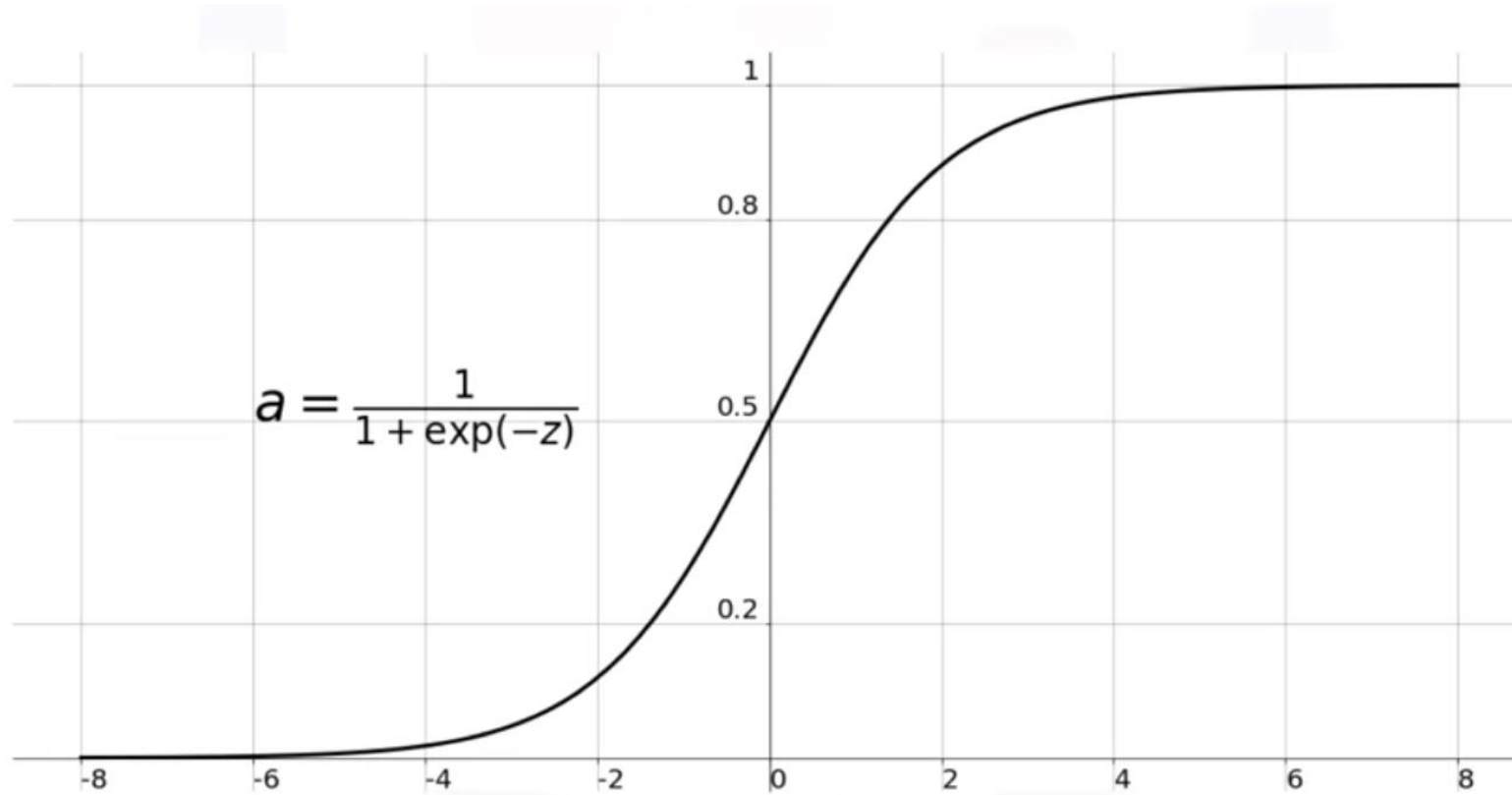




# Autoencoder



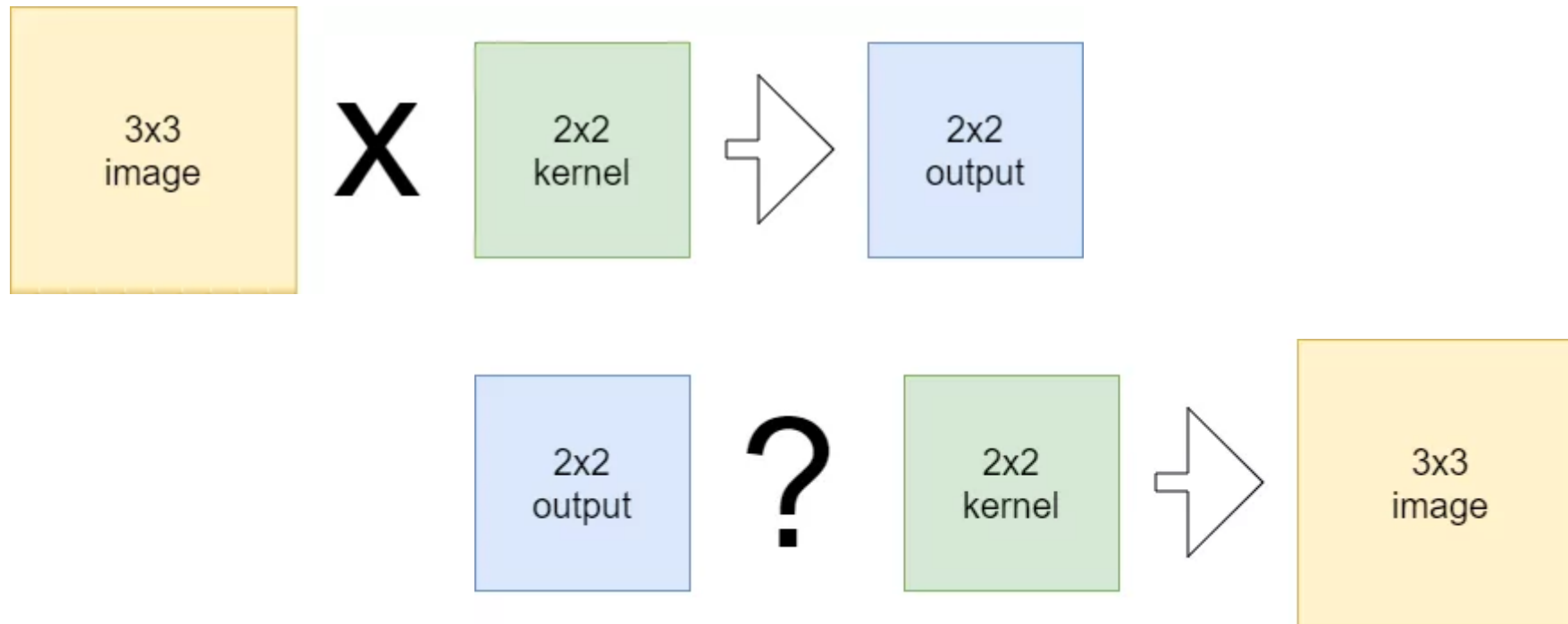
# Sigmoid



# Unsampling

given a summary, i.e. the result of the convolution,  
reconstructing the original input?

We call this “upsampling”.



**#DONTFORGETUS**

آموزش های  
رایگان بیشتر

[www.data-hub.ir](http://www.data-hub.ir)

[www.youtube.com/channel/datahub1](http://www.youtube.com/channel/datahub1)

[www.linkedin.com/company/data-hub-ir](http://www.linkedin.com/company/data-hub-ir)

[www.t.me/data hub ir](http://www.t.me/data_hub_ir)

[www.github.com/datahub-ir](http://www.github.com/datahub-ir)