UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

DEEP LEARNING
CNN

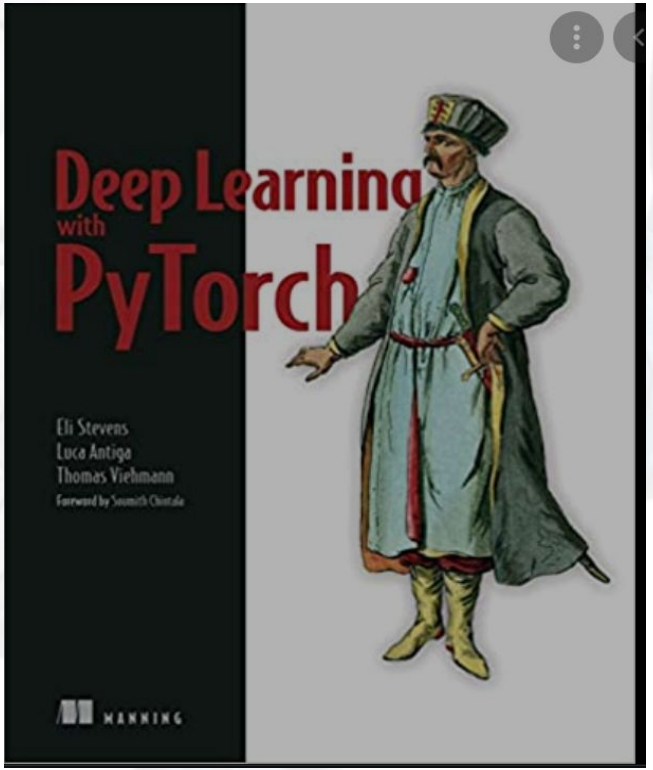
RESEARCHGROUP
IPRODAM3D

Cristian López Del Alamo
clopezd@utec.edu.pe

Noviembre, 2021

Schedule

1. Introduction
2. Kernel
3. Convolution
4. Pooling
5. CNN
6. Architecture
7. Applications



Fuente: Click

Faces with different emotions

Fuente: Click

1. Introduction

Image

0.1	0.8	1.3	.3	3.1	4.2

Feature vector

Wavelets, Harris, SIFT,
Histogram color, etc

Wavelets, Harris, SIFT,
Histogram color, etc

0.1	0.8	1.3	.3	3.1	4.2

v1: Feature Vector

d(v1, v2) < e

0.2	0.9	1.1	.2	3.5	4.0

v2: Feature Vector

Wavelets, Harris, SIFT,
Histogram color, etc

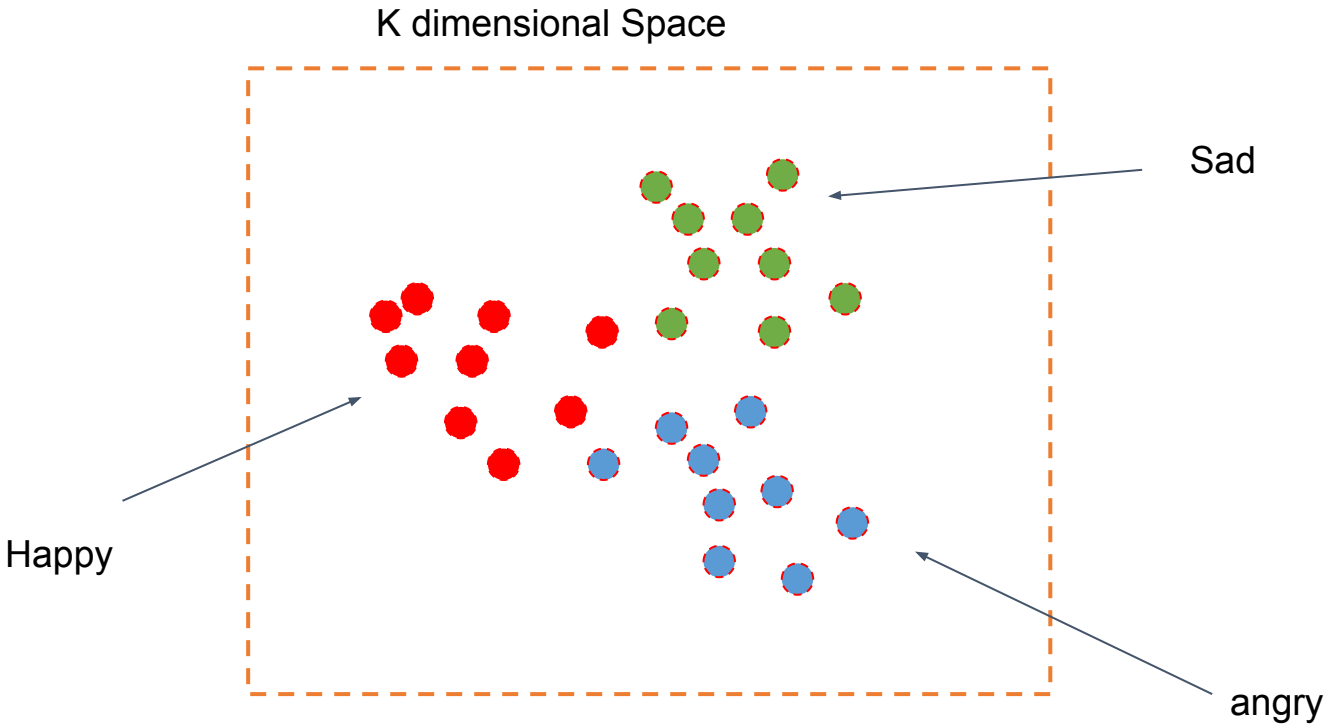
0.1	0.8	1.3	.3	3.1	4.2

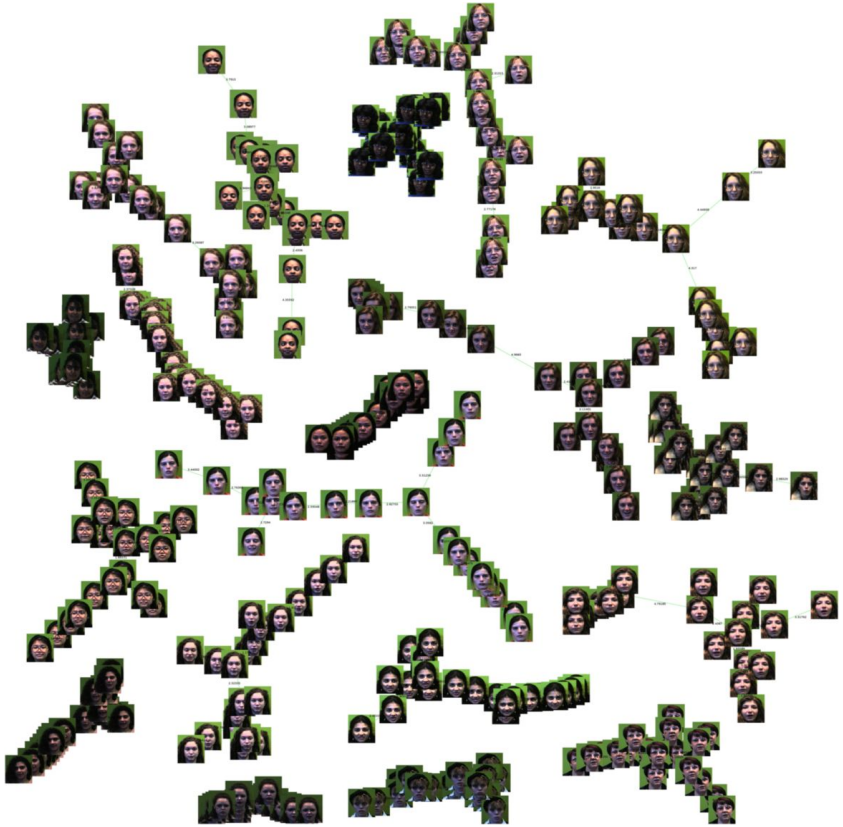
v1: Feature Vector

d(v1, v2) > e

2.2	3.9	1.6	2.2	6.5	1.0

v2: Feature Vector


K dimensional Space
Sad
Happy
angry



C. López, L. Arnaldo, L. Fuentes, W. Ramos, ***"Agrupamiento por similitud de imágenes mediante Árbol de Expansión Mínima y Soft Heap"***, XLI Latin American Computing Conference, pp. 1-7, 2013

X

MODEL

Neuronal Network

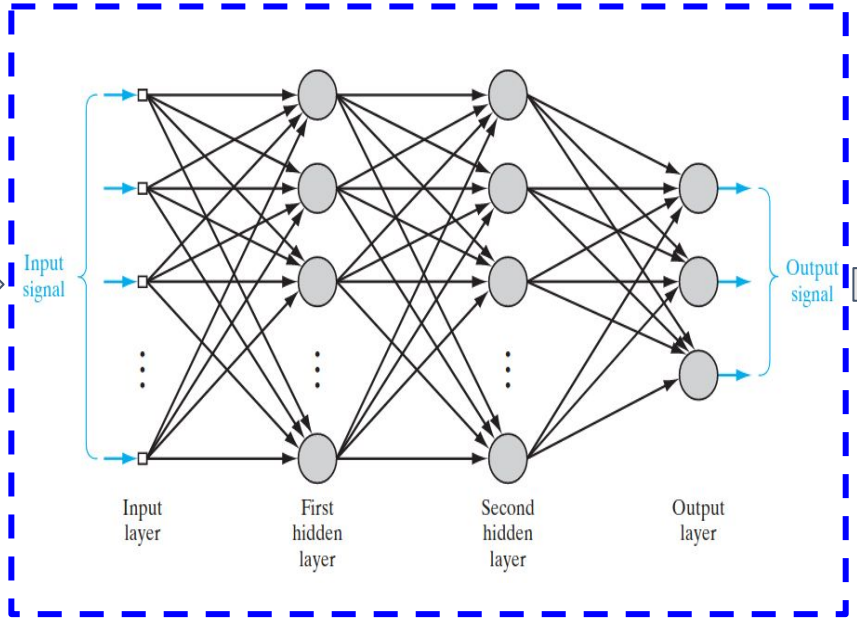
Happy

Y

Multilayer Perceptron

0.8
4.1
5.1
0.3
...
0.8

Feature Vector (SIFT, Hard Wavelets, Otsu, etc)



Model MLP

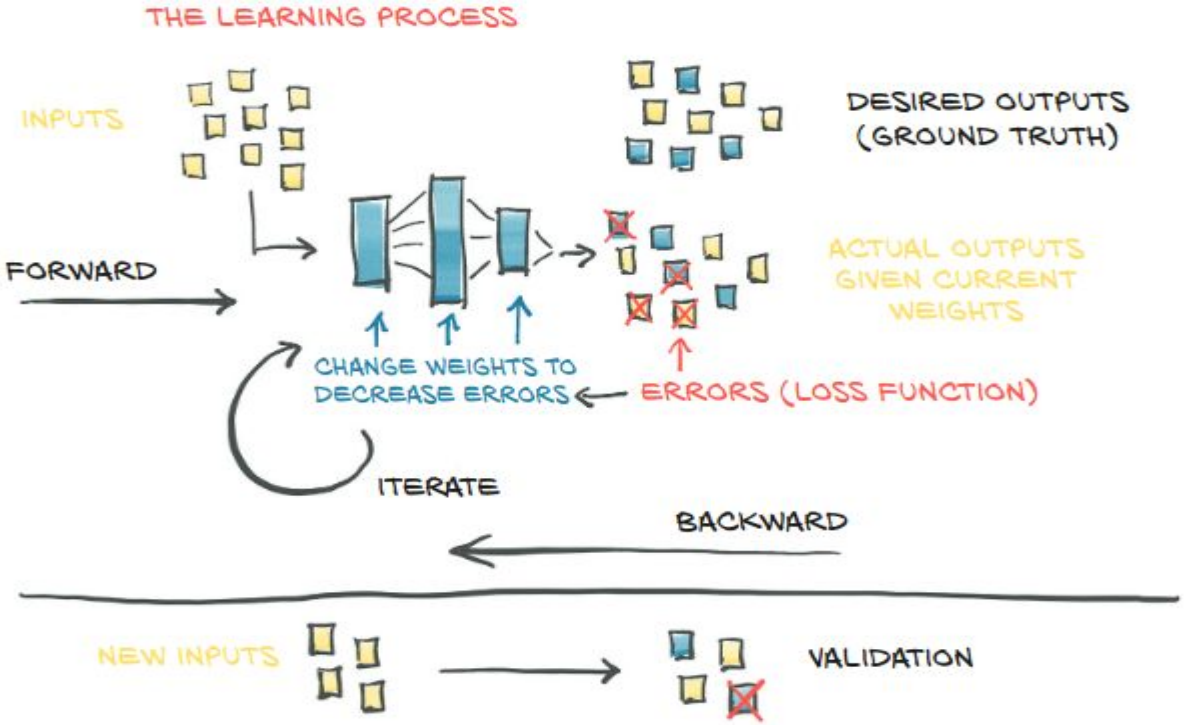
Loss Function

$\hat{y}$	y
0.8	1
0.1	0
0.1	0

Hot one encoded

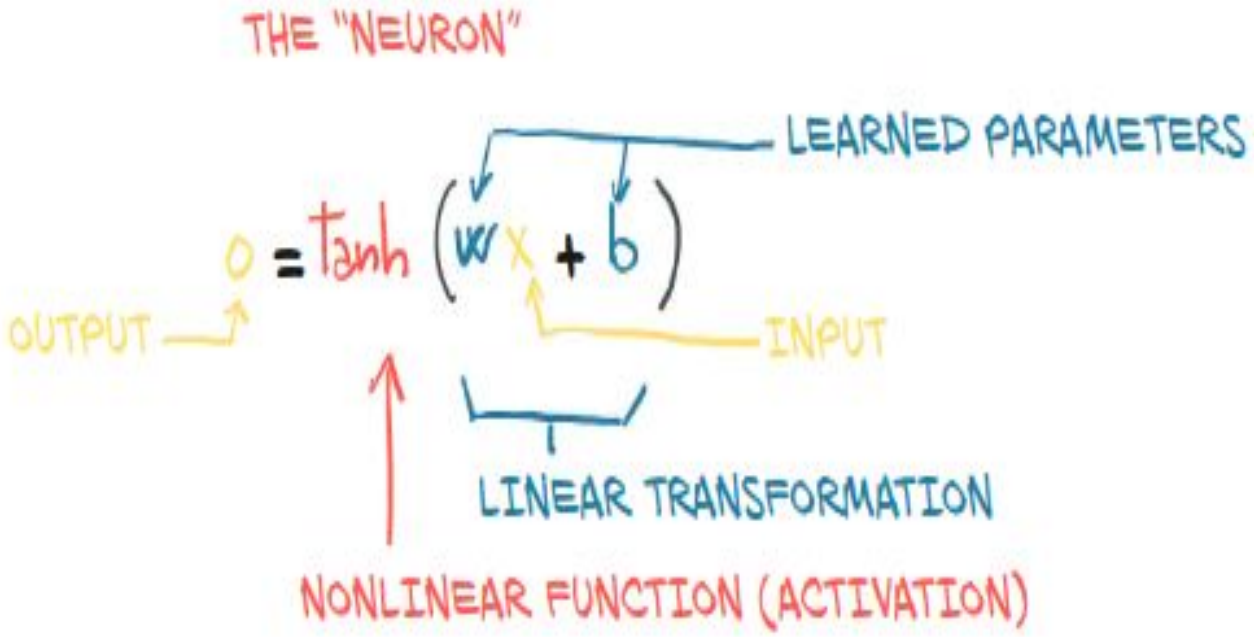
Happy

Multilayer Perceptron



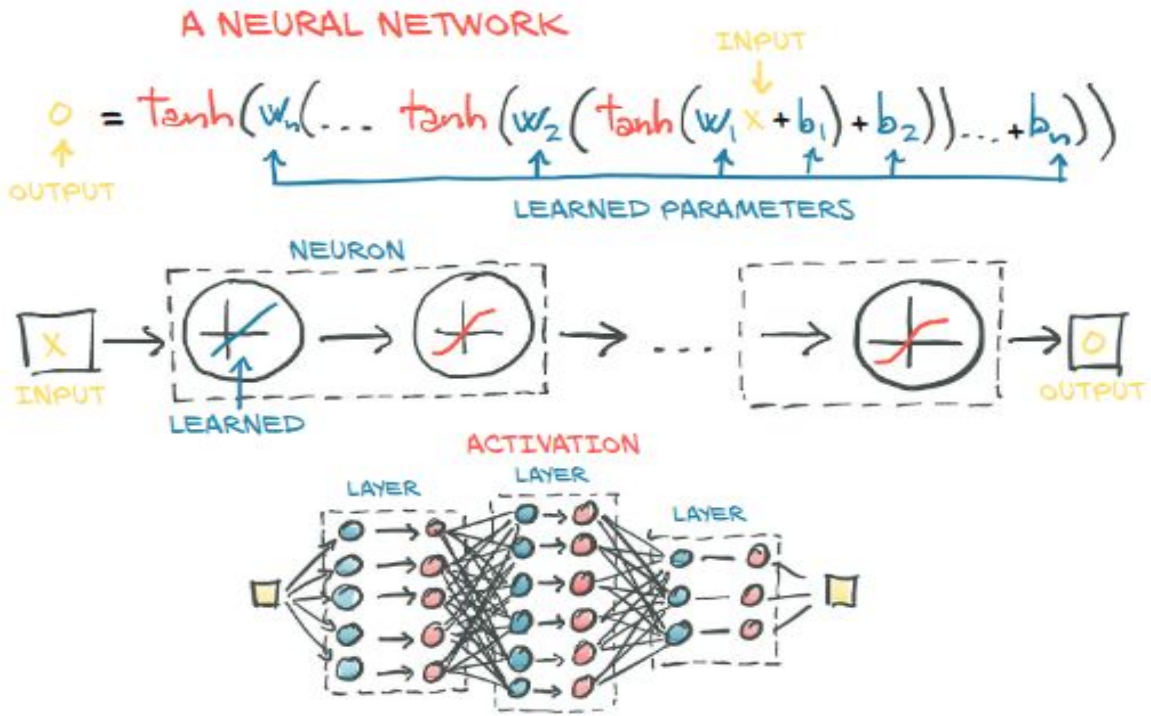
Fuente: Eli Stevens, et all, **Deep Learning with PyTorch**

Multilayer Perceptron



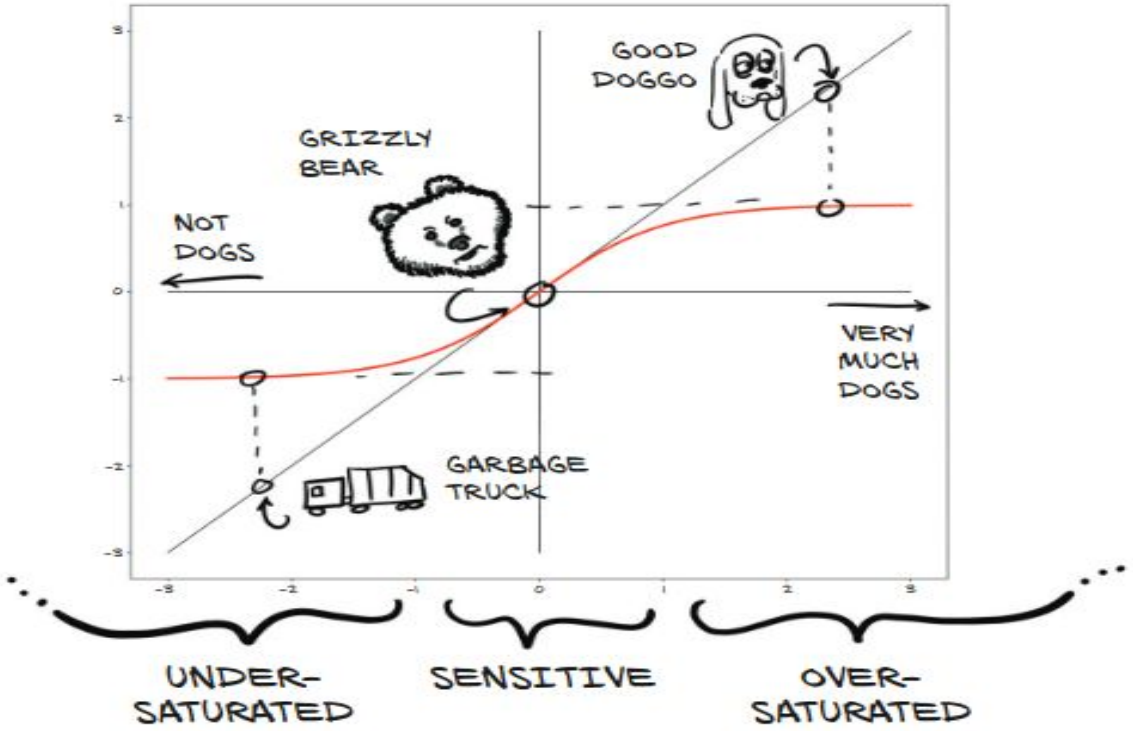
Fuente: Eli Stevens, et all, **Deep Learning with PyTorch**

Multilayer Perceptron



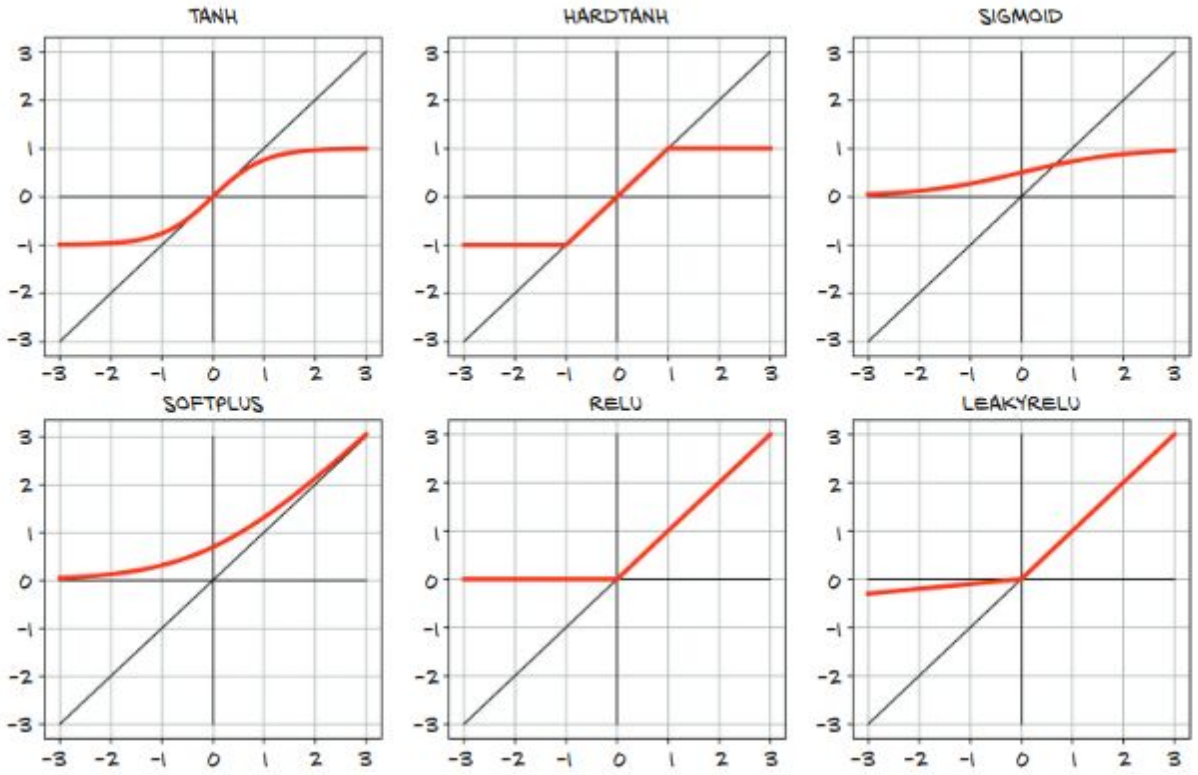
Fuente: Eli Stevens, et all, **Deep Learning with PyTorch**

Activation Function



Fuente: Eli Stevens, et all, **Deep Learning with PyTorch**

Activation Function

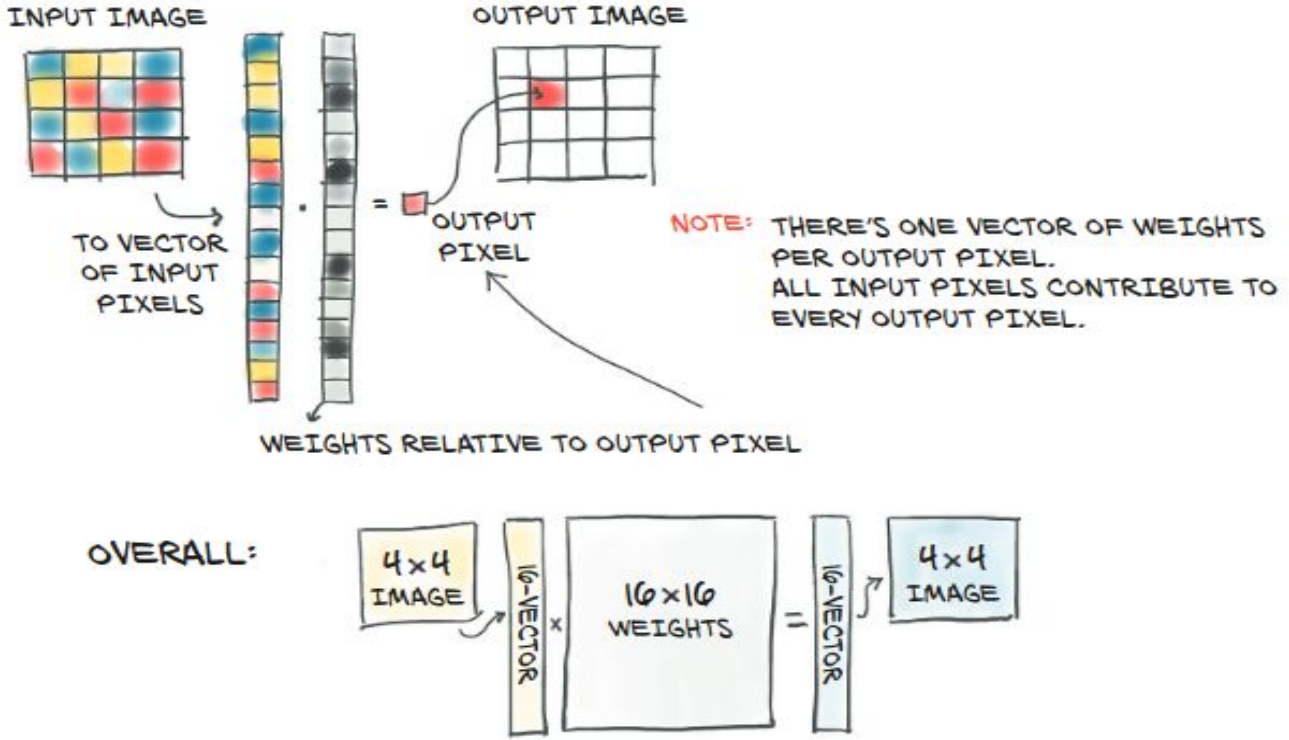


Fuente: Eli Stevens, et all, **Deep Learning with PyTorch**

Problems MLP

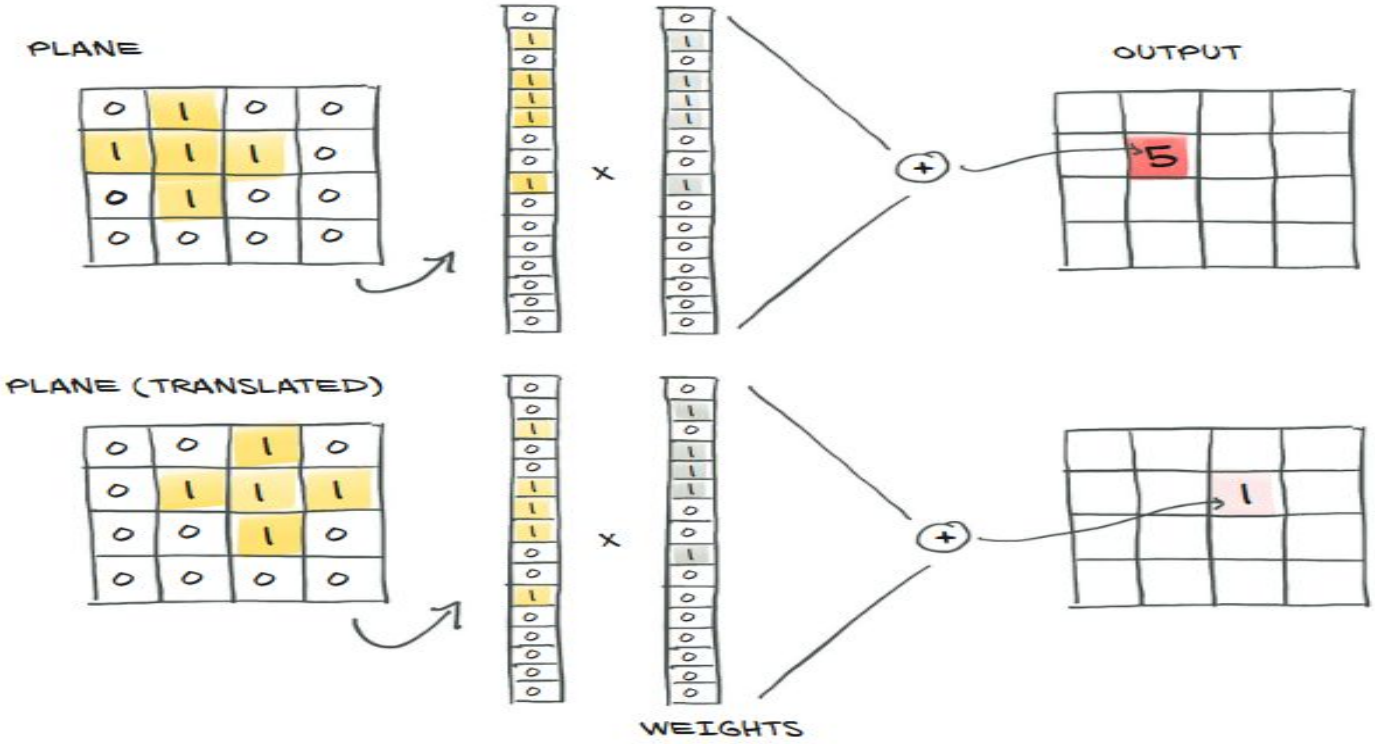
Fuente: Click

Problems with MLP



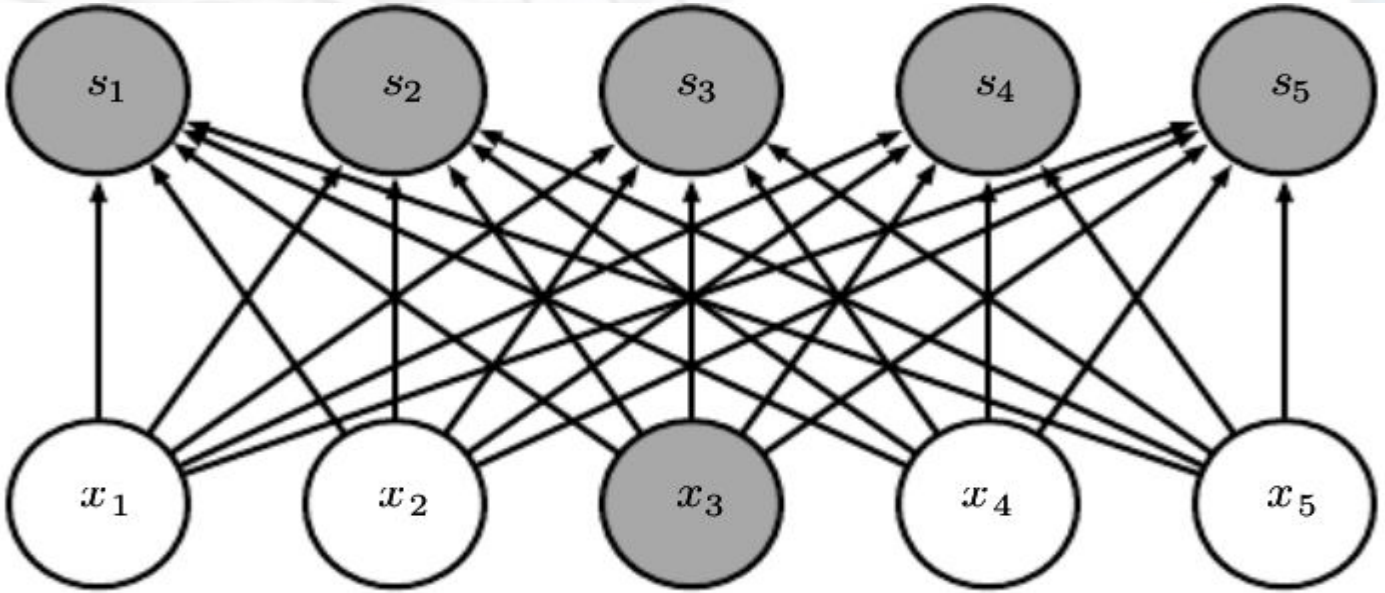
Fuente: Eli Stevens, et all, **Deep Learning with PyTorch**

Problems with MLP



Fuente: Eli Stevens, et all, **Deep Learning with PyTorch**

Problems with MLP



Ian Goodfellow , Yoshua Bengio, Aaron Courville; Deep Learning, 2016

CONVOLUTIONAL NEURAL NETWORK
CNN

Dr. Cristian López Del Alamo

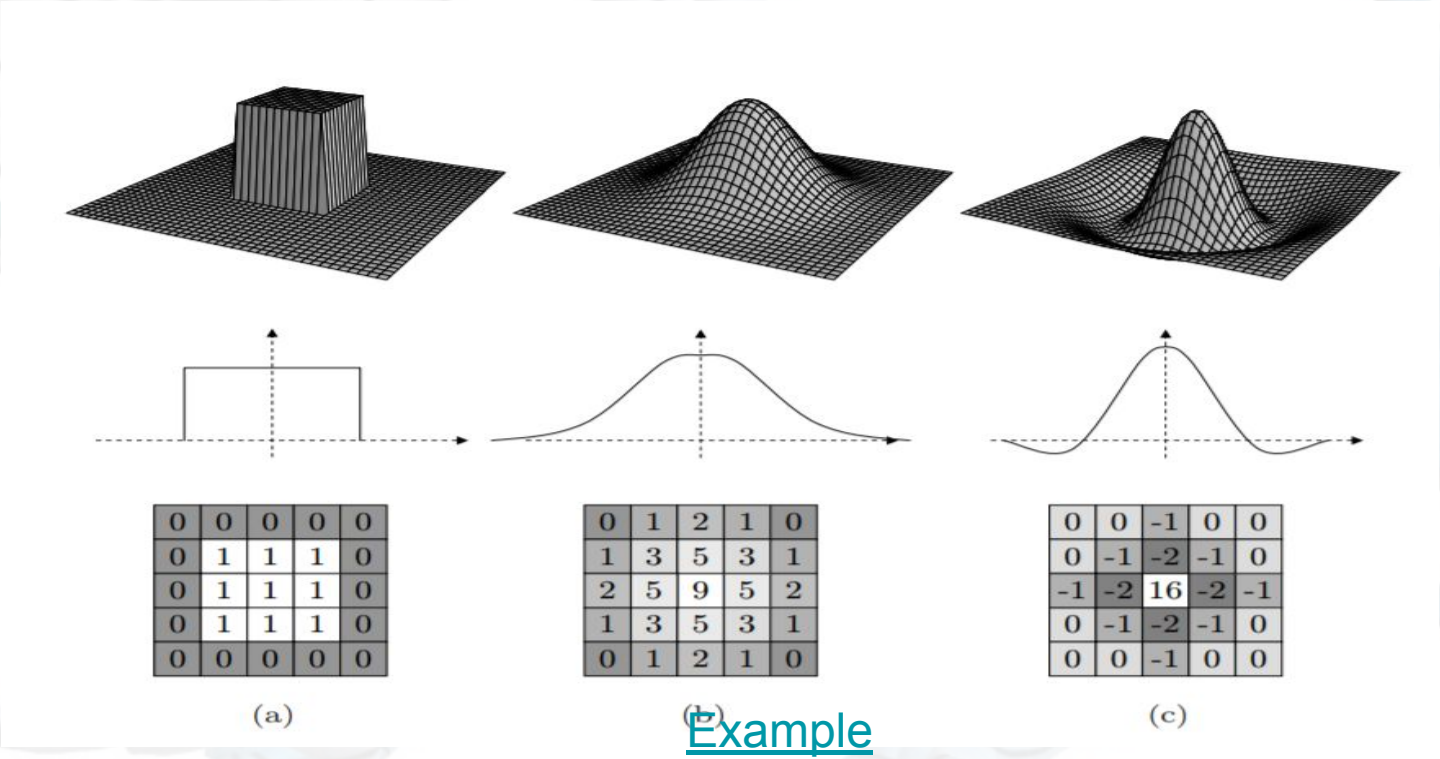
January 4, 2021

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

2. Kernel

Cristian López Del Alamo, IPRODAM3D, clopezd@utec.edu.pe

Kernel



0	0	0	0	0
0	1	1	1	0
0	1	1	1	0
0	1	1	1	0
0	0	0	0	0

(a)

0	1	2	1	0
1	3	5	3	1
2	5	9	5	2
1	3	5	3	1
0	1	2	1	0

(b)

0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

(c)

Example

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

3. Convolution

Convolution

0	0	0	0	0	0	0
0	60	113	56	139	85	0
0	73	121	54	84	128	0
0	131	99	70	129	127	0
0	80	57	115	69	134	0
0	104	126	123	95	130	0
0	0	0	0	0	0	0

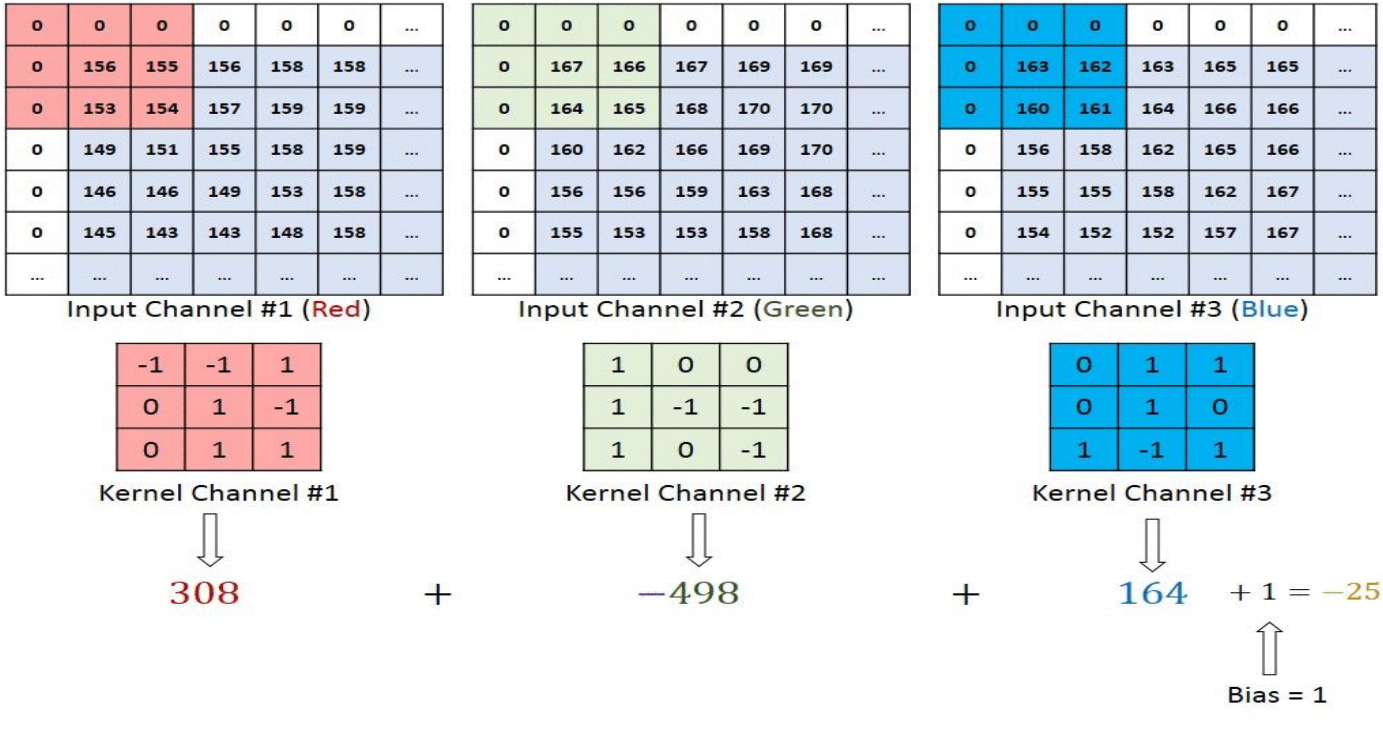
Kernel

0	-1	0
-1	5	-1
0	-1	0

114				

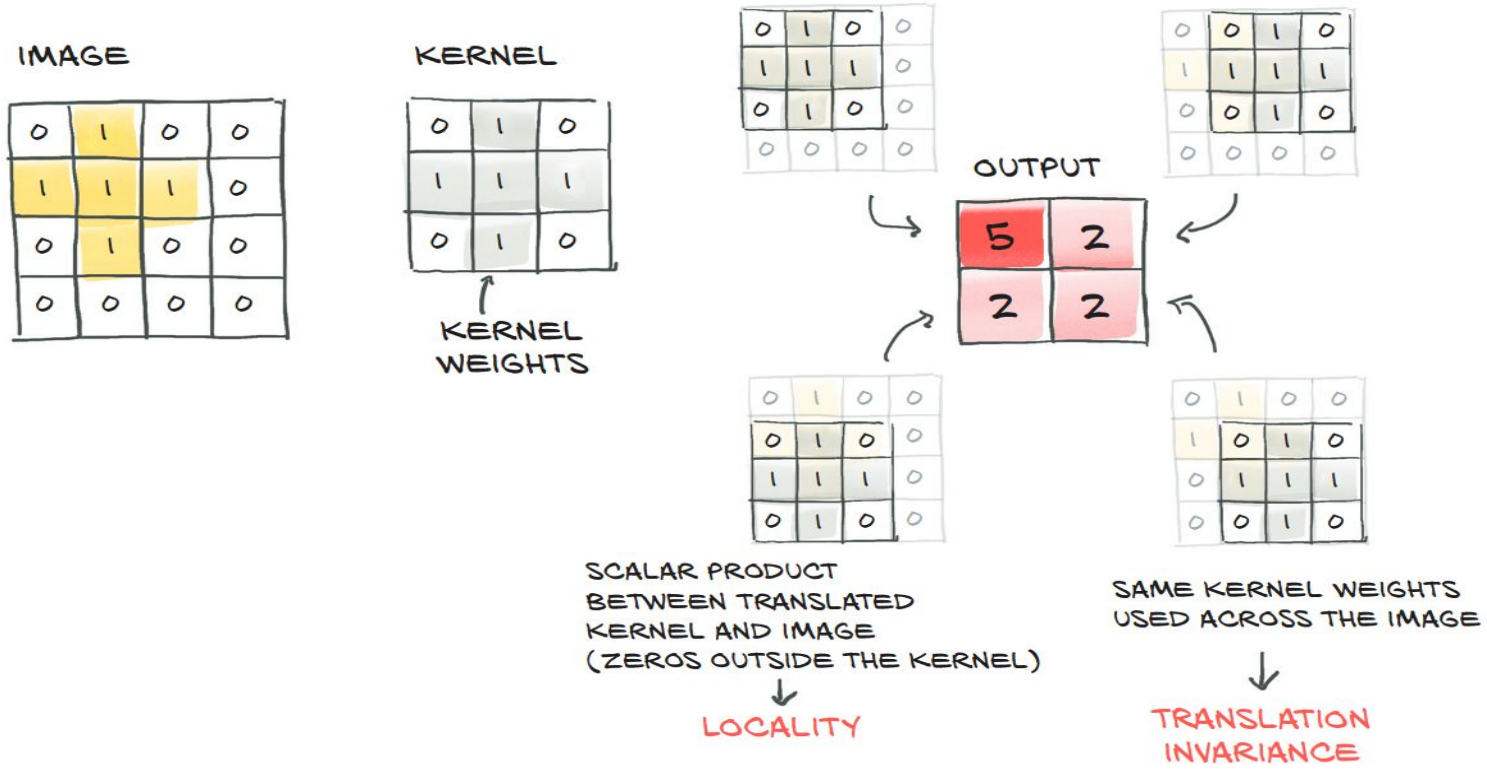
Fuente: Click

Convolution



Fuente: Click

Convolution



Fuente: Eli Stevens, et all, **Deep Learning with PyTorch**

Convolution layer : Stride

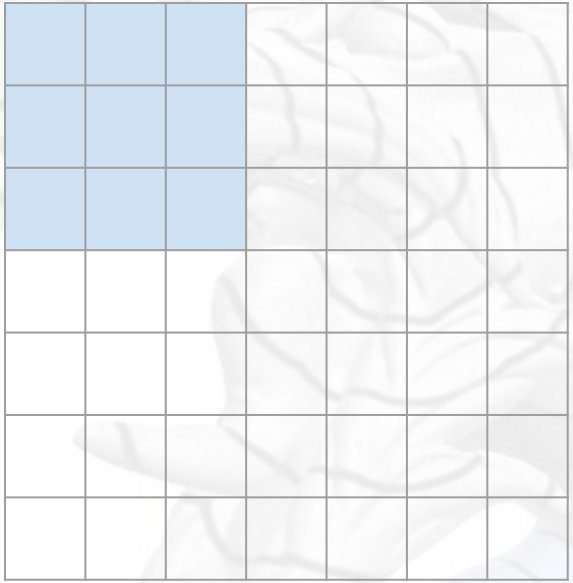


Image size = (7x7)
Filter size = (3x3)
padding = 0
stride = 1

Convolution layer : Stride

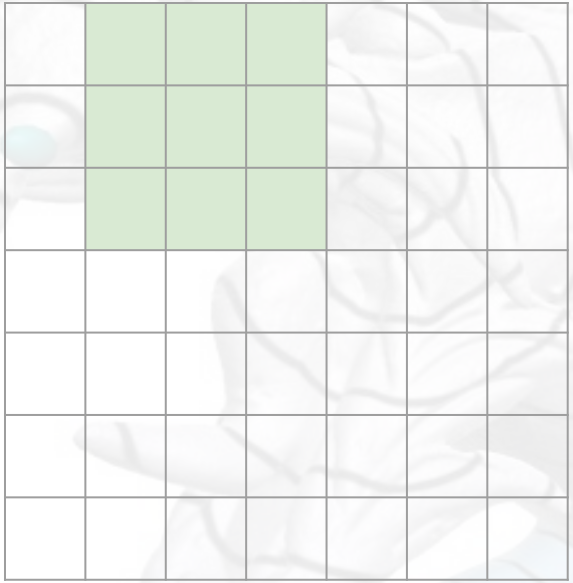


Image size = (7x7)
Filter size = (3x3)
padding = 0
stride = 1

Convolution layer : Stride

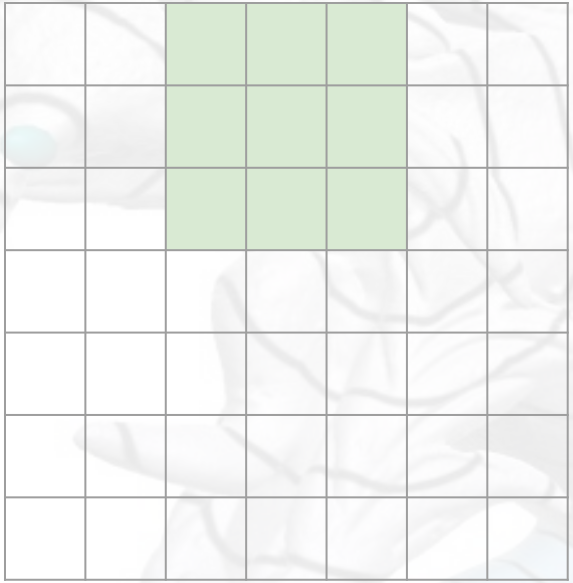


Image size = (7x7)
Filter size = (3x3)
padding = 0
stride = 1

Convolution layer : Stride

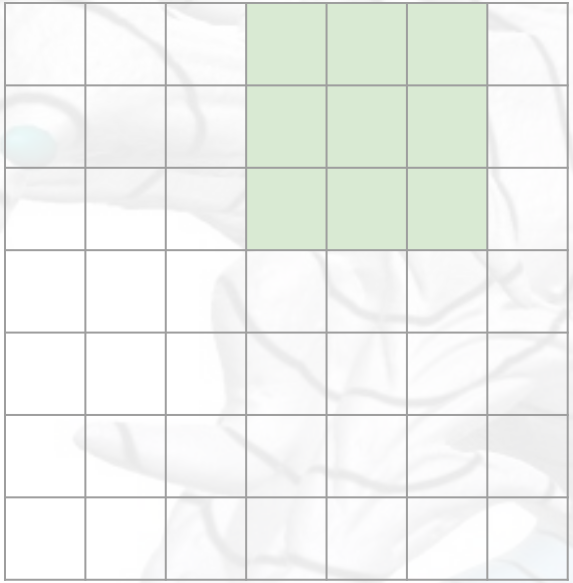


Image size = (7x7)
Filter size = (3x3)
padding = 0
stride = 1

Convolution layer : Stride

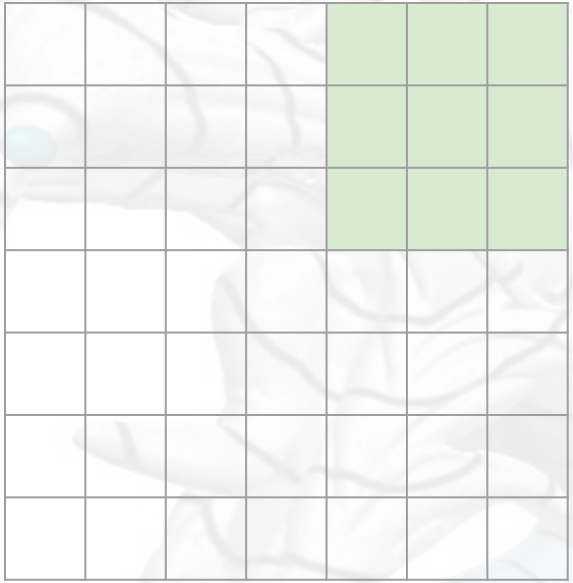


Image size = (7x7)
Filter size = (3x3)
padding = 0
stride = 1

output size = ?

Convolution layer : Stride

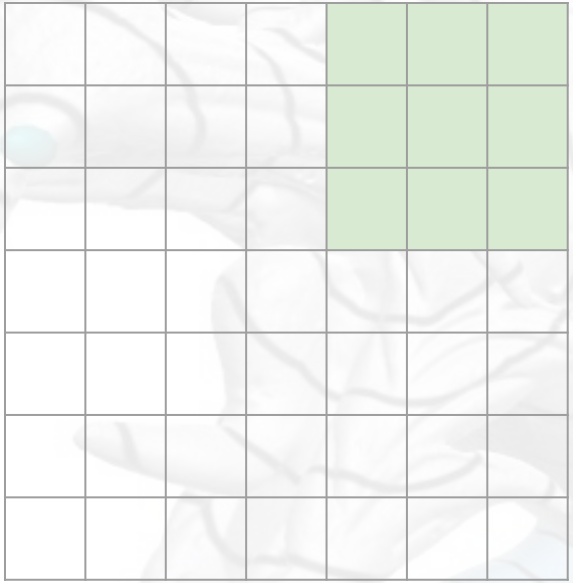


Image size = (7x7)
Filter size = (3x3)
padding = 0
stride = 1

output size = 5x5

Convolution layer : Stride

Image size = (7x7)
Filter size = (3x3)
padding = 0
stride = 2

Convolution layer : Stride

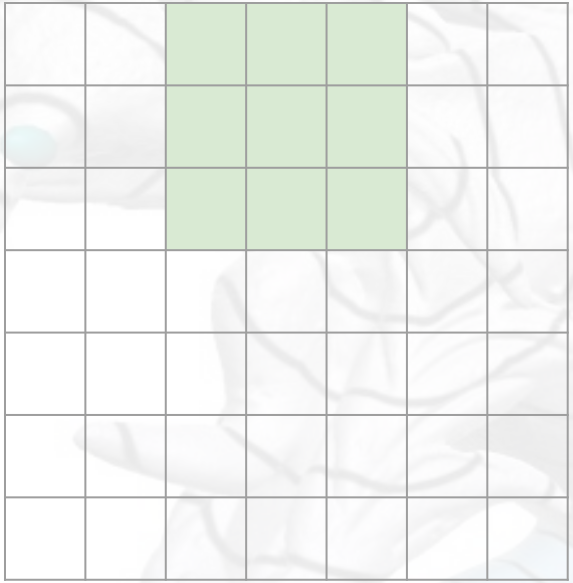


Image size = (7x7)
Filter size = (3x3)
padding = 0
stride = 2

Convolution layer : Stride

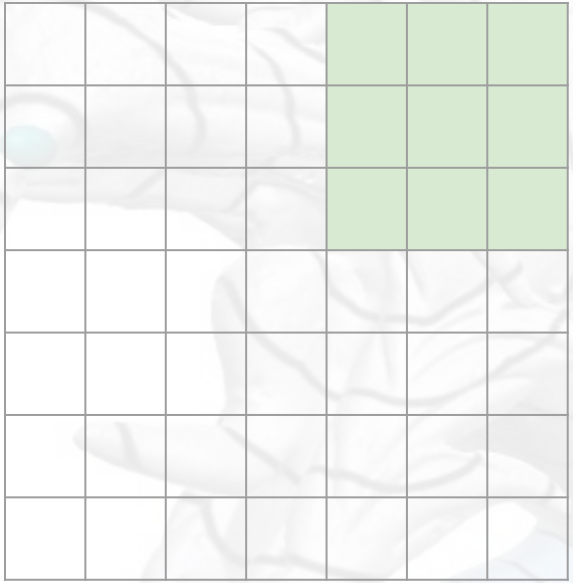
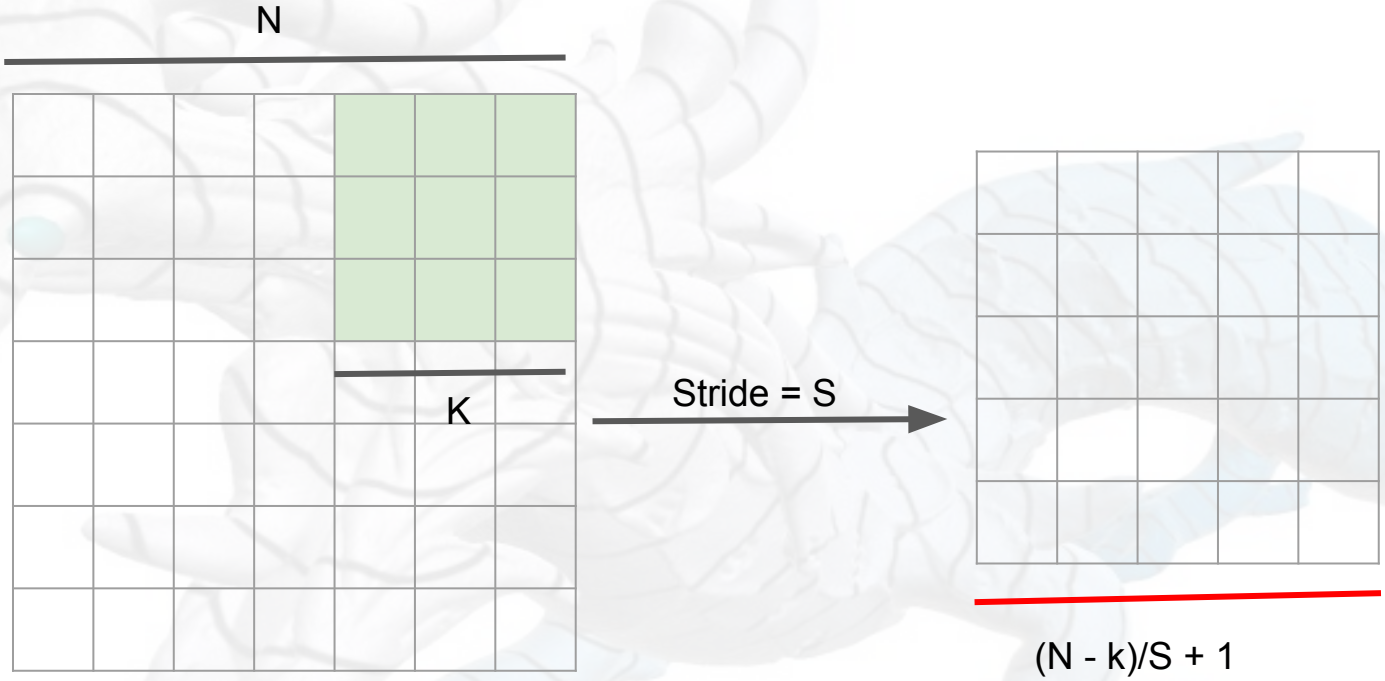


Image size = (7x7)
Filter size = (3x3)
padding = 0
stride = 2

output size = 3x3

What happen if the kernel size is 3x3 ?

Convolution layer : Stride



Convolution layer : Padding

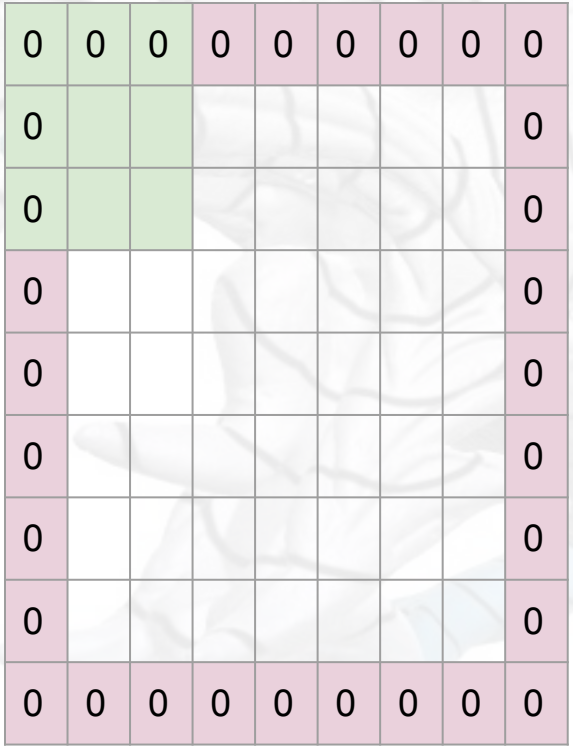


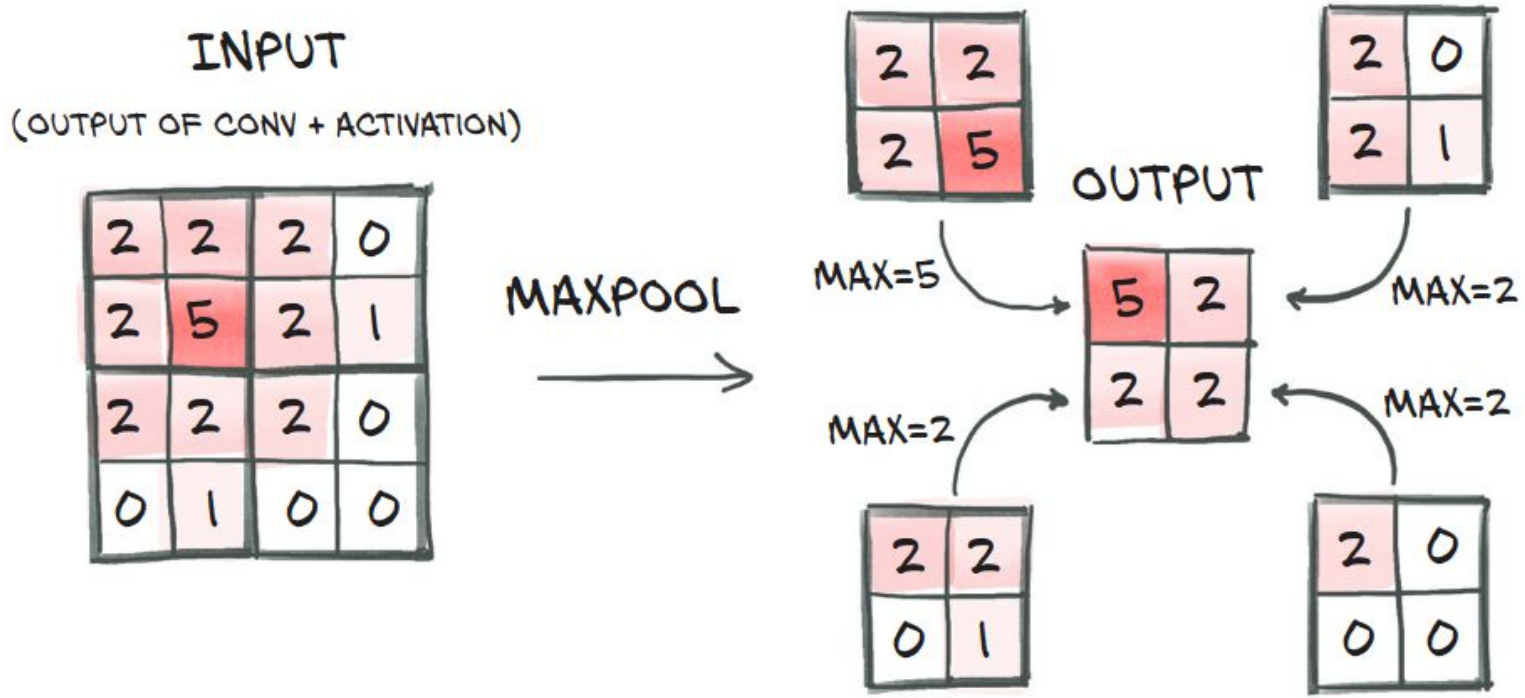
Image size = (7x7)
Filter size = (3x3)
padding = 1
stride = 1

What the output size?

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

4. Pooling

Pooling



Fuente: Eli Stevens, et all, **Deep Learning with PyTorch**

Pooling layer

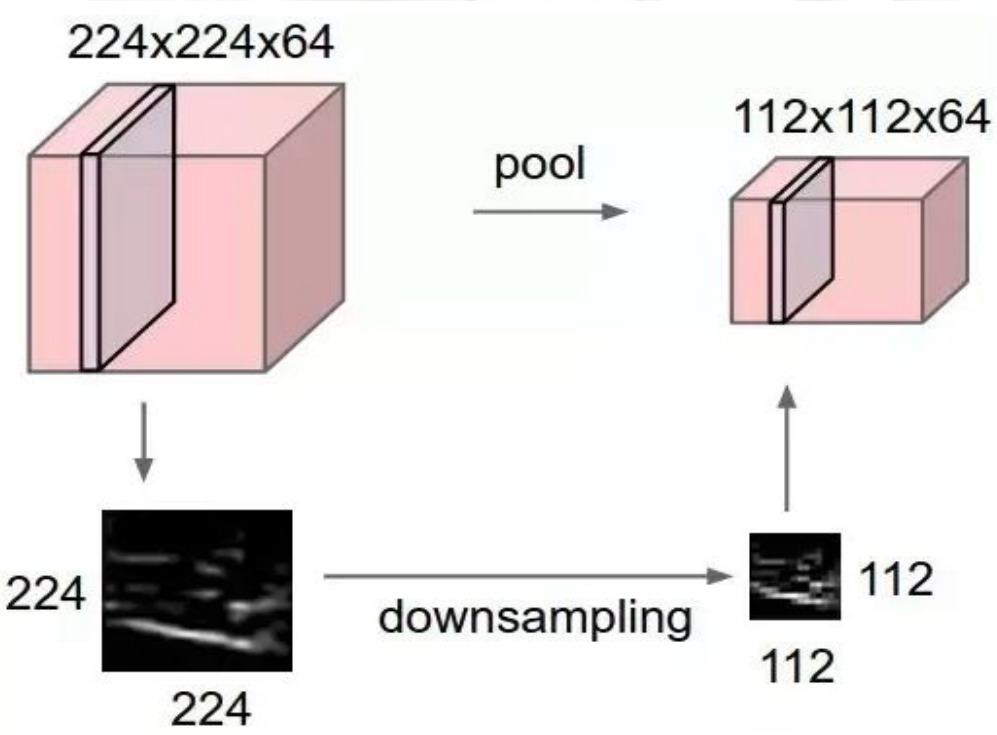


image source: click

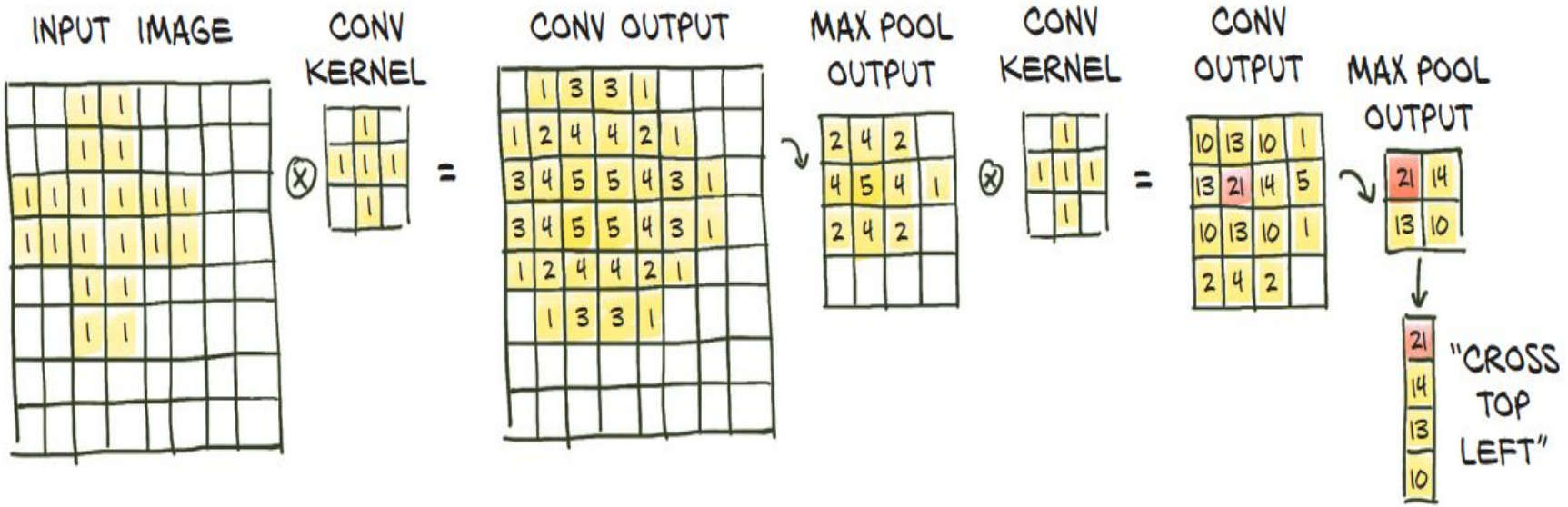
- Invariant to small translation ot th input
- Can handle inputs of variable size and return outputs of the same size

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

5. CNN

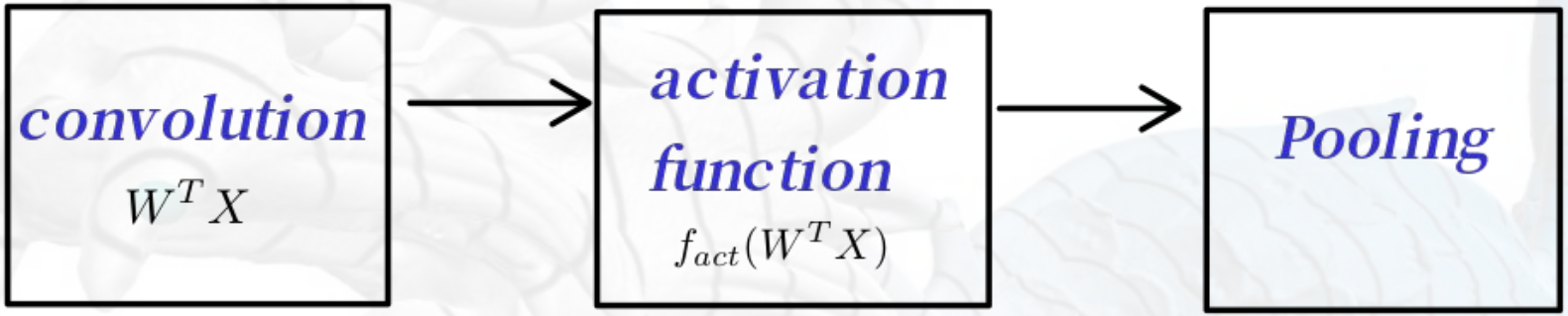
Cristian López Del Alamo, IPRODAM3D, clopezd@utec.edu.pe

Convolution and Pooling



Fuente: Eli Stevens, et all, **Deep Learning with PyTorch**

Typical layer in CNN



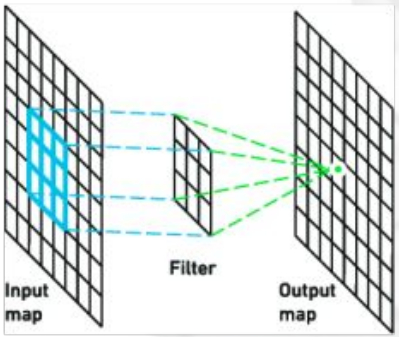


image source: click

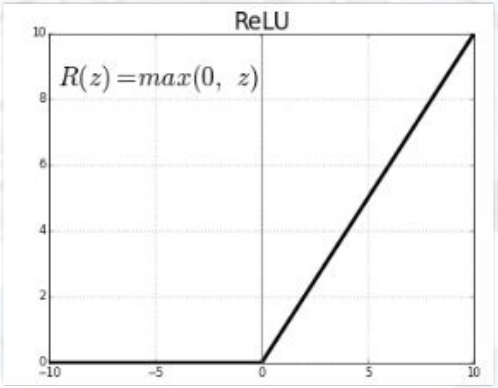


image source: click

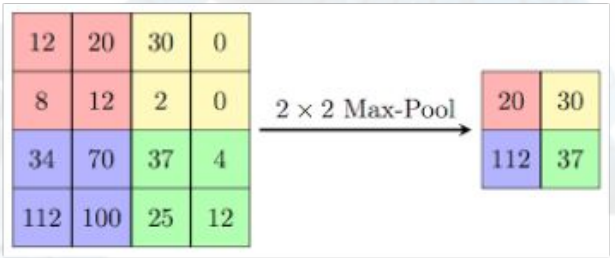
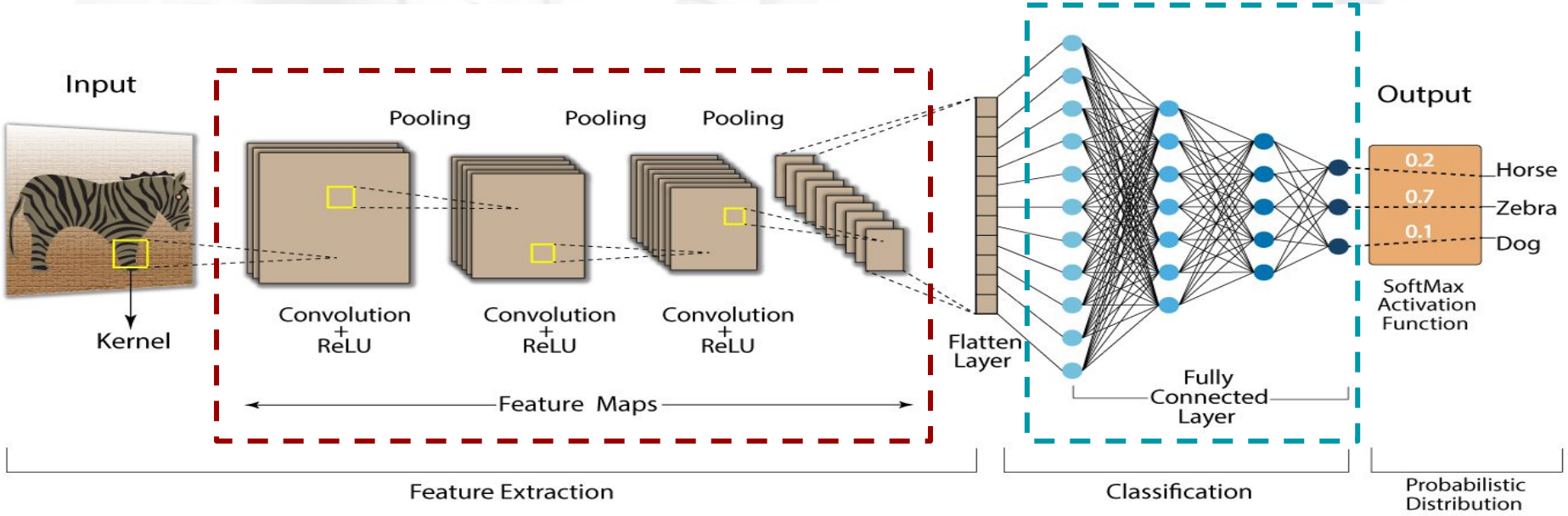


image source: click

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

6. Architecture

Architecture



$$f\left(\quad\right) = \hat{y} \qquad |y - \hat{y}|_p^p < \epsilon$$

CNN Architecture

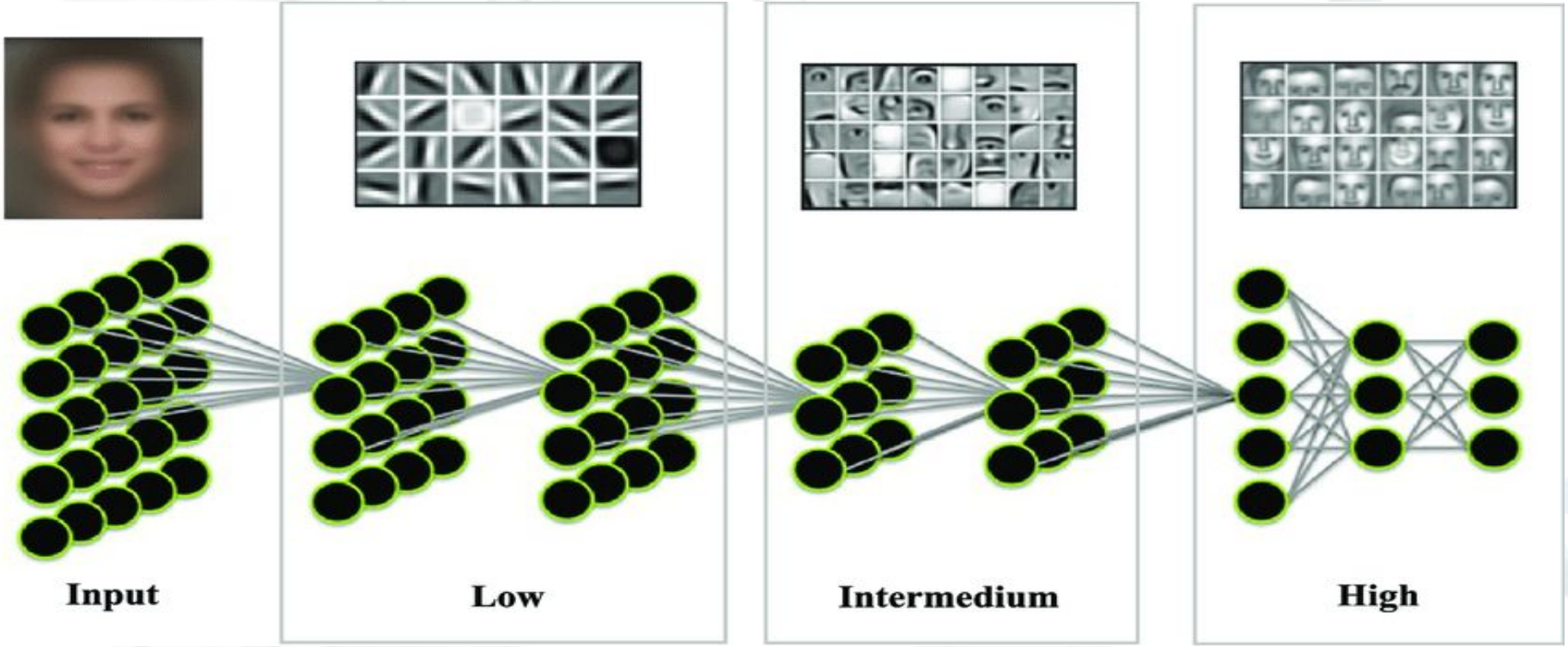
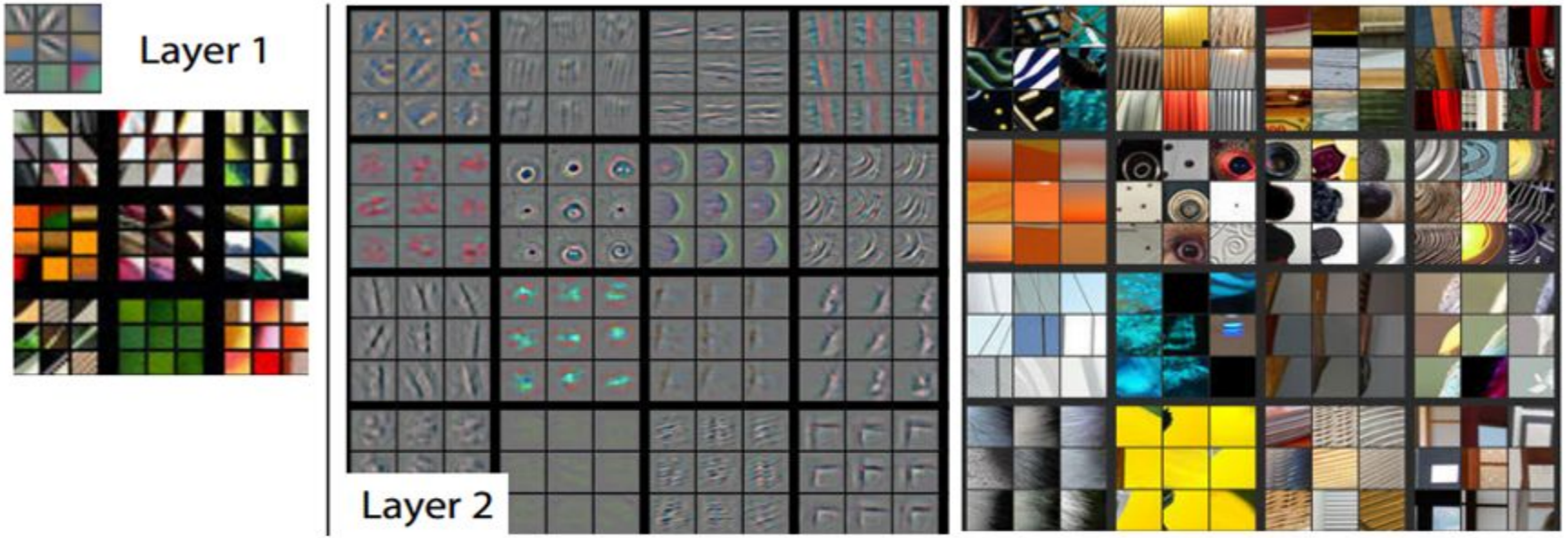


image source: click

CNN Architecture



Visualizations of Layer 1 and 2. Each layer illustrates 2 pictures, one which shows the filters themselves and one that shows what part of the image are most strongly activated by the given filter. For example, in the space labled Layer 2, we have representations of the 16 different filters (on the left)

image source: click

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

7. Applications

Applications

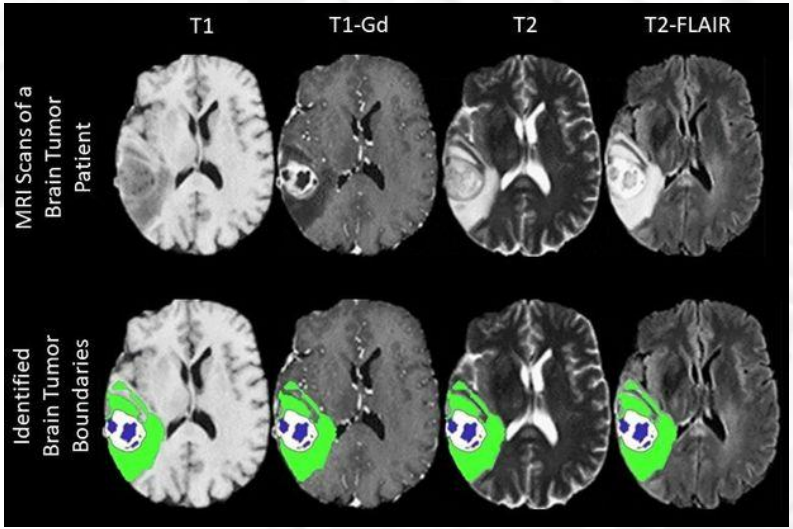
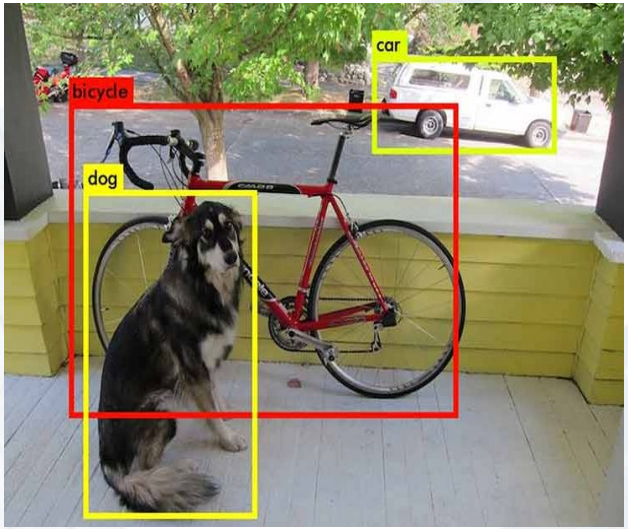


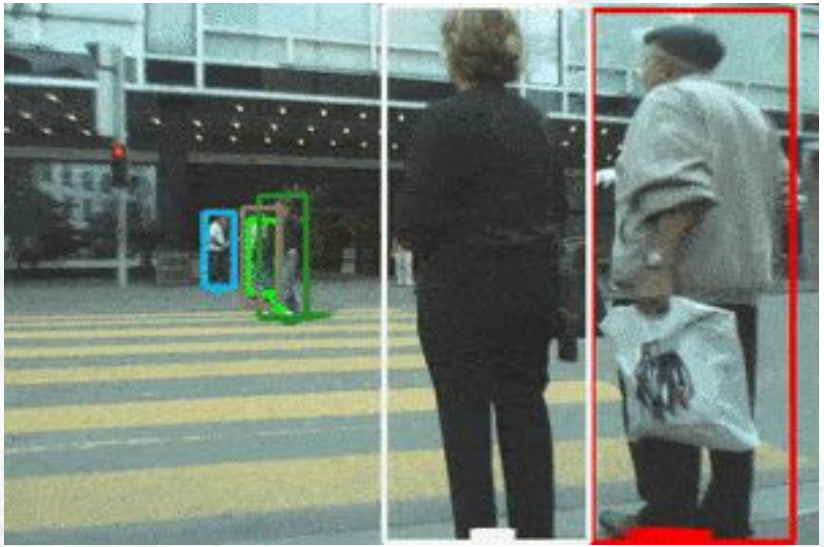
Image Segmentation



Object Detection and Segmentation

Applications

Image Segmentation



Video Tracking

Applications

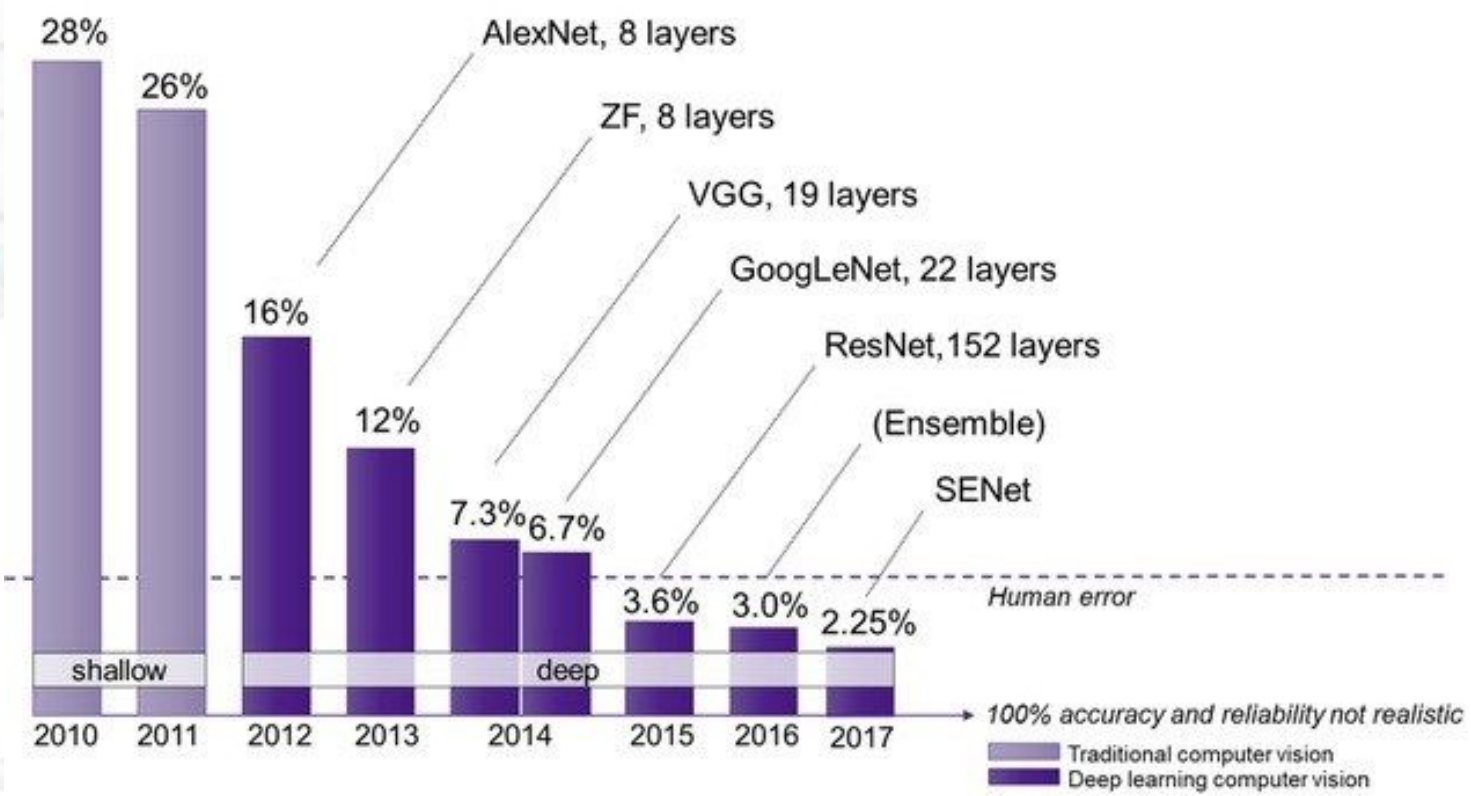
Denoising

Super resolution

Applications

Inpainting

INTRODUCTION

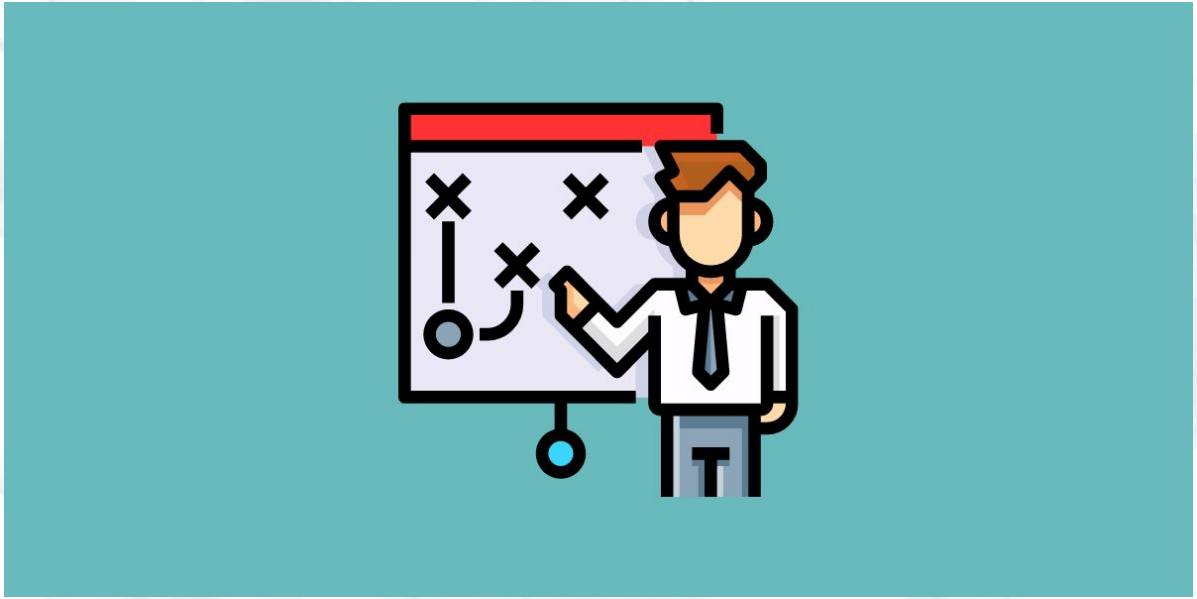


ImageNet Large Scale Visual Recognition Challenge results show that deep learning is surpassing human levels of accuracy.
Source: click

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Pythor Example

CNN Pythorh



Colab Example

Extra Information

CNN Backpropagation derivatives