CFI
SUMMER SCHOOL '22

github.com/analytics-club-iitm

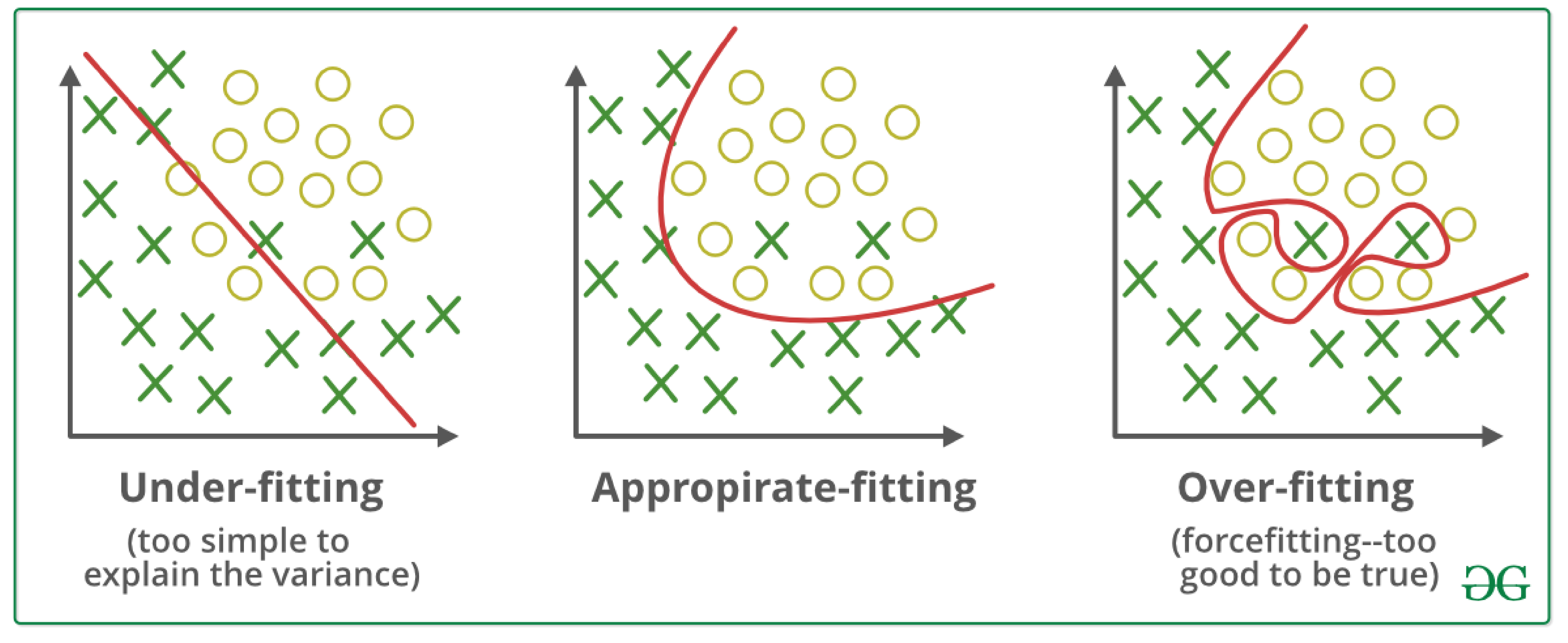
github.com/iitmcvg

CONTENTS

- REGULARIZATION
- OPEN-CV

THE WHAT AND WHY OF REGULARIZATION

- Regularization is a technique which is done in order to overcome over-fitting.
- Even if a model doesn't overfit on a "proper" training data, if some noise is added to the training data, it tends to overfit



THE WHAT AND WHY OF REGULARIZATION

Now onto the formal definition:

- Regularization is a form of regression which reduces the coefficient estimates
- This inturn reduces the possibility of learning a more complex model
- Regularization is done by adding a penalty term to the loss function which depends upon the type of Regularization
- This penalty term controls the excessively fluctuating function by reducing the coefficients of higher dgree terms more quickly.

TYPES OF REGULARIZATION

- Regularization can classified into different types depending on the type of the penalty term.

For L_n,

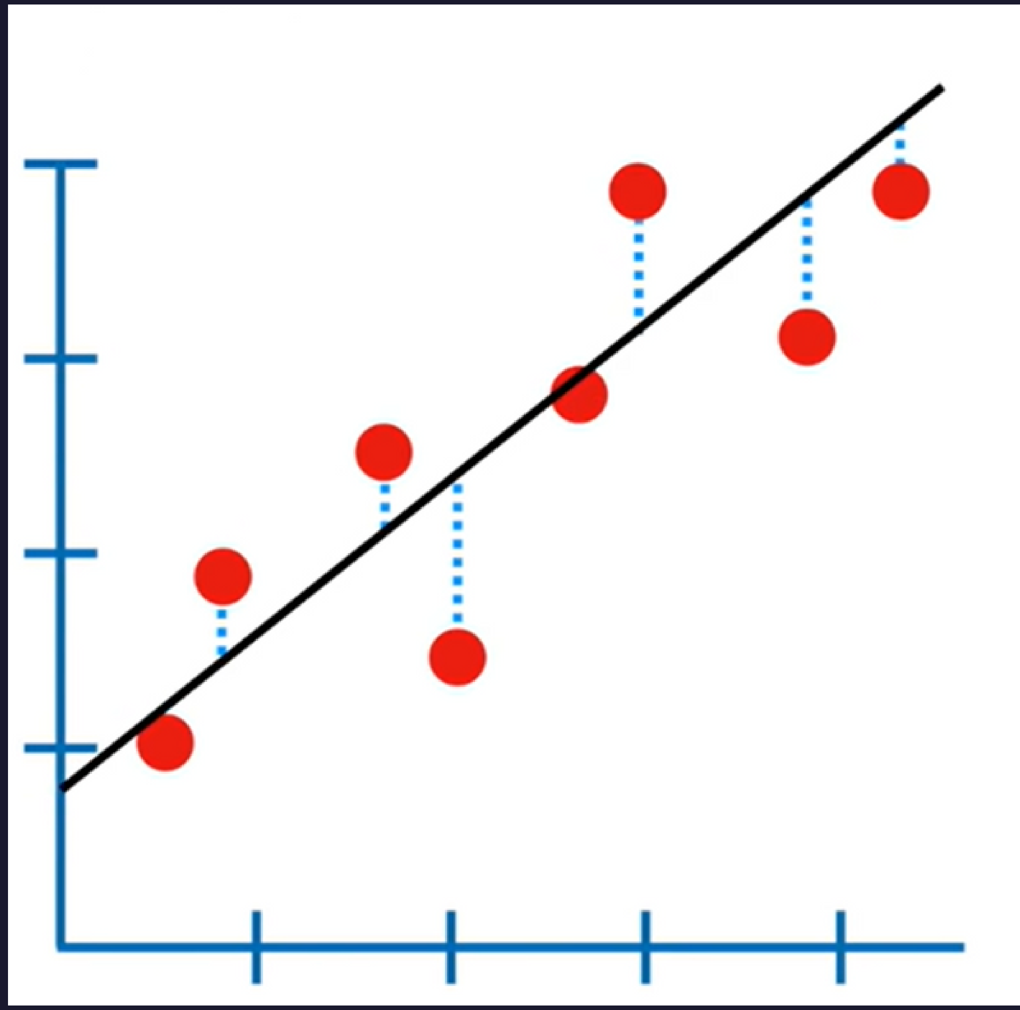
$$\text{Penalty term} = \lambda(\Sigma|\text{parameter}|^n)$$

(λ is the "regularization hyper-parameter")

- The most commonly used types are L1 and L2
- L1 regularization is also known as Lasso Regression
- L2 regularization is also known as Ridge Regression

RIDGE REGRESSION(L2)

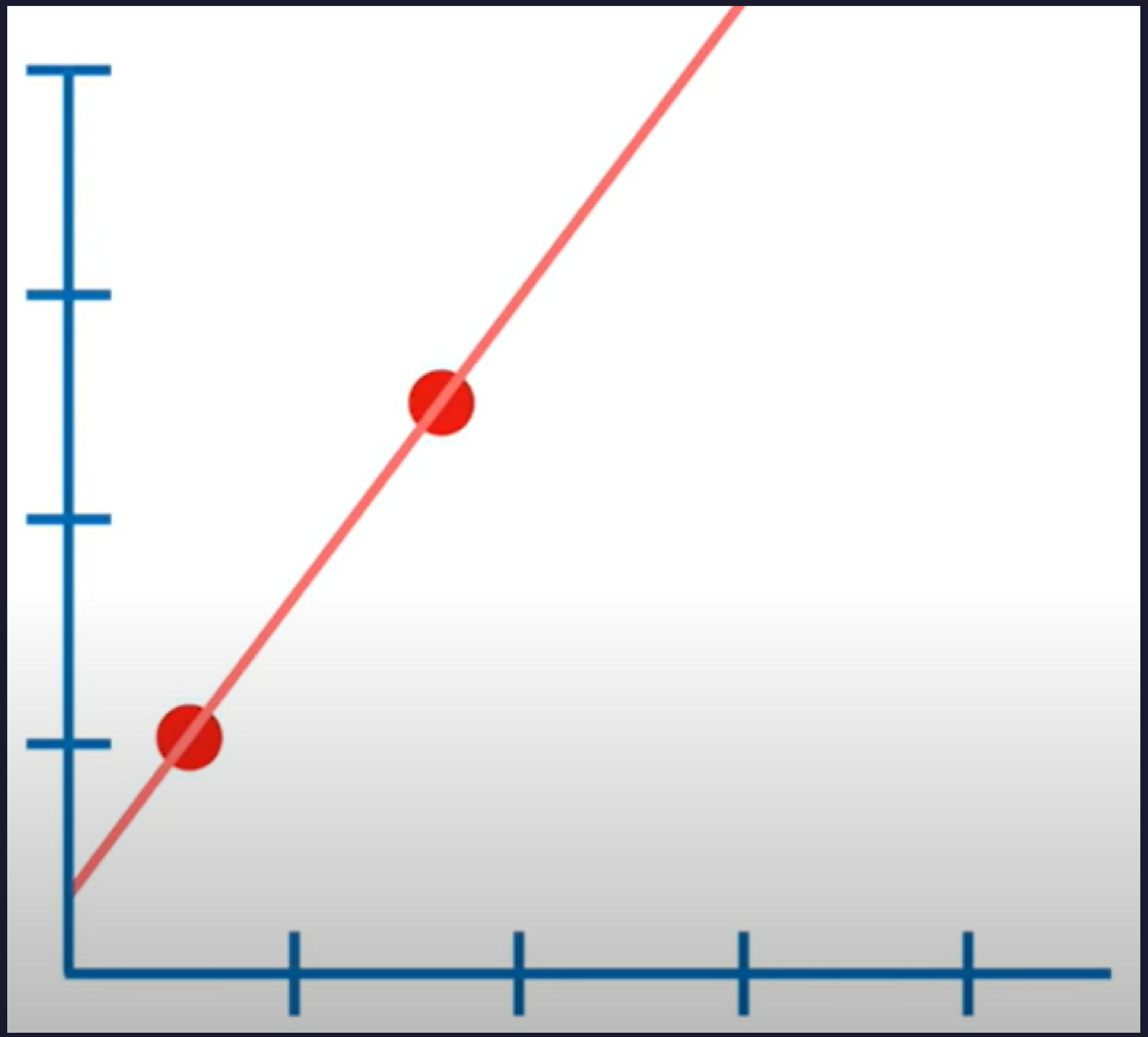
Let's try to understand this using a simple case of linear regression.



In this case, linear regression works fine

RIDGE REGRESSION(L2)

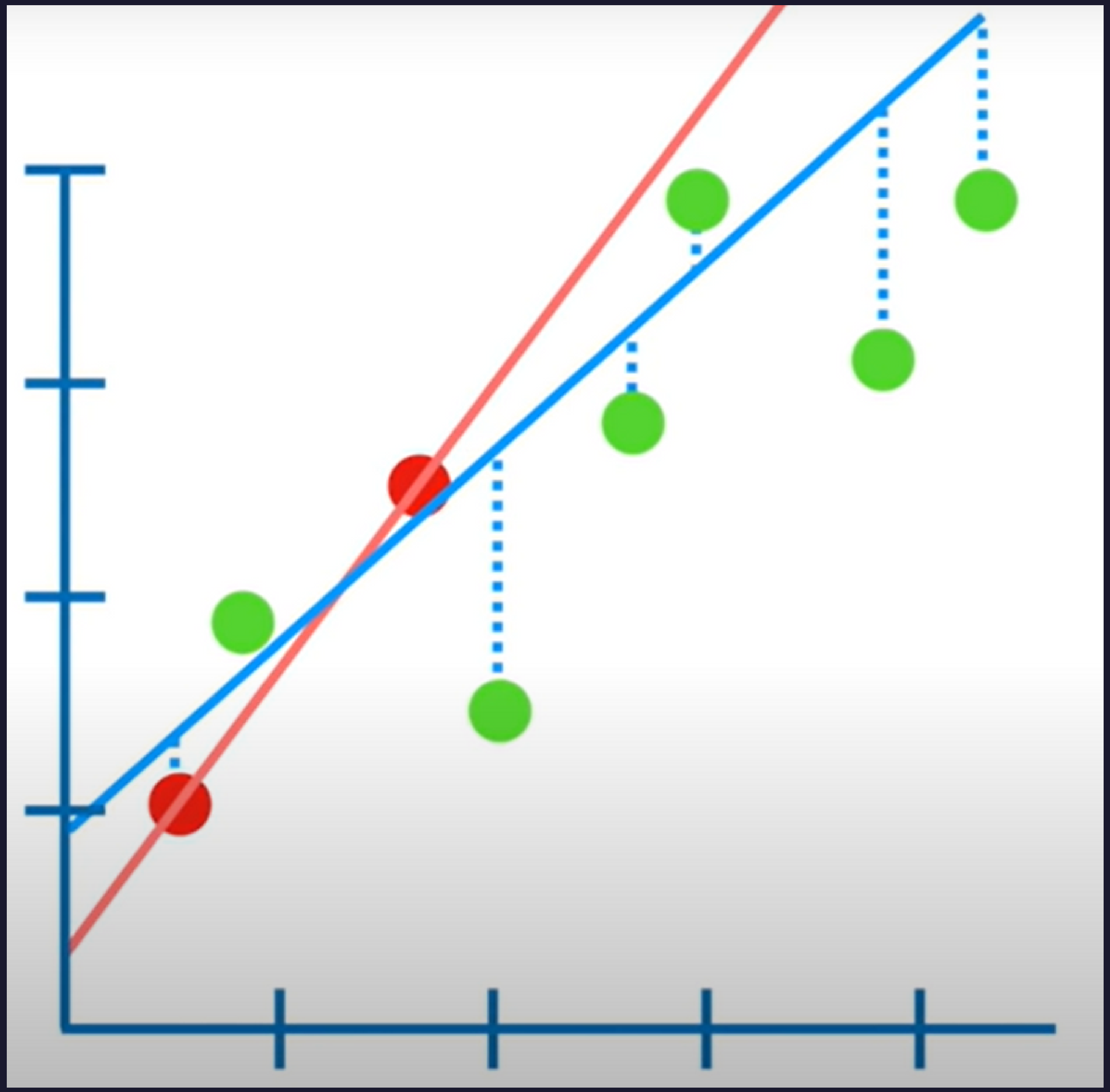
But what about this case?



In this case, the prediction made by linear regression model is clearly wrong. This is because the model overfits the training data.

The intuition behind regularization is introducing a small amount of bias to the model to reduce the variance by a significant amount.

RIDGE REGRESSION(L2)



The line obtained after performing regularization is clearly better than the one predicted by the traditional linear regression model.

$$\text{Penalty term} = \lambda(\Sigma|\text{parameter}|^2)$$

LASSO REGRESSION(L1)

Behaves in the same way as Ridge regression. Except for the penalty term of course.

$$\text{Penalty term} = \lambda(\Sigma|\text{parameter}|)$$

Lasso vs Ridge :

We can define and use as many parameters as we want but all parameters may not contribute to the model significantly.

In lasso, the useless parameters become zero quickly for a finite value of λ. In ridge, all the parameters only tend to 0 asymptotically as λ tends to infinity.

REGULARIZATION HYPER-PARAMETER

Important points on the hyper-parameter:

- λ is non-negative.
- As λ increases, the slope of the line predicted decreases.
- As λ tends to infinity, the slope of the line tends to 0 and the paramerters become redundant in predicting. (not in case of lasso regression)
- In the penalty term, we shouldn't include the y-intercept.
- λ is chosen by 10-fold cross validation.

ELASTIC NET REGERESSION

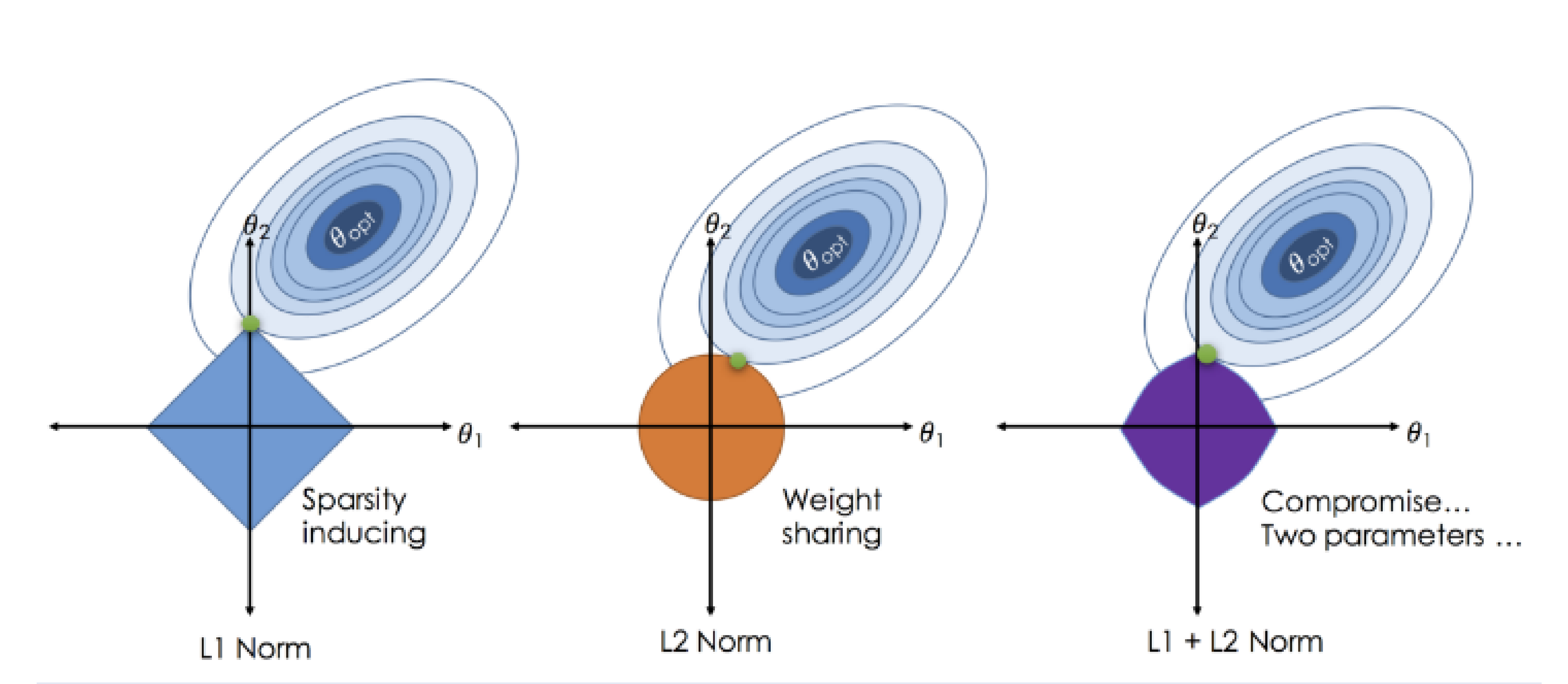
Lasso regression works well when there are a lot of useless parameters and Ridge regression works well when there are a lot of useful parameters. But what if we have both?

This is when elastic net regression comes into the picture. This method is just a combination of lasso and ridge regression (i.e). the penalty term is just a linear combination of L1 and L2.

$$P_\alpha(\beta) = \frac{(1-\alpha)}{2}\|\beta\|_2^2 + \alpha\|\beta\|_1 = \sum_{j=1}^{p}\left(\frac{(1-\alpha)}{2}\beta_j^2 + \alpha|\beta_j|\right).$$

It works better than both when correlated parameters are involved.

RIDGE v/s LASSO v/s ELASTIC



When the ellipse touches the shaded region, the parameter corresponding to the horizontal axis becomes redundant

A VERY IMPORTANT APPLICATION

Whenever we do linear regression, the number of datapoints in training set should be greater than or equal to the number of parameters.

But in most cases, the number of parameters is high and it is practically impossible to get that a dataset which has that many datapoints.

This issue can be solved by regularization. As if we use regularization, we can generate a model without taking in that many number of datapoints.

DROP-OUT

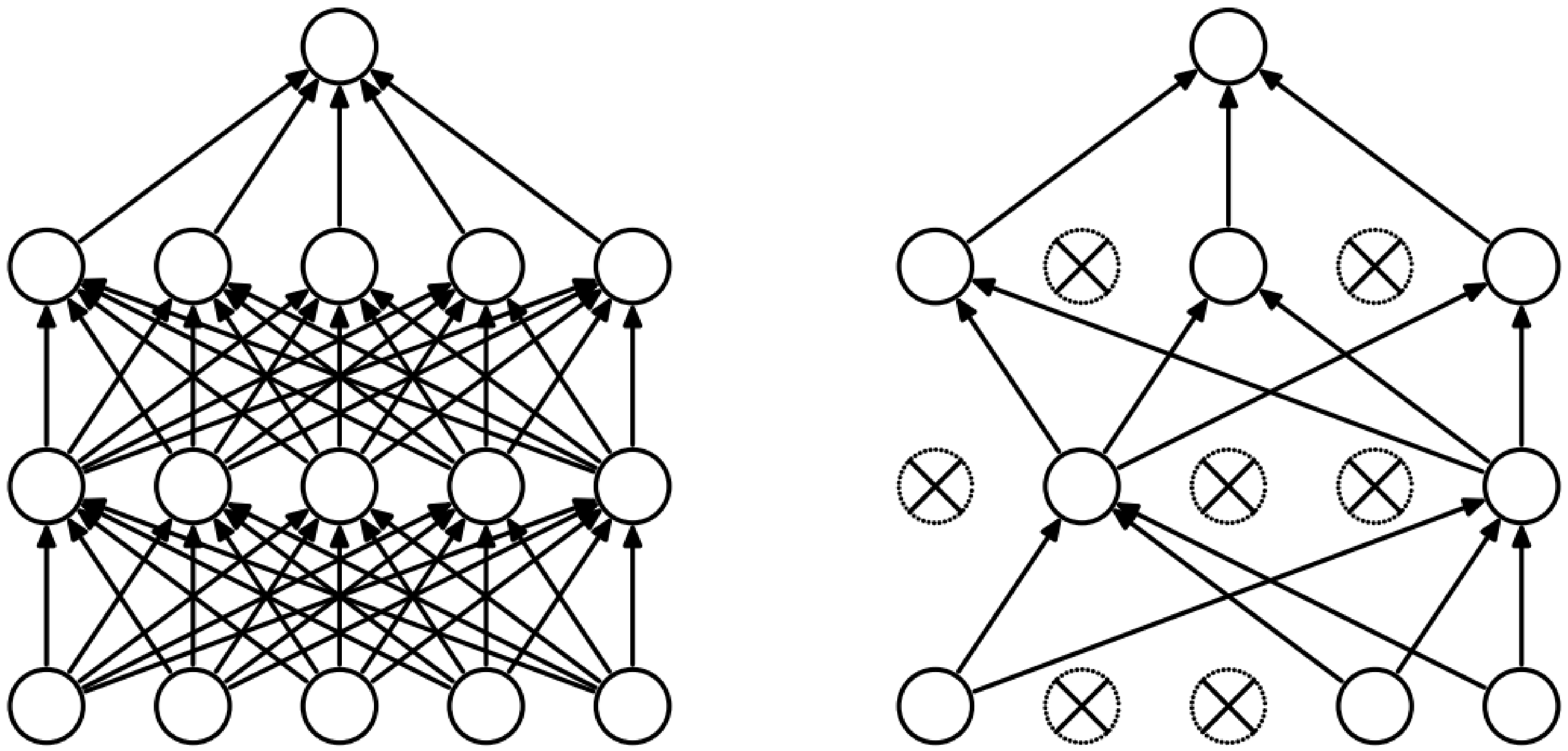
A technique to minimize overfitting in neural networks.
Overfitting in neural networks happen when the network is deep and is trained on relatively small datasets.

The best way to “regularize” a fixed-sized model is to average the predictions of all possible settings of the parameters, weighting each setting by its posterior probability given the training data.

But this would require huge amount of computation..

DROP-OUT

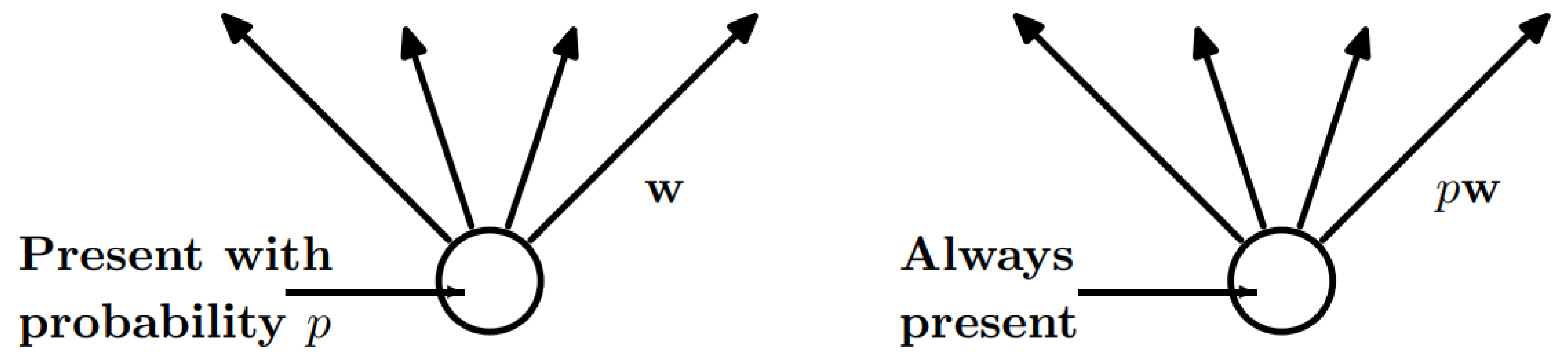
So, as the name suggests, we dropout several nodes of the neural network at a time.


(a) Standard Neural Net

(b) After applying dropout.

DROP-OUT

The choice of dropping a node is random.
While testing the model, all nodes will be present and the weights of the layers emerging from each node should be multiplied by the probability of the node being in the model during training.



(a) At training time (b) At test time

Research paper : "https://jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf"

OPENCV

What is it?

OpenCV is a library of programming functions aimed at real time computer vision.

What is it used for

OpenCV is used in areas powered by AI or ML algorithms that need image processing.

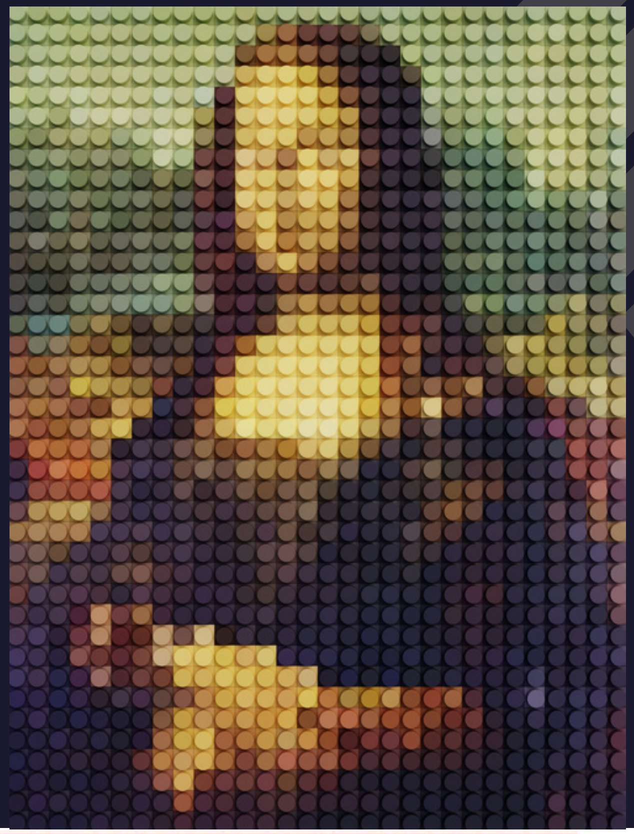
PIXEL REPRESENTATION

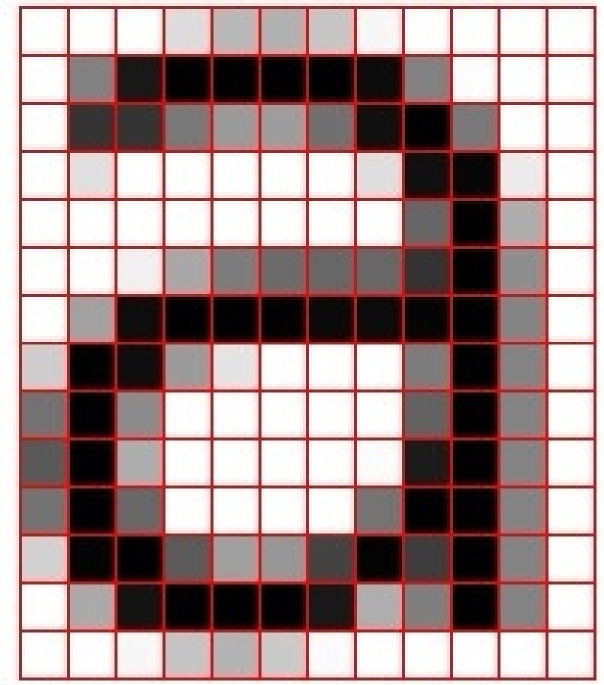
How computers see images

Images are represented as a matrix of pixels. Now a pixel is the most minor controllable component of an image.
So an image is a collection of these pixels.
The dimension of the matrix is :

image height x image width x no. of channels

Now each of these pixels stores a value. These values represent
the color at that particular pixel.





=

1.0	1.0	1.0	0.9	0.6	0.6	0.6	1.0	1.0	1.0	1.0	1.0
1.0	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.5	1.0	1.0	1.0
1.0	0.2	02	0.5	0.6	0.6	0.5	0.0	0.0	0.5	1.0	1.0
1.0	0.9	1.0	1.0	1.0	1.0	1.0	0.9	0.0	0.0	0.9	1.0
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.5	0.0	0.5	1.0
1.0	1.0	1.0	0.5	0.5	0.5	0.5	0.5	0.4	0.0	0.5	1.0
1.0	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5	1.0
0.9	0.0	0.0	0.6	1.0	1.0	1.0	1.0	0.5	0.0	0.5	1.0
0.5	0.0	0.6	1.0	1.0	1.0	1.0	1.0	0.5	0.0	0.5	1.0
0.5	0.0	0.7	1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.5	1.0
0.6	0.0	0.6	1.0	1.0	1.0	1.0	0.5	0.0	0.0	0.5	1.0
0.9	0.1	0.0	0.6	0.7	0.7	0.5	0.0	0.5	0.0	0.5	1.0
1.0	0.7	0.1	0.0	0.0	0.0	0.1	0.9	0.8	0.0	0.5	1.0
1.0	1.0	1.0	0.8	0.8	0.9	1.0	1.0	1.0	1.0	1.0	1.0

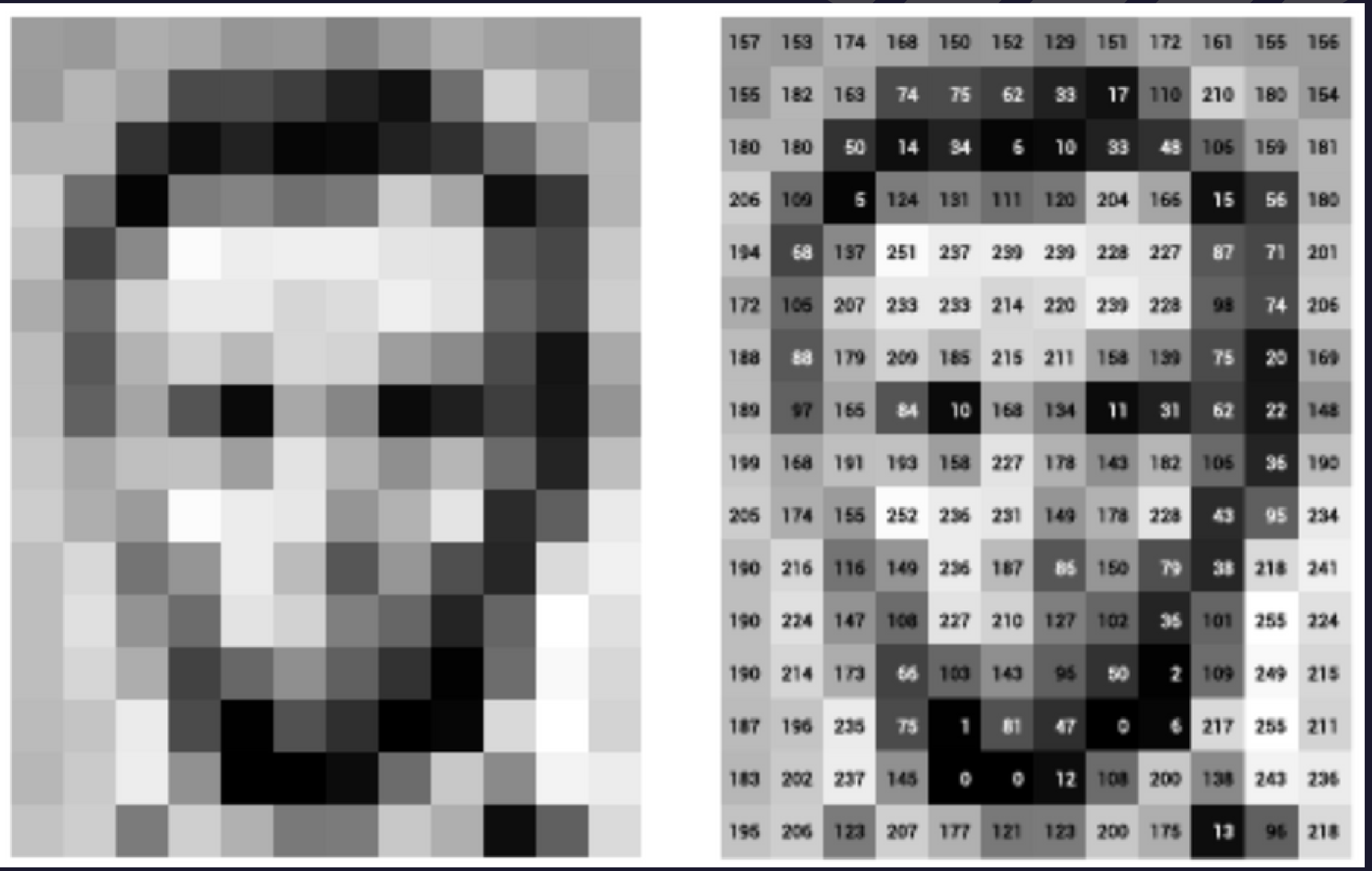
COLORSPACES

The various ways in which we can represent images

Colorspaces are mathematical models describing how colors can be represented, basically a mapping from a certain colors to a set of numbers (usually in 8 bit size).

Grayscale

Each pixel has one channel with a 8 bit number from 0 to 255. A pixel value of 0 corresponds to black and a value of 255 corresponds to white.
Anything else represents the values in between.



COLORSPACES

The various ways in which we can represent images

BGR/RGB

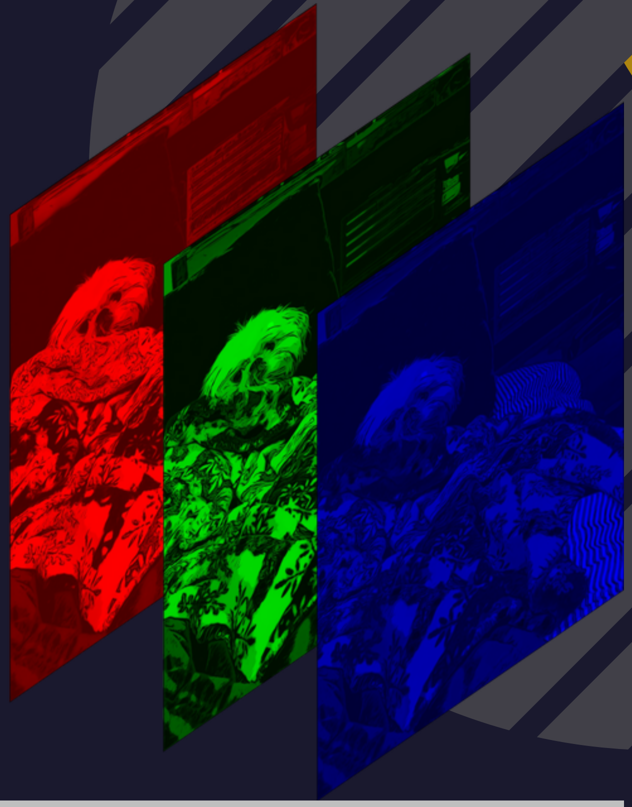
In this colorspace, each pixel has 3 channels. One for blue, red and green respectively.

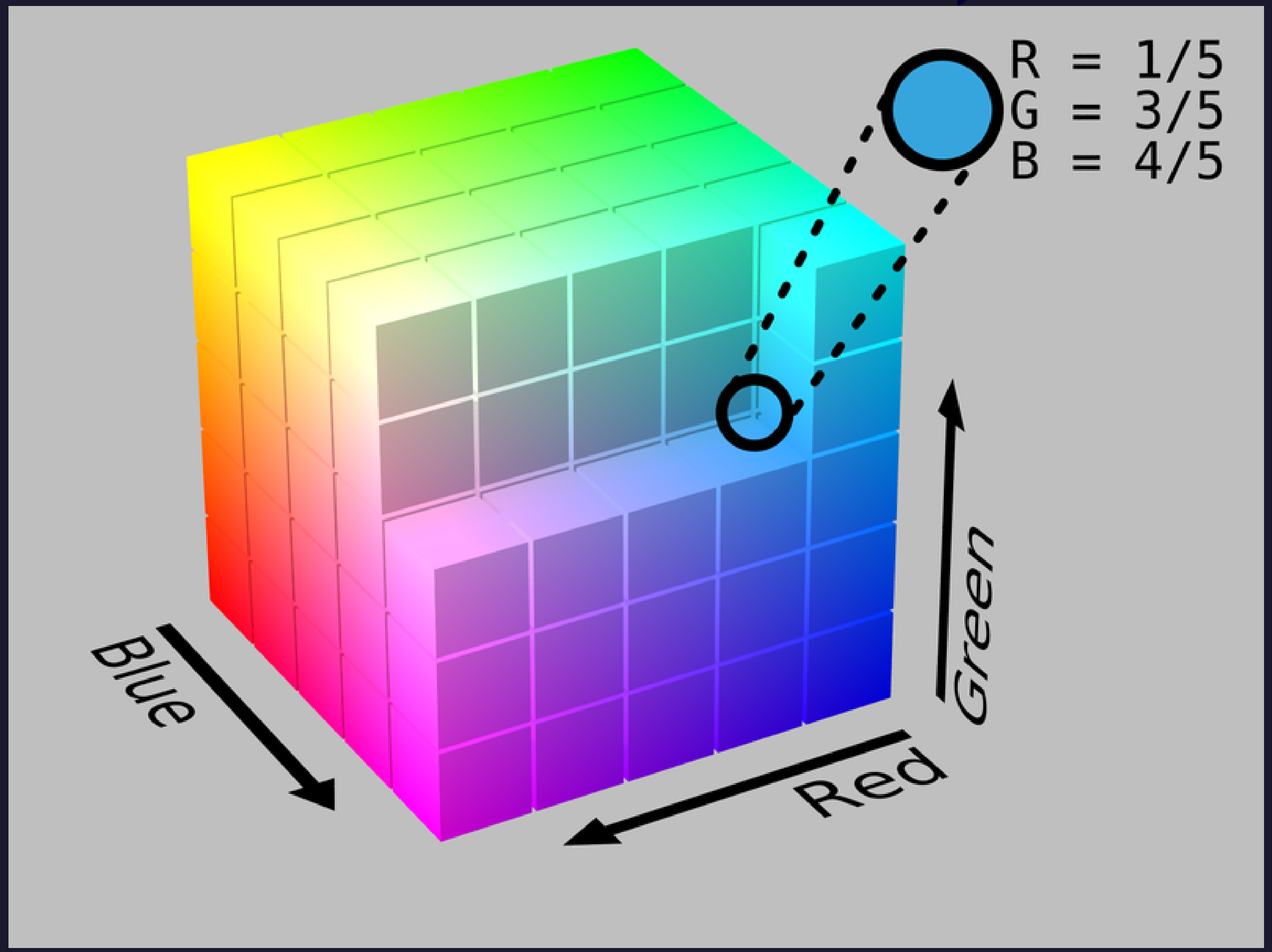
R: 0-255

G: 0-255

B: 0-255.

Each number between 0-255 corresponds to a shade of that color. A combination of the 3 values is used to represent the image.



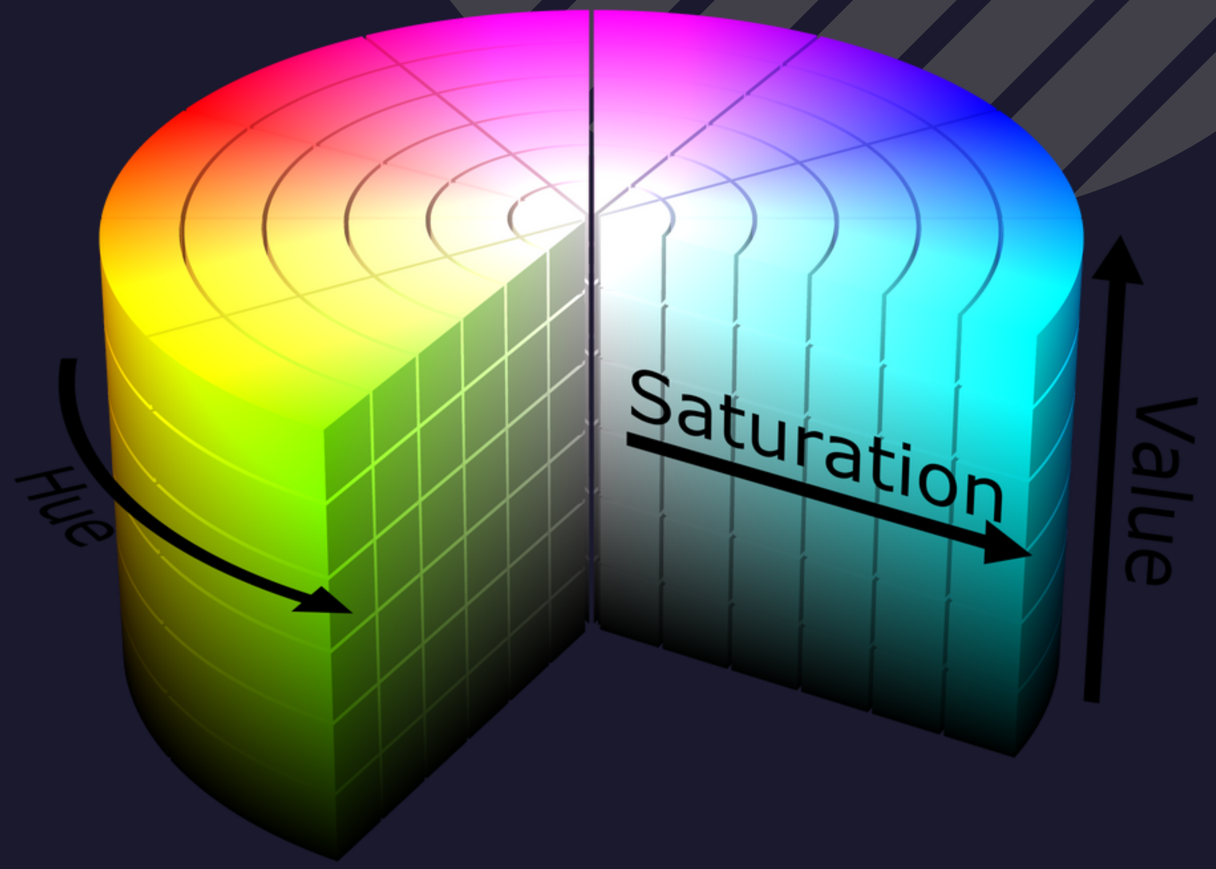


COLORSPACES

The various ways in which we can represent images.

HSV

HSV colorspace is a cylindrical model that remaps RGB values
to
dimensions that are easier to work with in cases of object
detection.
These dimensions are hue,saturation and value.
Hue: specifies the angle on the hsv colorcircle. 0 corresponds to
red 120 corresponds to green and 240 corresponds to blue
Saturation: specifies the amount of color used. 100%
 corresponds to the purest shade used.
Value: controls the brightness of the color.



CONVOLUTION FILTERS

The basics to image processing

The process of convolution is drawn from signal processing where basic 1D convolution was applied to incoming signal.

1D CONVOLUTION

Say we have a 1D filter kernel G and a matrix F on which we want to apply the filter, then in 1D convolution

1. we first invert the kernel G
2. multiply it with the matrix F from which we wish to start
3. take a stride of n steps horizontally till all the cells of the matrix are covered and do padding if necessary

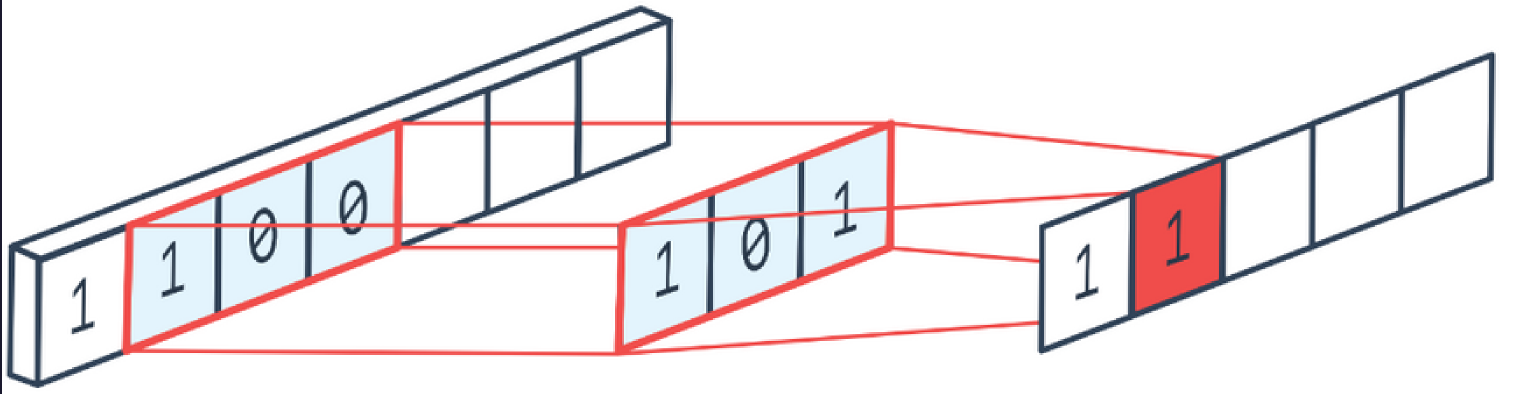


Figure 1. A 1D convolution with a kernel sized 3 and stride 1.

CONVOLUTION FILTERS

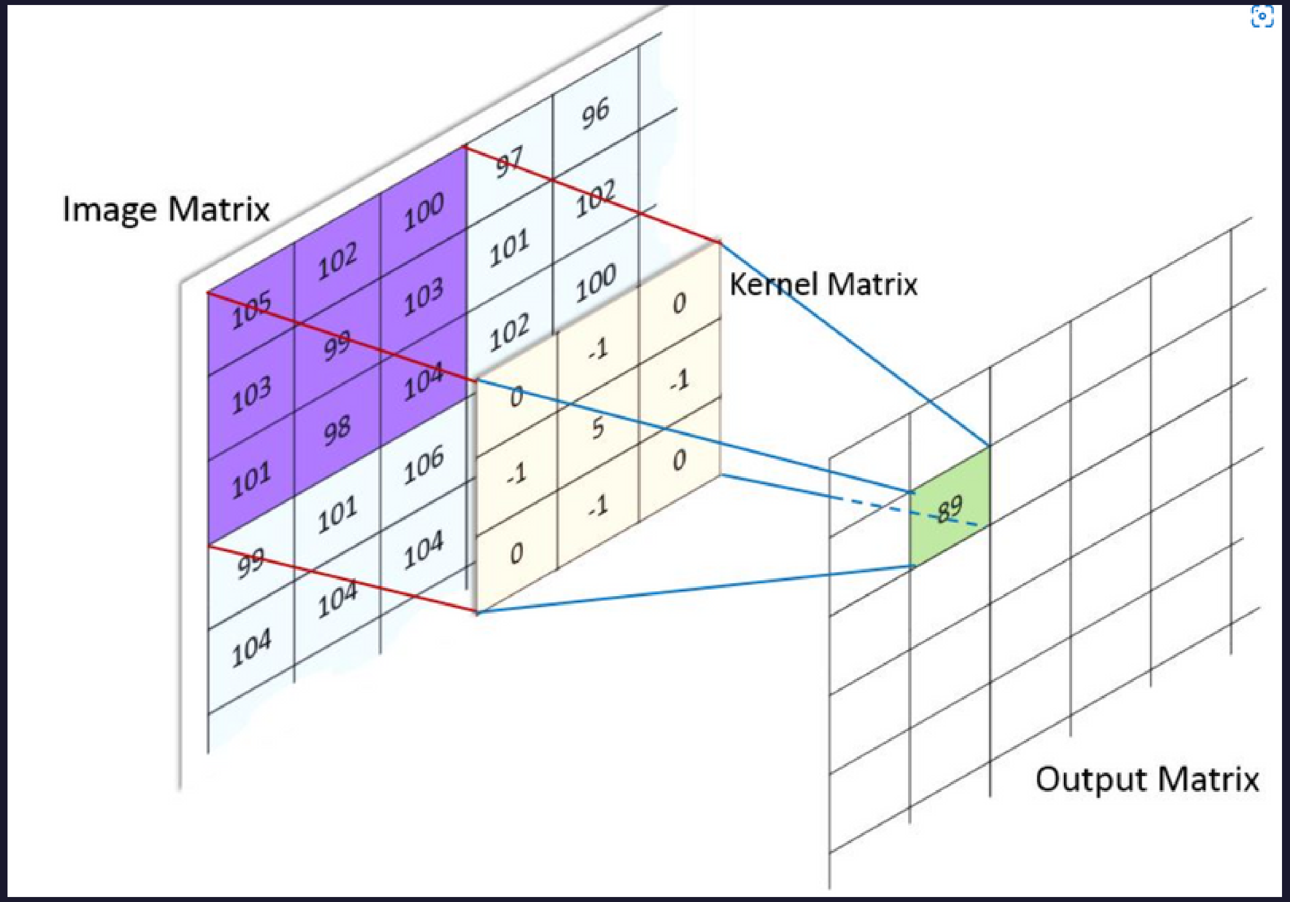
The basics of image processing

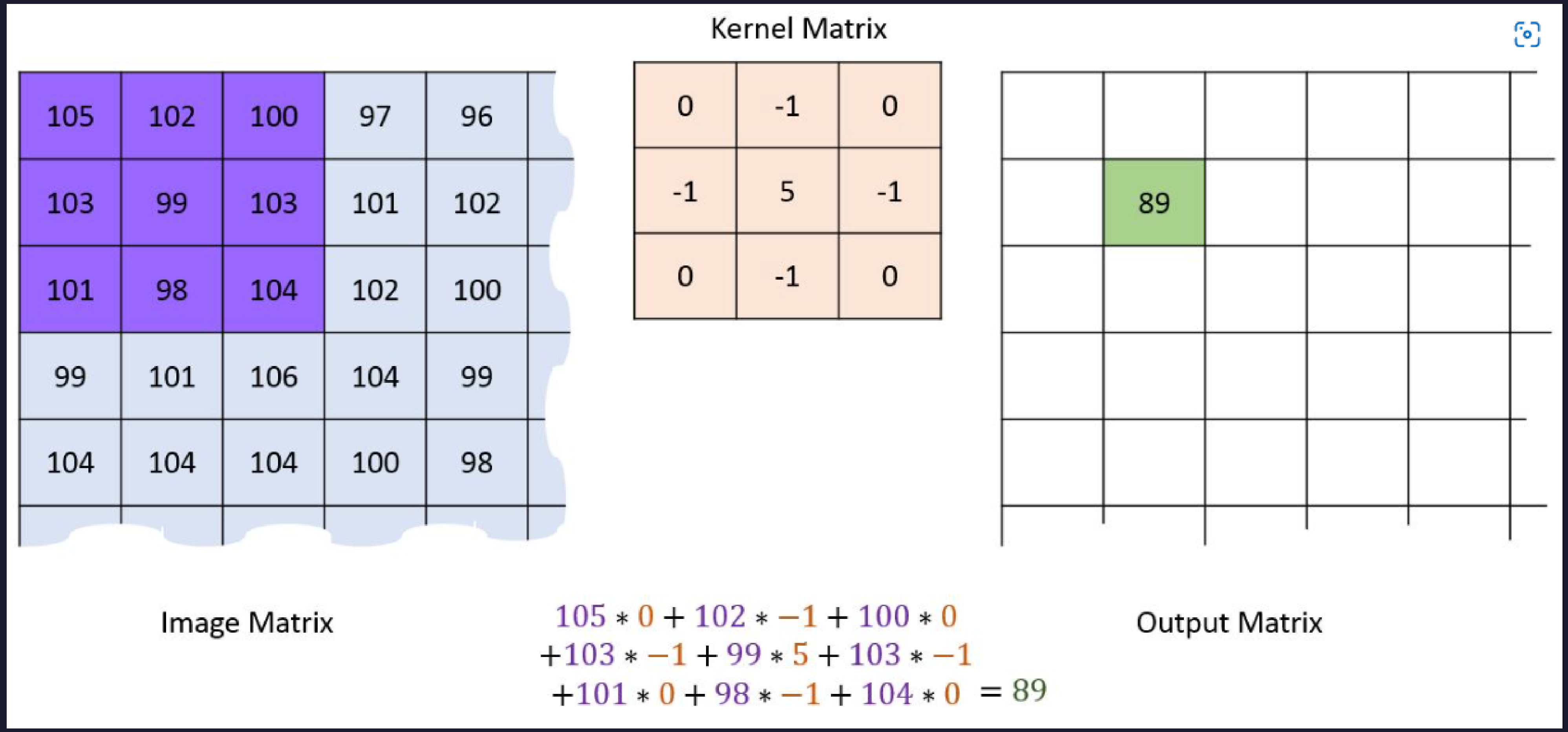
2D CONVOLUTION

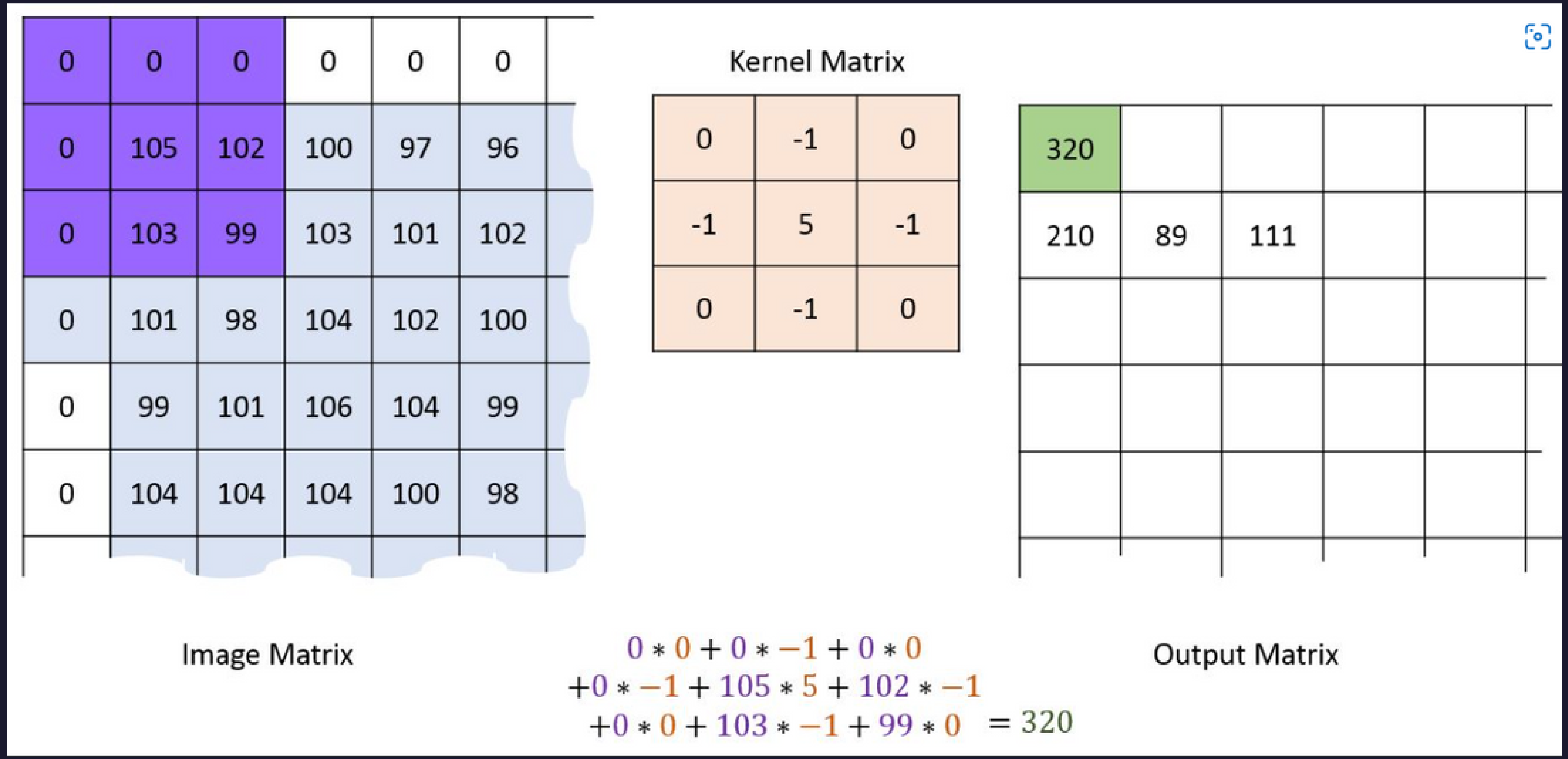
In continuation of the 1D convolution, the 2D convolution works similarly.
2D convolution of the image matrix with the filter matrix consists of the following steps:
1. Flip the filter both horizontally and vertically.
2. Put the first element of the kernel at every pixel of the image (element of the image matrix). Then each kernel element will stand on top of an element of the image matrix.
3. To calculate the value of convolution output at pixel (2,2), center the kernel at that position on the image matrix.
4. Multiply each element of the kernel with its corresponding element of the image matrix (the one which is overlapped with it)
5. Sum up all product outputs and put the result at the same position in the output matrix as the center of the kernel in the image matrix.
6. For the pixels on the border of the image matrix, some kernel elements will stand out from the image matrix and therefore do not have any corresponding element from the image matrix. In this case, we may apply 0-padding to the input matrix (based on the kernel size, we might need one or more pixels padding; in our example, we just need 1-pixel padding).

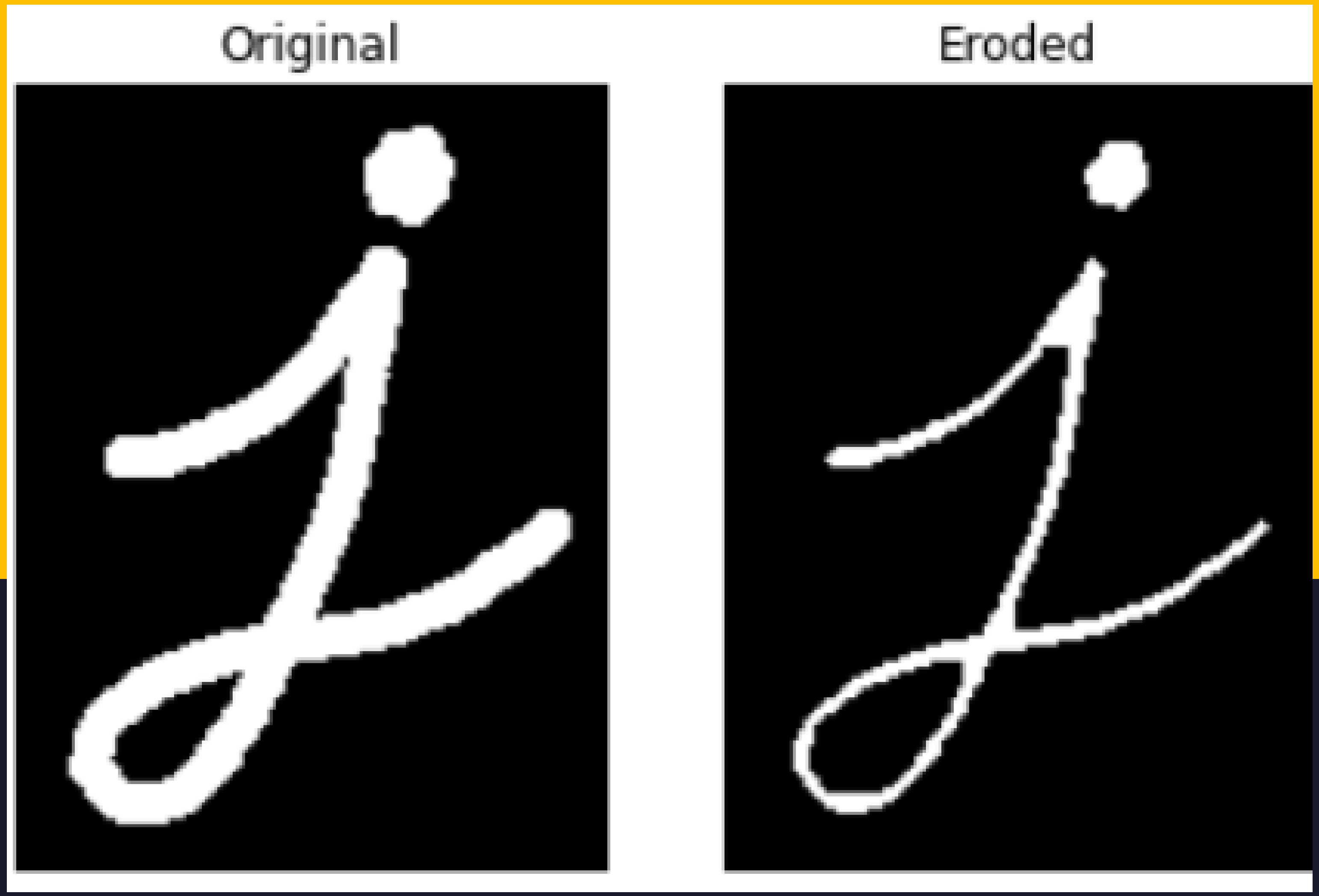
If, after the convolution operation, an output element happens to exceed 255 or is less than 0, it is thresholded to 255 or 0, respectively.

2D CONVOLUTION

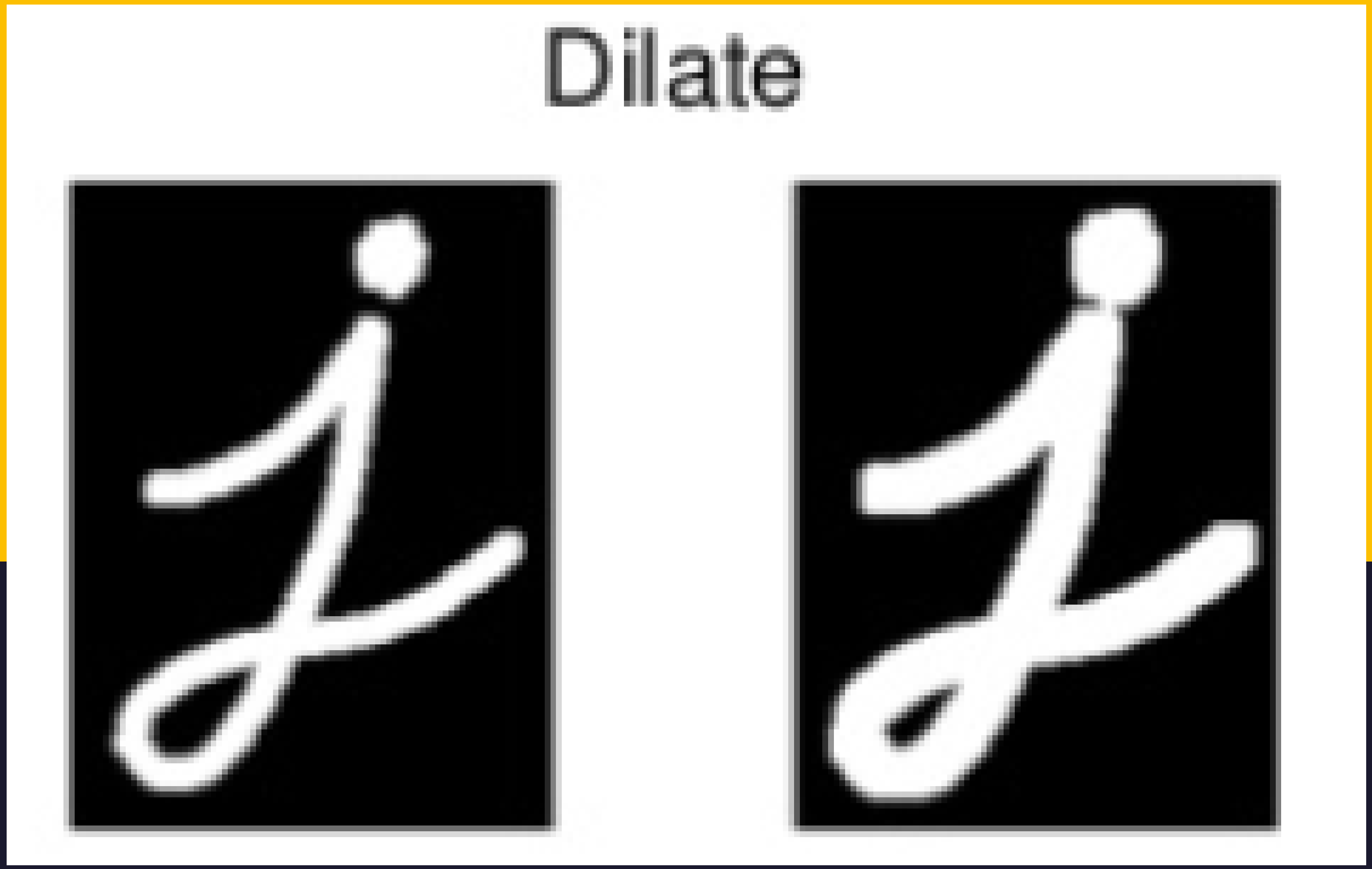








Erosion



Dilation

MORPHOLOGICAL TRANSFORMATION

Morphological operations are a set of operations that process images based on shapes. They apply a structuring element to an input image and generate an output image.
The most basic morphological operations are two: Erosion and Dilation.

Erosion

Erodes the boundaries of the foreground object. It is used to diminish the features of an image.

Working of erosion:

A kernel of odd size (3, 5, or 7) is convolved with the image. Then a pixel in the original image (either 1 or 0) will be considered one only if all the pixels under the kernel are 1. Otherwise, it will be eroded (made to zero). All the pixels near the boundary will be discarded depending upon the kernel size. So the foreground object's thickness or size decreases or the image's white region decreases.

Dilation

It Increases the object area. It is also used to accentuate features.

Working of dilation:

A kernel of odd size (3, 5 or 7) is convolved with the image. Then a pixel element in the original image is '1' if atleast one pixel under the kernel is '1'. It increases the white region in the image or in other words, the size of the foreground object increases.