

Aula 1

Introducción a las redes neuronales convolucionales CNN

Impacto y crecimiento de vision computacional

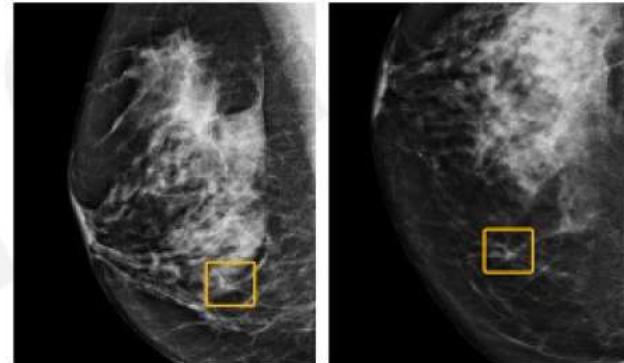
Robotics



Accessibility



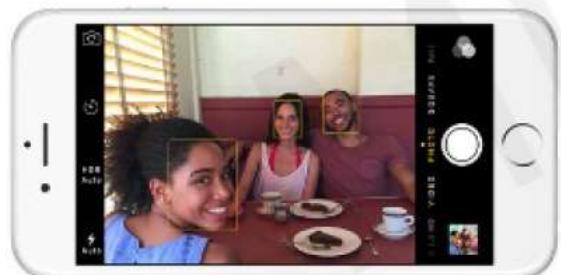
Biology & Medicine



Autonomous driving

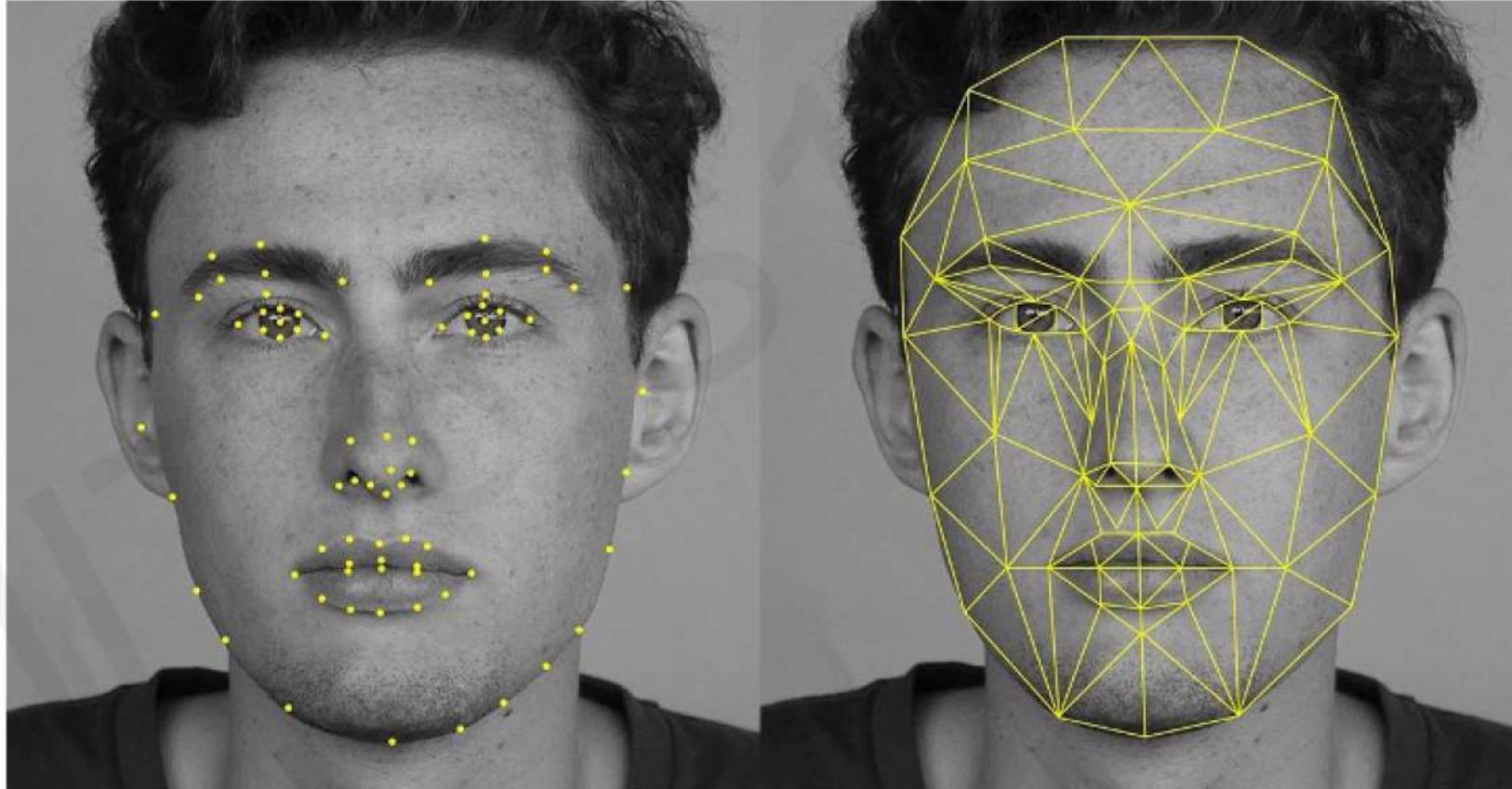
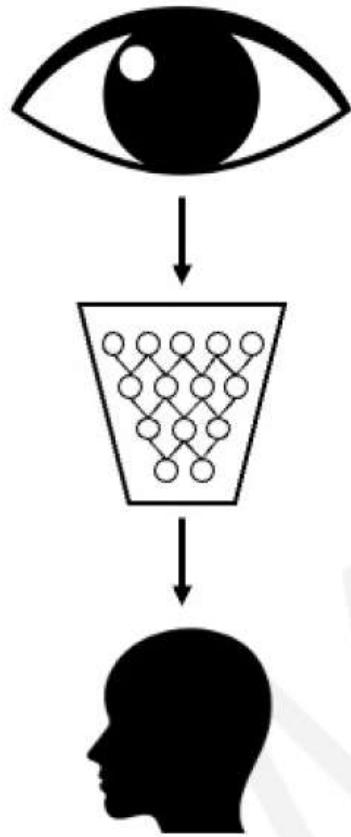


Mobile computing



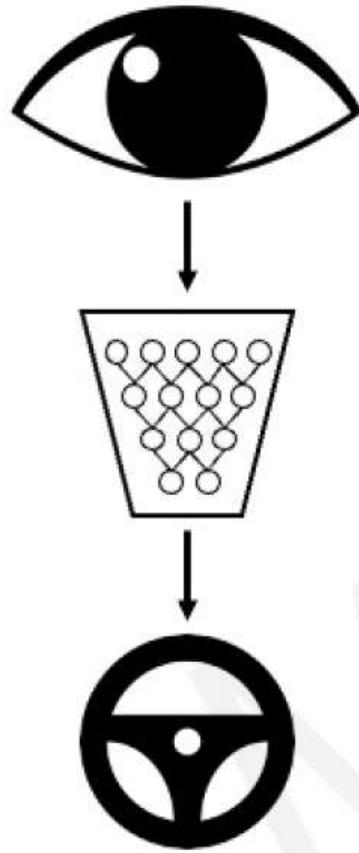
Fuente: <http://introtodeeplearning.com/>

Detección y reconocimiento facial



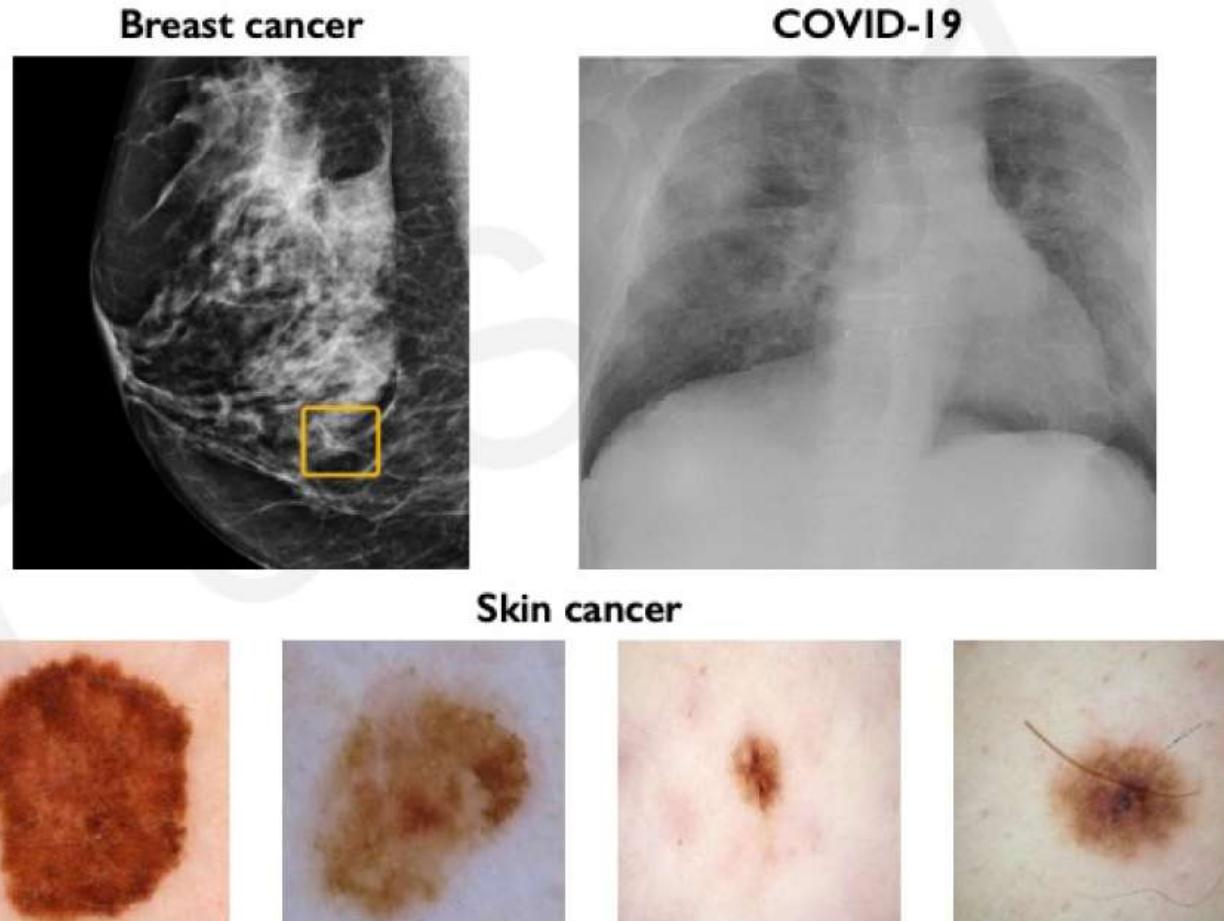
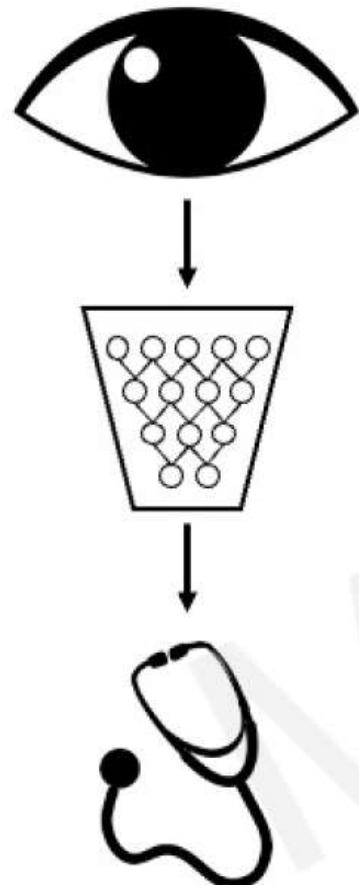
Fuente: <http://introtodeeplearning.com/>

Carros autónomos



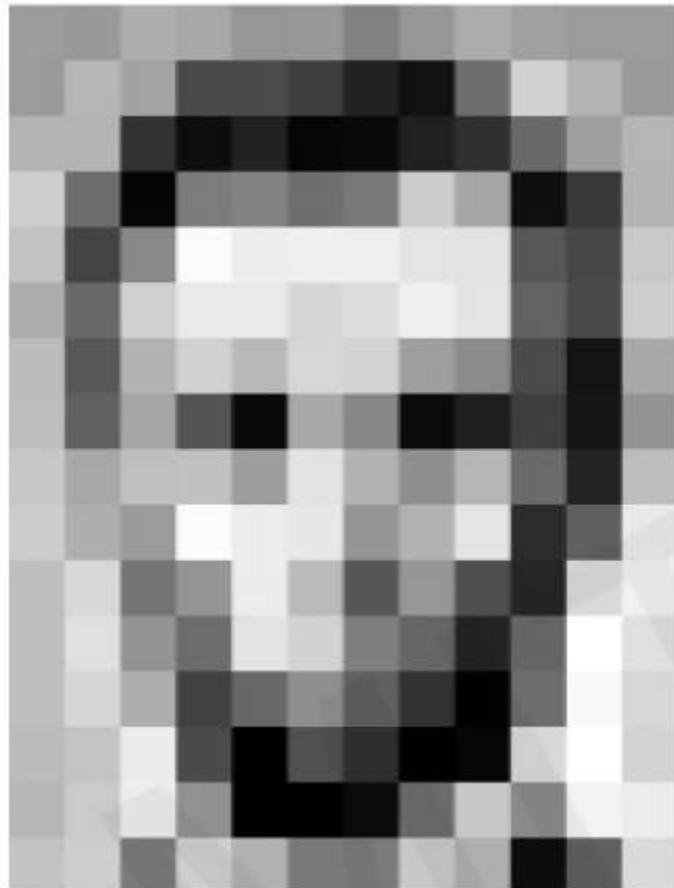
Fuente: <http://introtodeeplearning.com/>

Cuidado de la salud



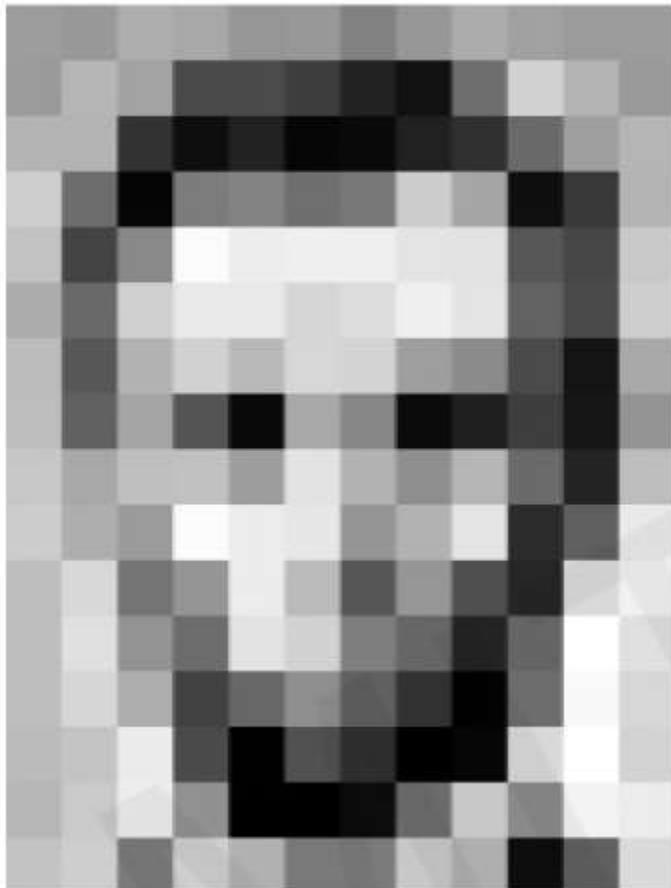
Fuente: <http://introtodeeplearning.com/>

Como ven las computadoras?



Fuente: <http://introtodeeplearning.com/>

Como ven las computadoras?



157	153	174	168	150	152	129	151	172	161	155	156
155	162	163	74	75	62	33	17	110	210	160	154
180	180	50	14	34	6	10	33	48	105	159	181
206	169	5	124	191	111	120	204	166	15	56	180
194	68	137	251	297	299	299	228	227	87	71	201
172	165	207	233	233	214	220	239	228	95	74	206
188	88	179	269	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	105	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	85	150	79	38	218	241
190	224	147	168	227	210	127	102	36	101	255	224
190	214	173	66	103	143	95	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	209	138	243	236
196	206	123	207	177	121	123	209	175	13	96	218

Fuente: <http://introtodeeplearning.com/>

Como ven las computadoras?

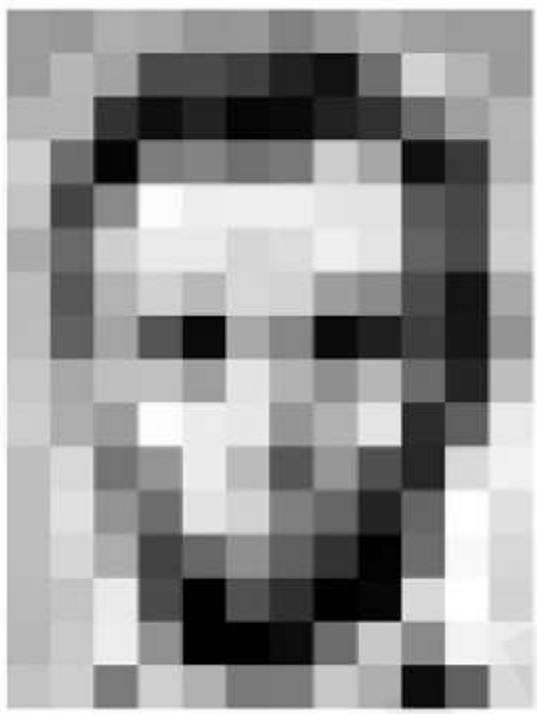


157	153	174	168	160	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	105	159	181
206	109	5	124	191	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	105	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	168	227	210	127	102	36	101	255	224
190	214	173	66	103	143	95	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	209	138	243	236
196	206	123	207	177	121	123	209	175	13	96	218

157	153	174	168	160	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	191	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	105	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	168	227	210	127	102	36	101	255	224
190	214	173	66	103	143	95	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	209	138	243	236
196	206	123	207	177	121	123	209	175	13	96	218

Fuente: <http://introtodeeplearning.com/>

Tarea de visión computacional



167	153	124	168	166	162	129	151	172	161	146	146
166	162	163	74	75	62	39	17	113	216	180	154
180	180	59	14	34	6	10	33	48	106	155	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	261	237	236	239	229	227	87	71	201
172	105	257	233	233	214	220	239	238	98	74	206
188	88	179	208	185	216	211	158	139	76	20	149
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	196	227	178	143	182	106	36	190
205	174	155	252	236	231	149	129	228	43	95	234
190	216	116	149	236	187	86	150	79	36	216	241
190	234	147	159	227	210	127	102	36	101	255	224
190	214	173	66	103	142	96	93	3	106	246	216
187	196	235	79	1	81	47	9	6	217	256	211
183	202	237	145	0	0	12	108	209	158	243	236
195	206	123	207	177	121	123	200	175	12	96	218

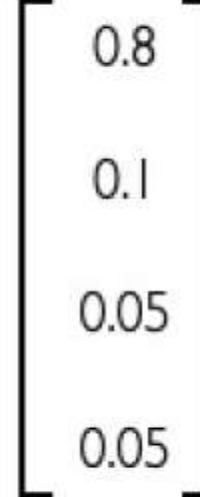
classification

Lincoln

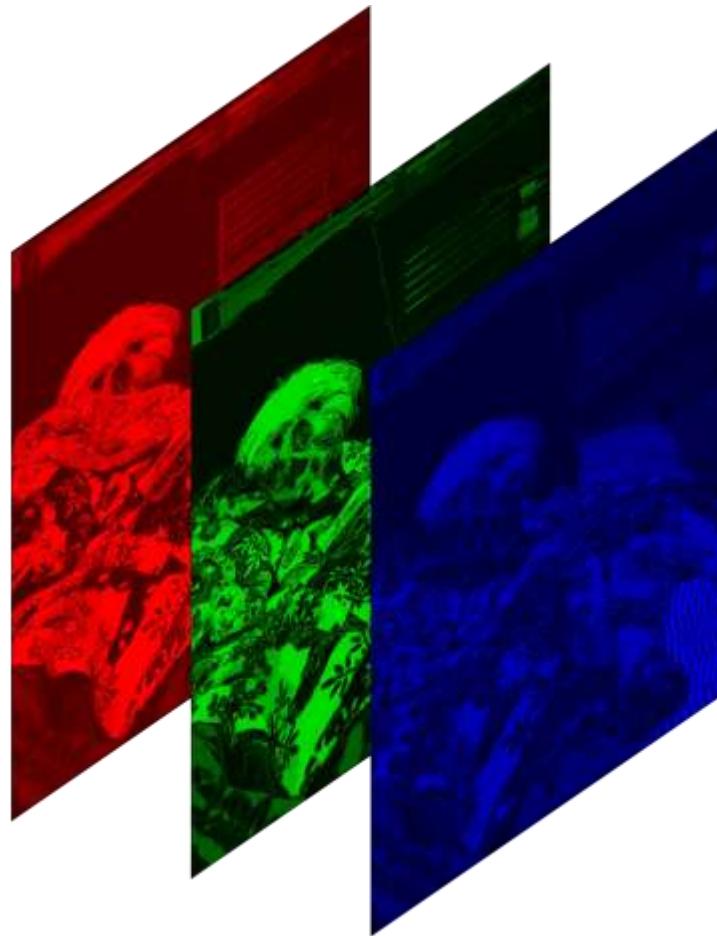
Washington

Jefferson

Obama



Representación de imágenes



Fuente: https://e2eml.school/convert_rgb_to_grayscale.html

Representación de imágenes



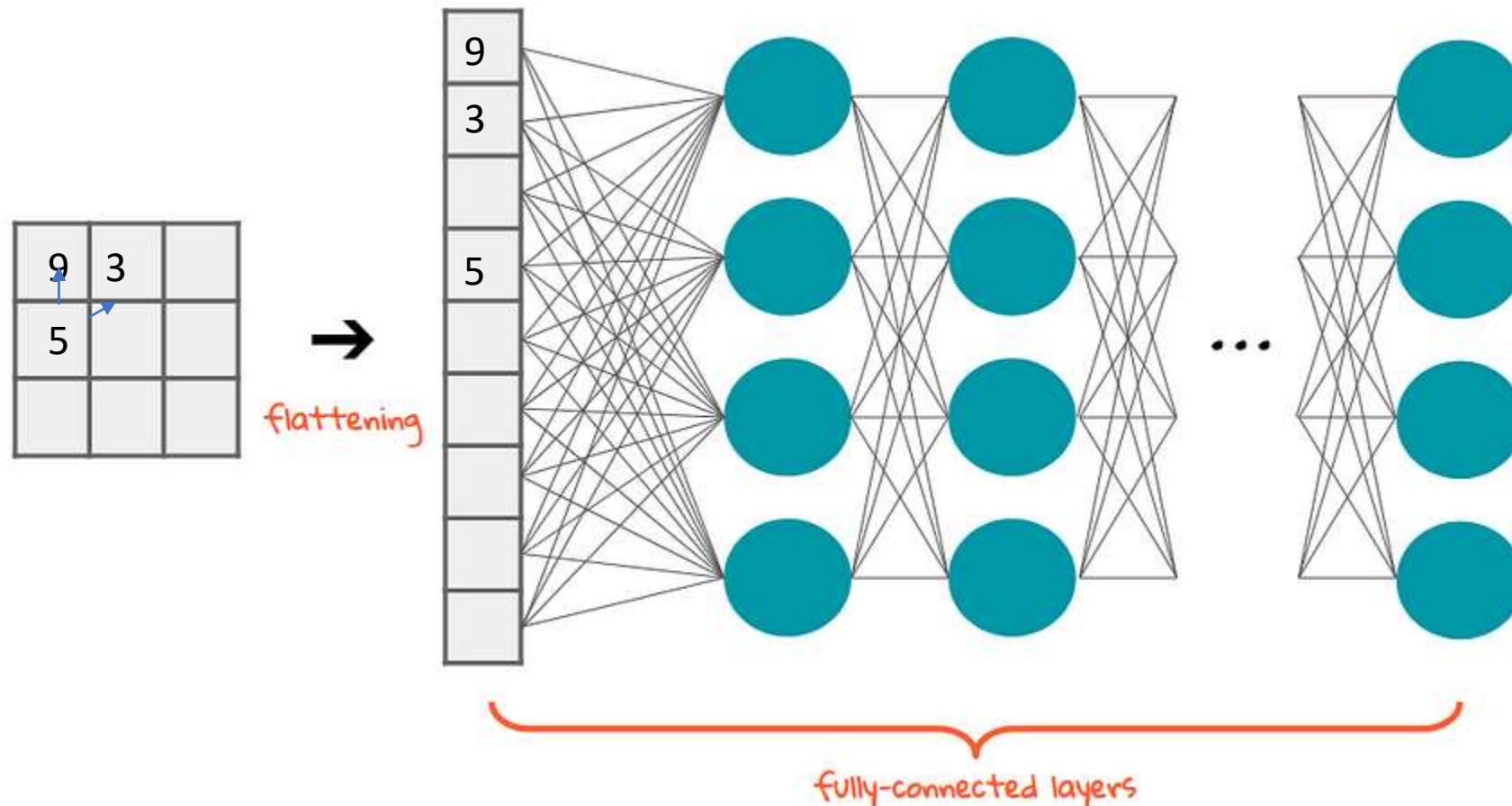
Perro o muffin?



$$64 \times 64 \times 3 = 12\,288$$

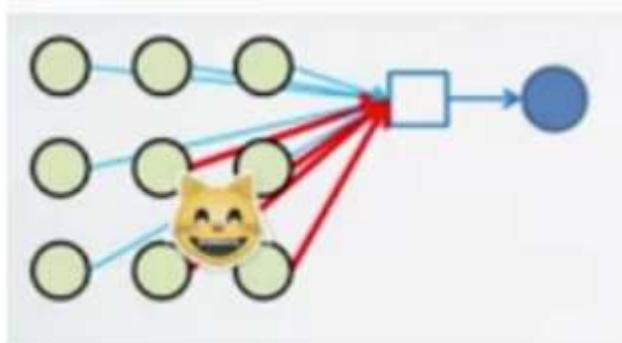
Fuente: <https://www.freecodecamp.org/>

Problemas con MLP

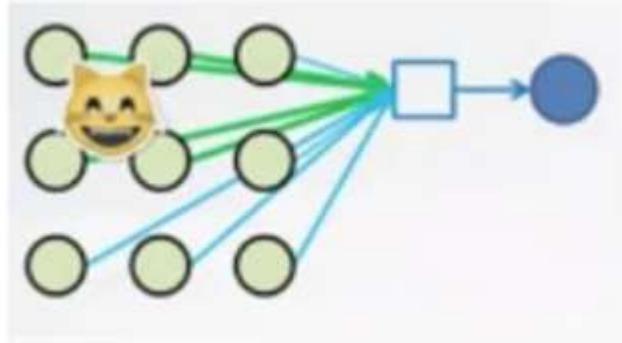


Fuente: <https://towardsdatascience.com/the-most-intuitive-and-easiest-guide-for-convolutional-neural-network-3607be47480>

Problemas con MLP



In this case, the **red weights** will be modified to better recognize cats



In this case, the **green weights** will be modified.

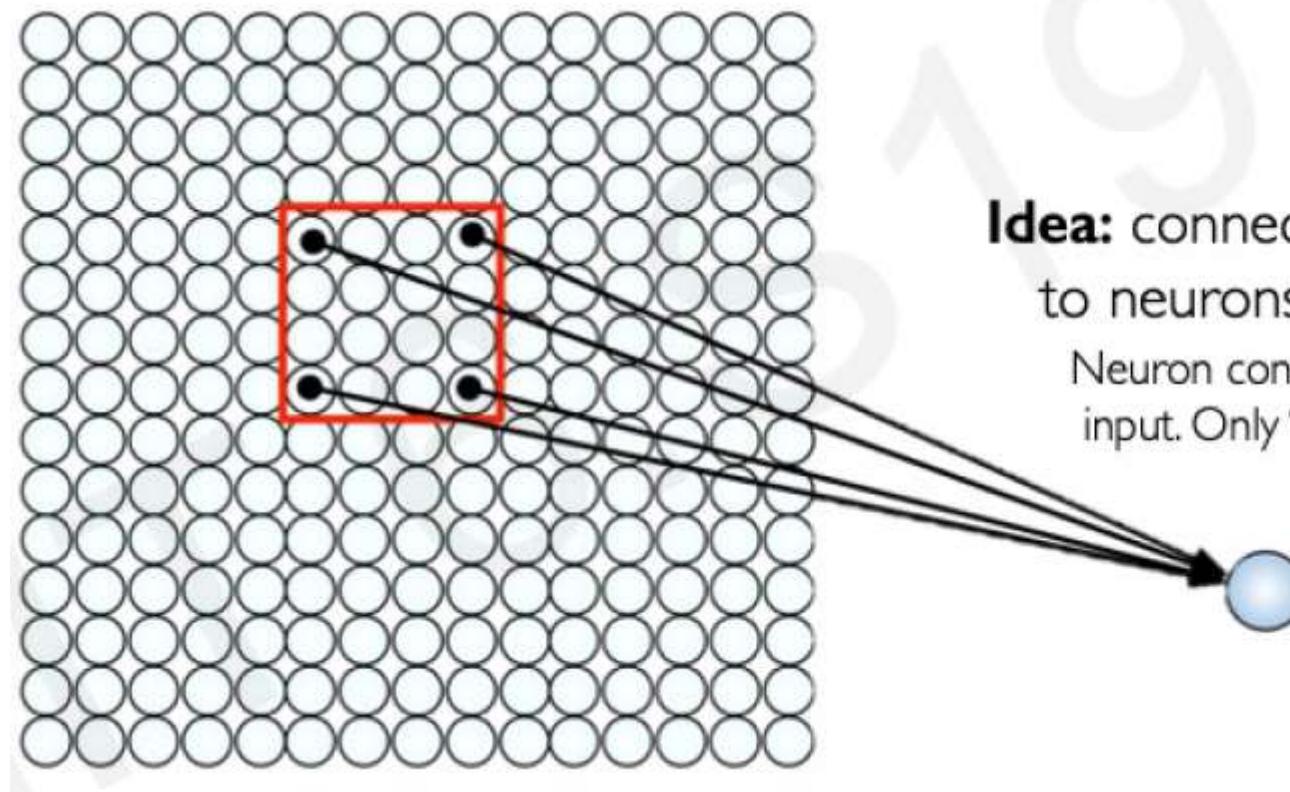
Detección de características

Pixeles próximos están más relacionados que píxeles distantes



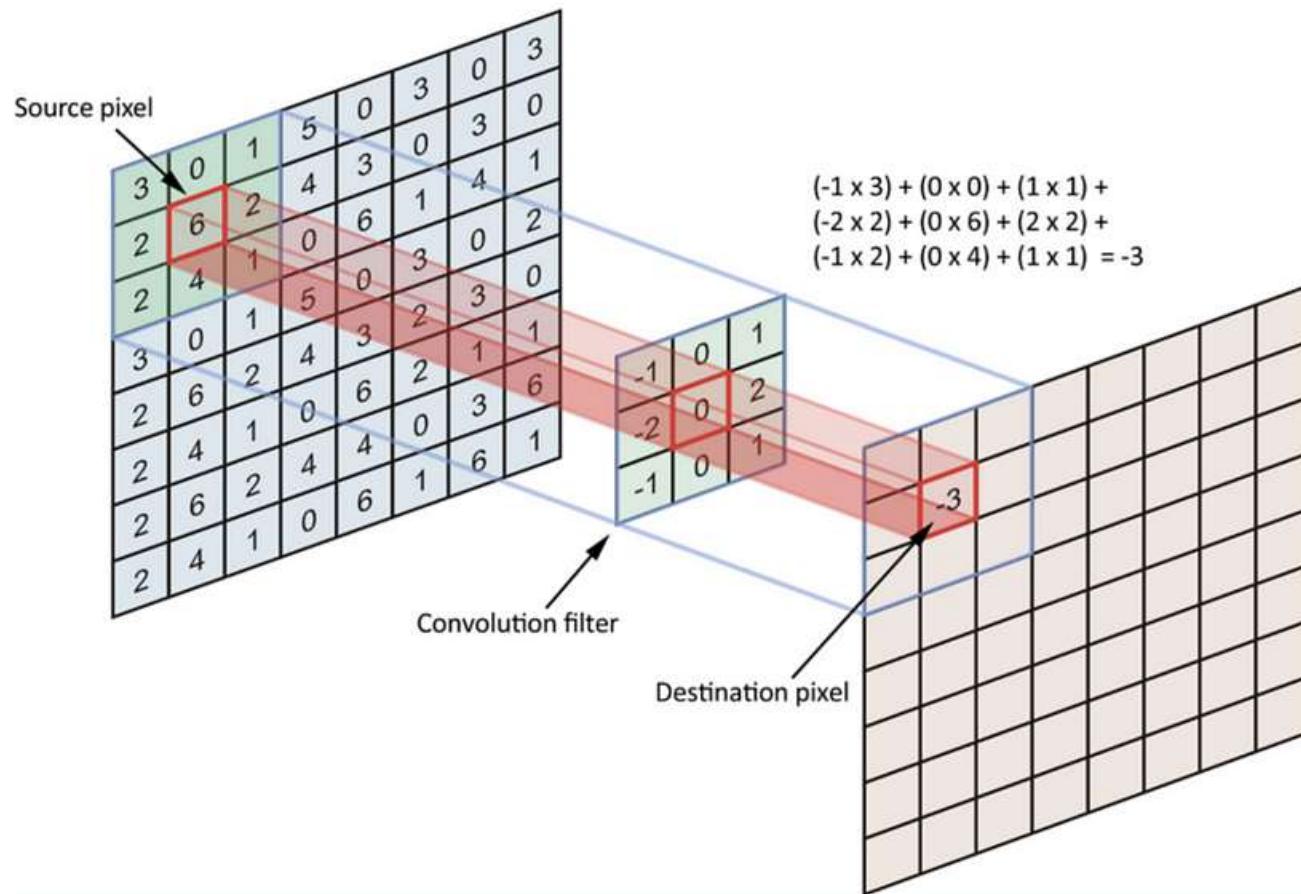
Fuente: <http://introtodeeplearning.com/>

Usando estructuras espaciales



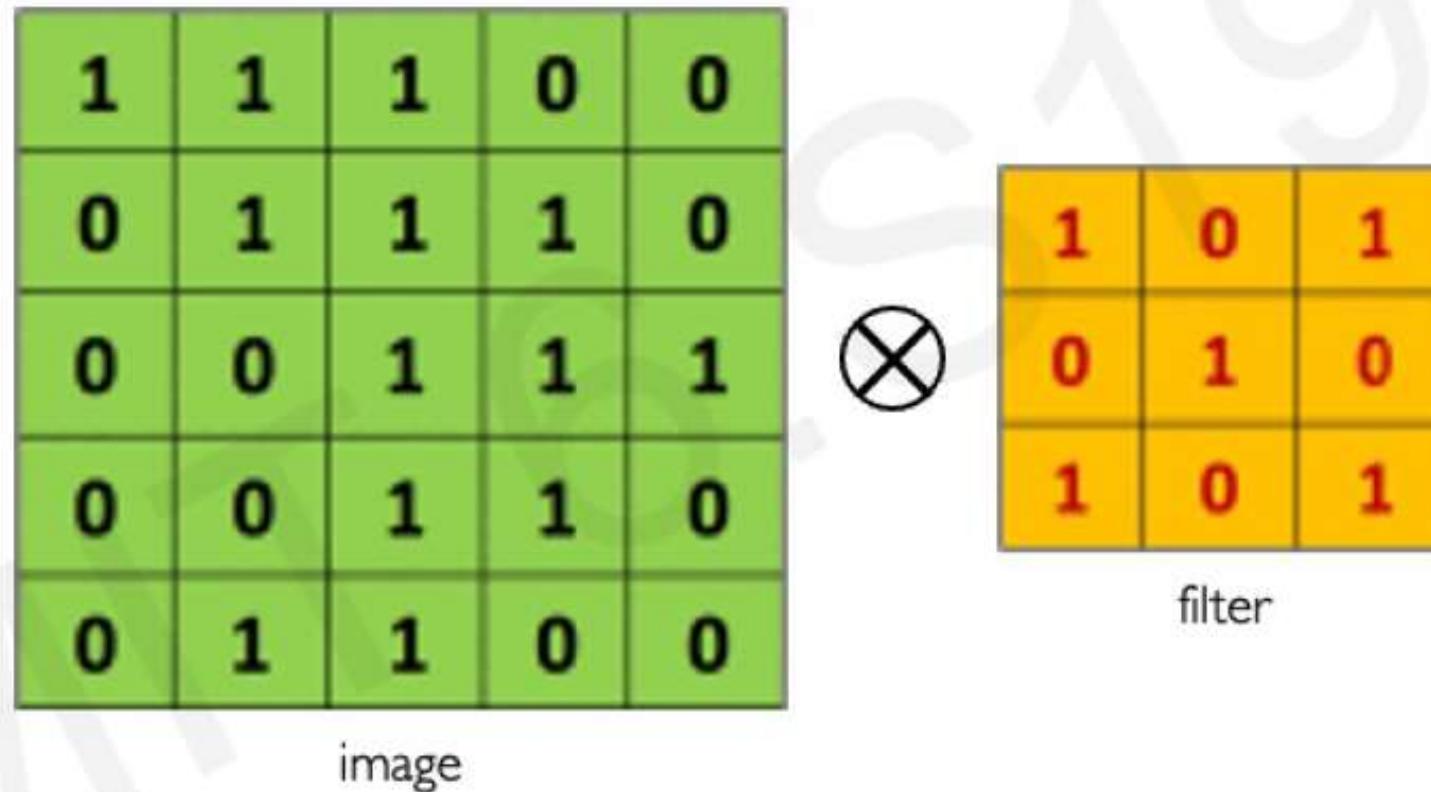
Idea: connect patches of input
to neurons in hidden layer.
Neuron connected to region of
input. Only “sees” these values.

Convolución



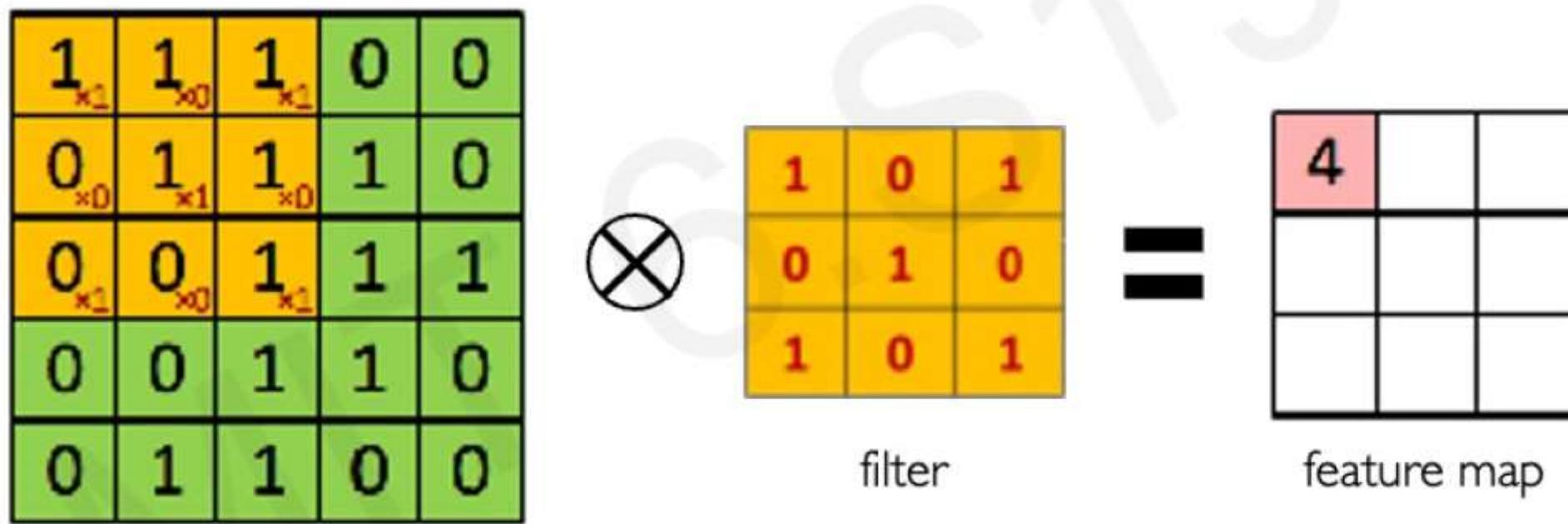
Fuente: <https://towardsdatascience.com/simple-introduction-to-convolutional-neural-networks-cdf8d3077bac>

Convolución - ejemplo



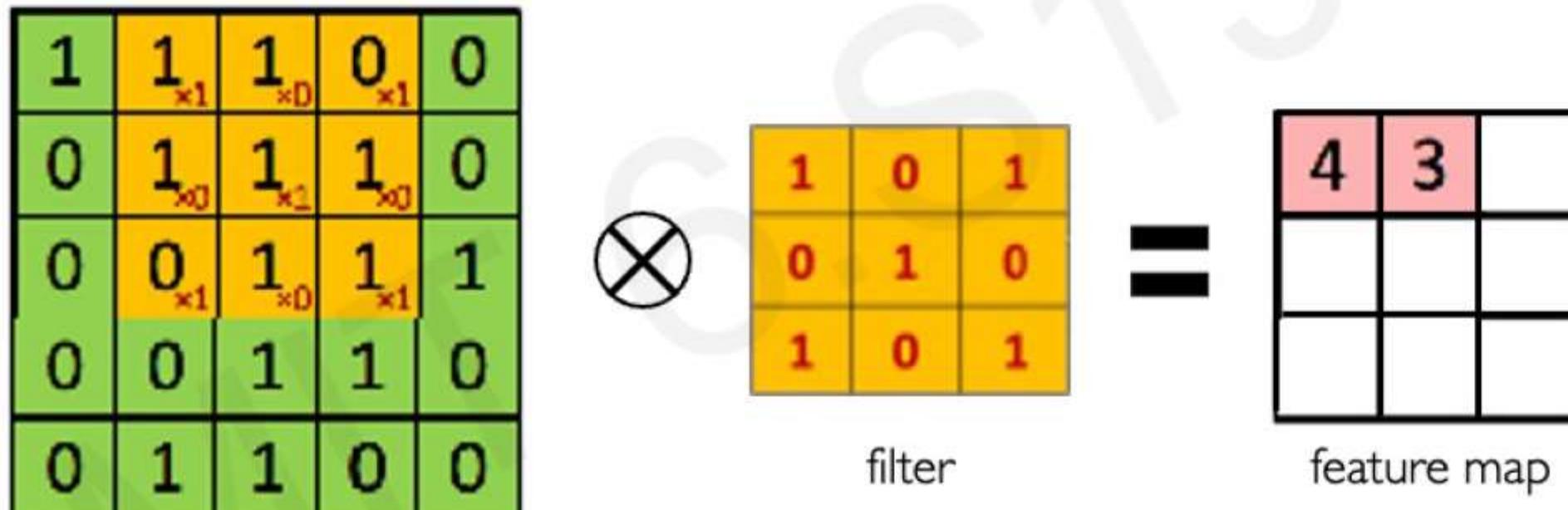
Fuente: <http://introtodeeplearning.com/>

Convolución - ejemplo

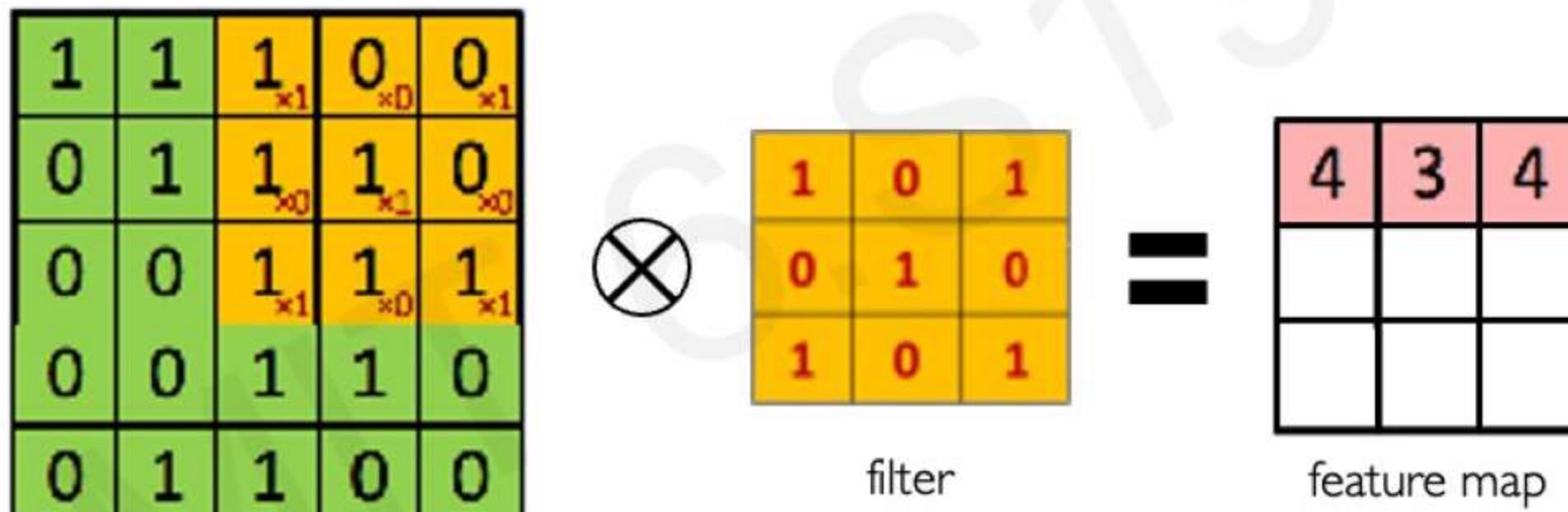


Fuente: <http://introtodeeplearning.com/>

Convolución - ejemplo

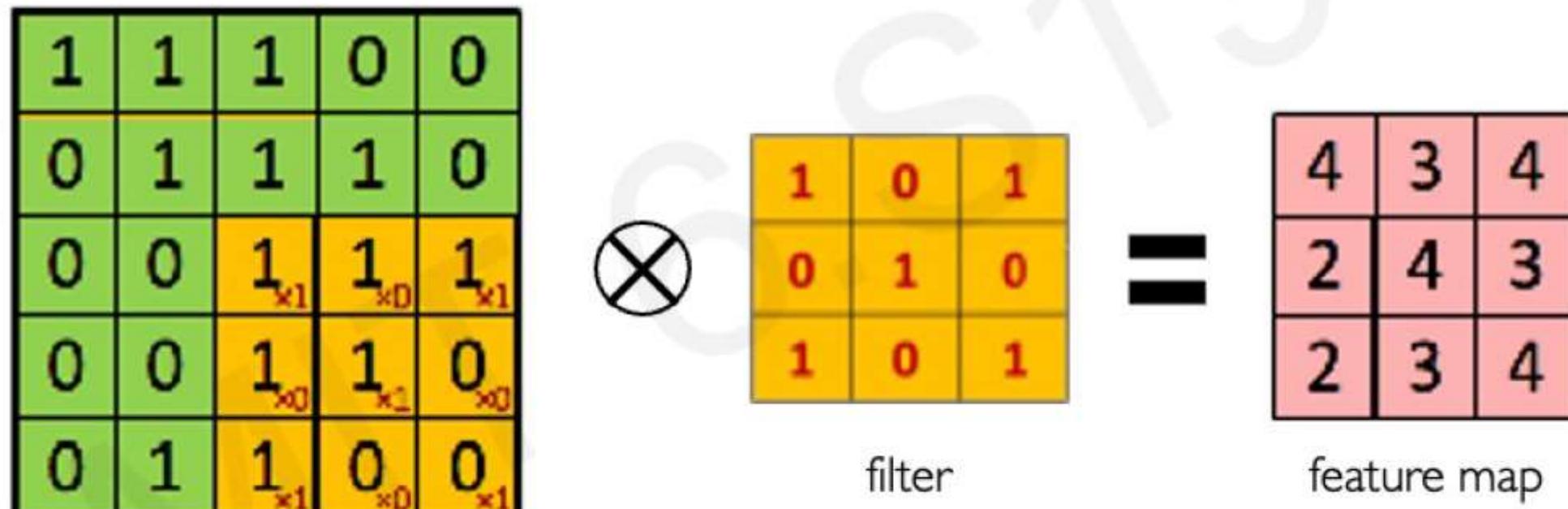


Convolución - ejemplo



Fuente: <http://introtodeeplearning.com/>

Convolución - ejemplo



Fuente: <http://introtodeeplearning.com/>

Mapa de características



Original



Sharpen



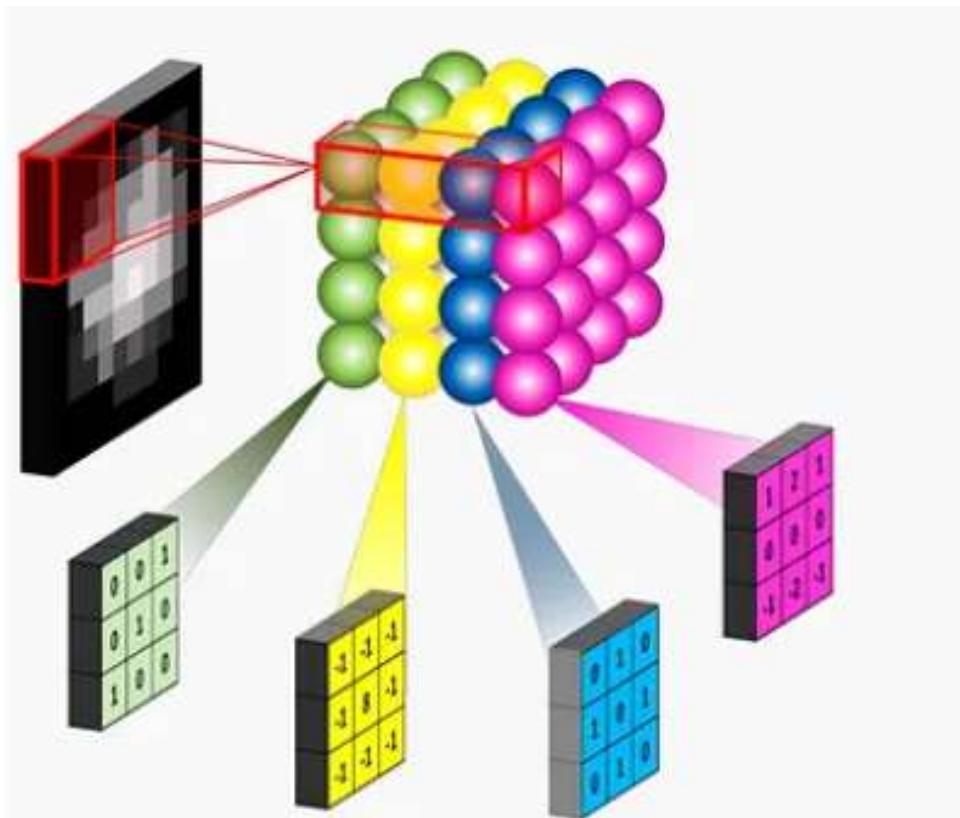
Edge Detect



"Strong" Edge
Detect

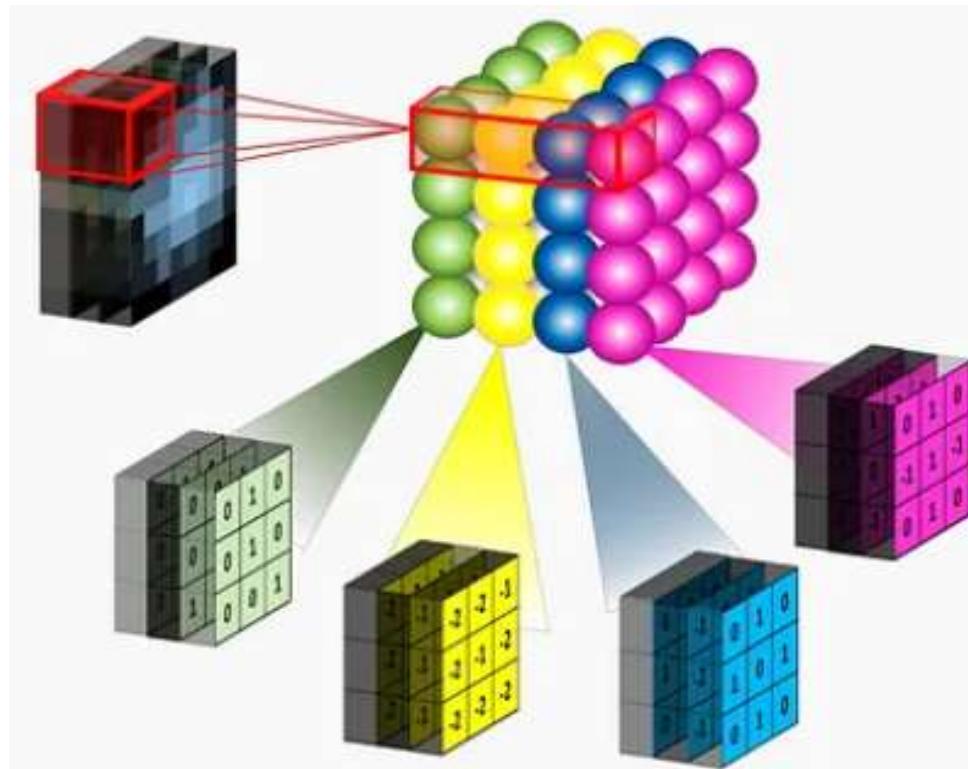
Fuente: <http://introtodeeplearning.com/>

Disposición espacial del volumen de salida



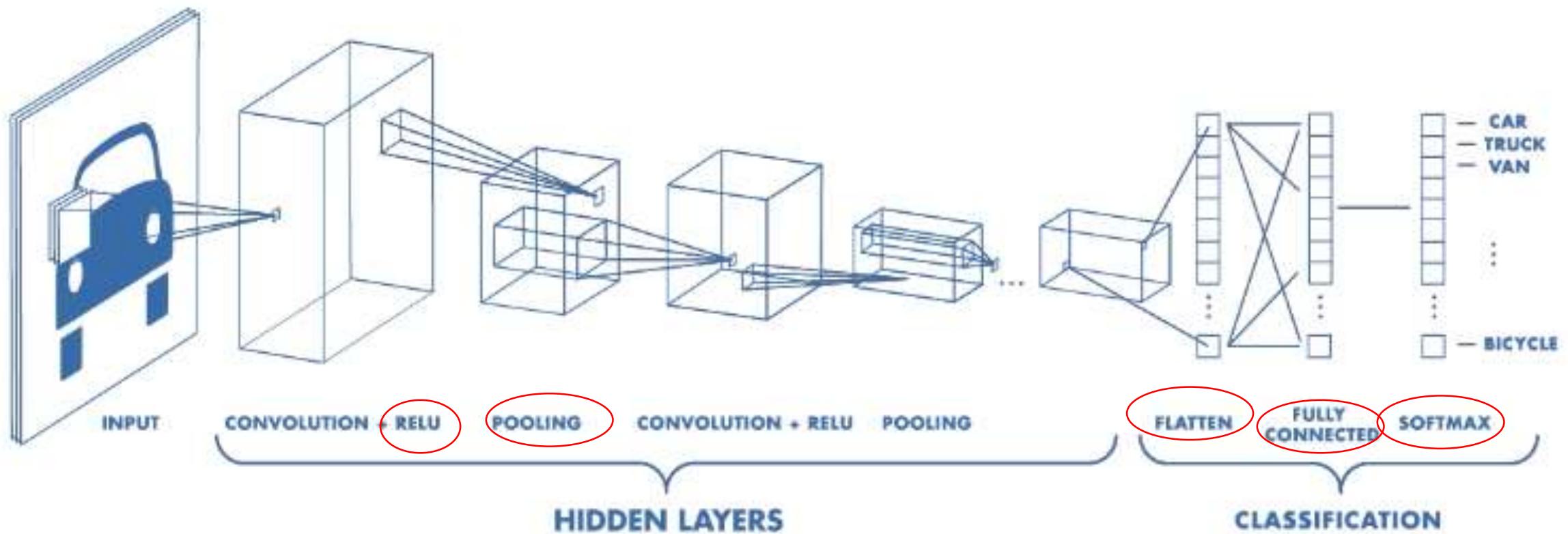
Convolutional layer with four 3×3 filters on a
black and white image (just one channel)

Disposición espacial del volumen de salida



Convolutional layer with four 3x3 filters on an **RGB image**. As you can see, the filters are now cubes, and they are applied on the full depth of the image..

Actividad para la próxima aula: Recordar conceptos clave



Aula 2

Aula 2

Conceptos básicos



CORRELACIÓN VS. CAUSALIDAD

CAUSALIDAD

Cuando una cosa (una causa) es responsable de que ocurra otra cosa (un efecto)



CAUSA



EFFECTO

CORRELACIÓN

Cuando dos o más cosas parecen estar relacionadas

EN VERANO



Aumentan las ventas de helados



Aumentan las quemaduras solares

Las ventas de helados y las tasas de quemaduras solares están correlacionadas



¿Significa esto que comer helado aumenta el riesgo de sufrir una quemadura solar?

¡La correlación no siempre significa causalidad!



En días soleados, es más probable que la gente:



coma helados



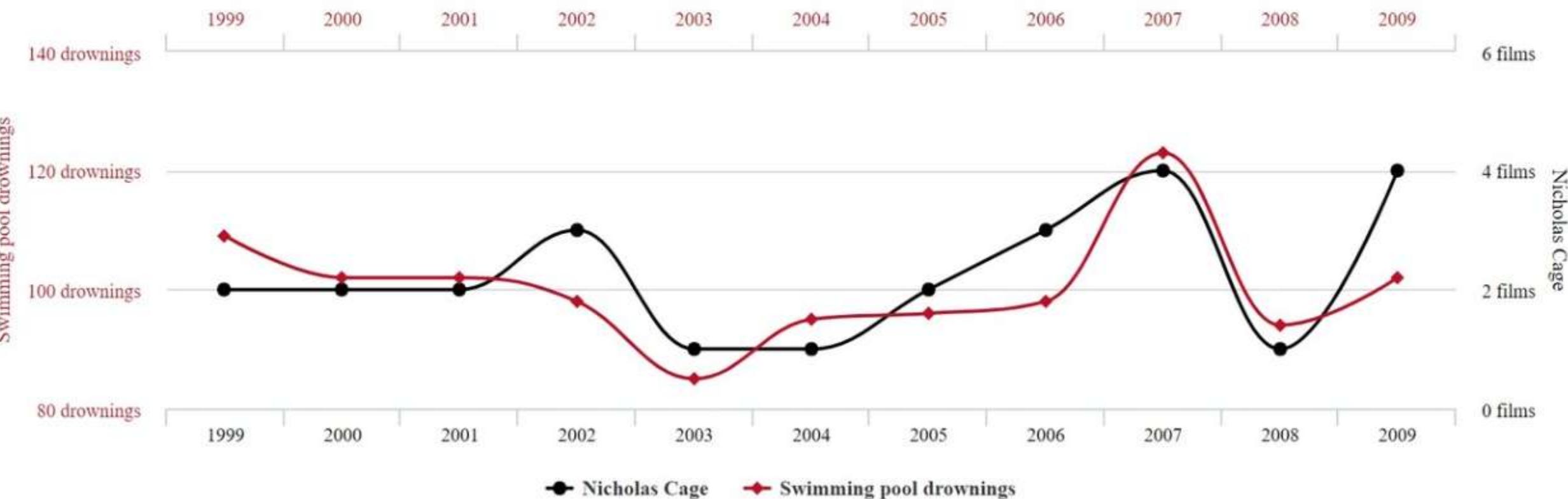
se queñe con el sol



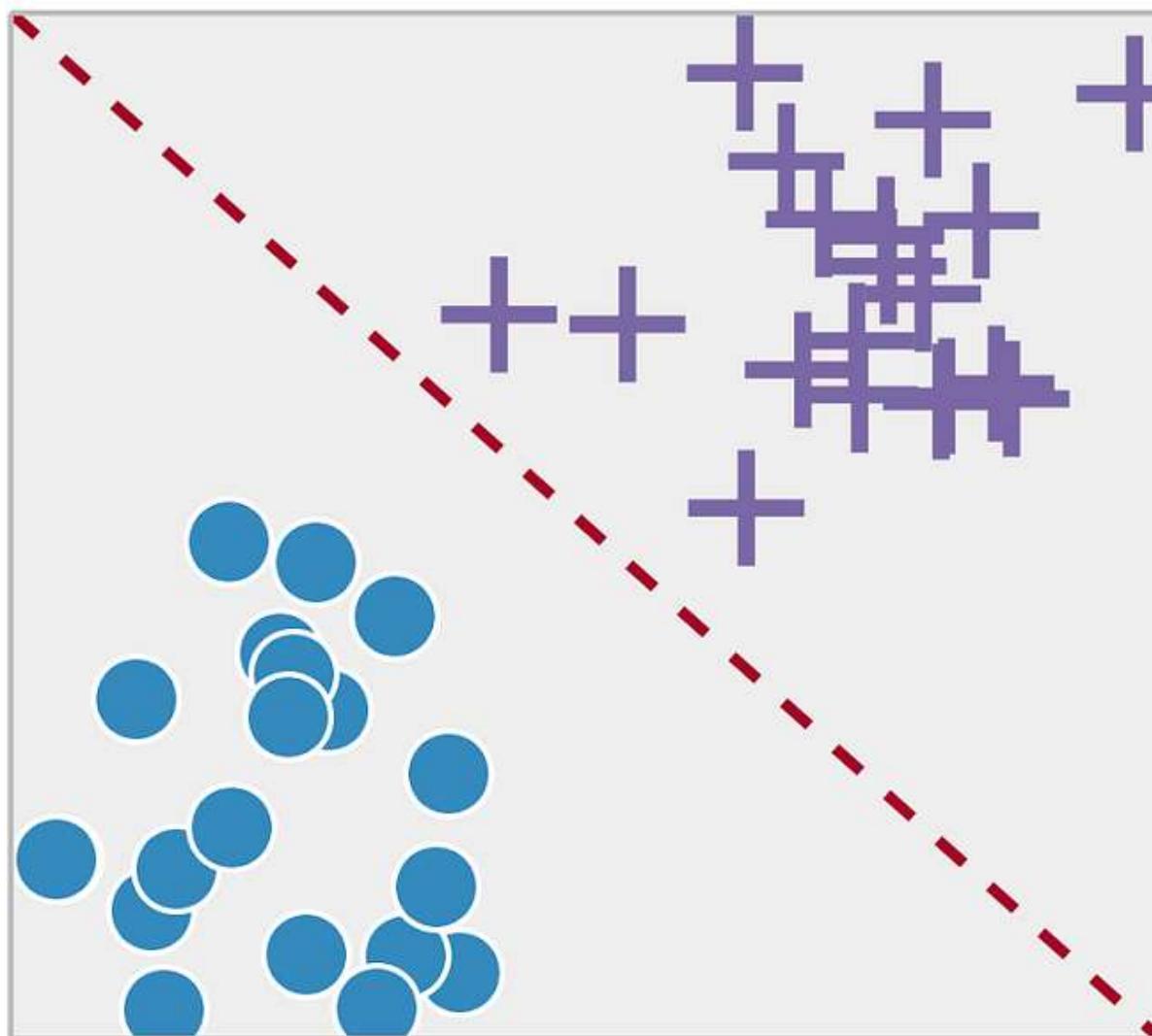
¡Una correlación no siempre significa que una cosa causa otra!

Number of people who drowned by falling into a pool correlates with Films Nicolas Cage appeared in

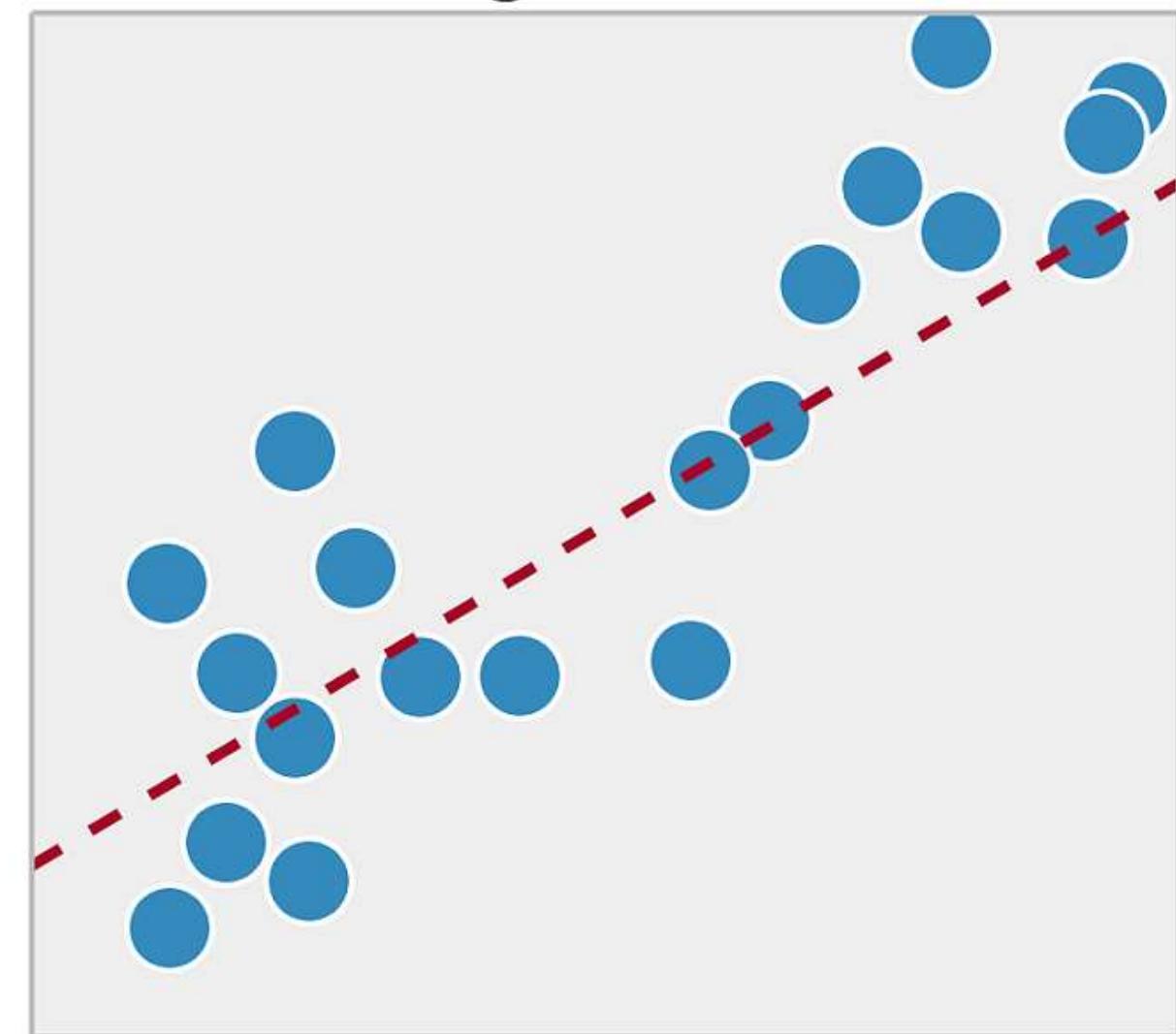
Correlation: 66.6% ($r=0.666004$)



Classification



Regression



Classification Data

X_1	X_2	X_3	X_p	Y
				Paga
				No paga
				Paga
				No paga

Categorical
"Labels"

Regression Data

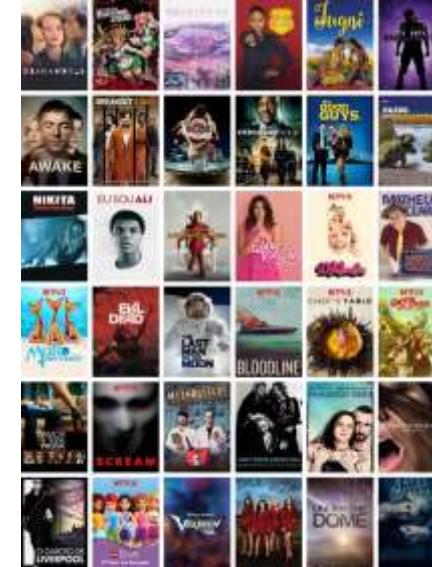
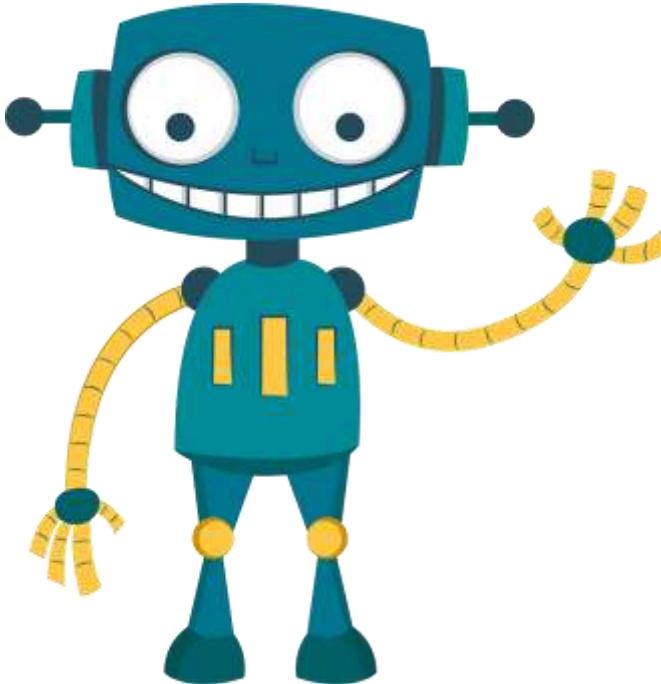
X_1	X_2	X_3	X_p	Y
				5.2
				1.3
				23.0
				7.4

Numeric
Target

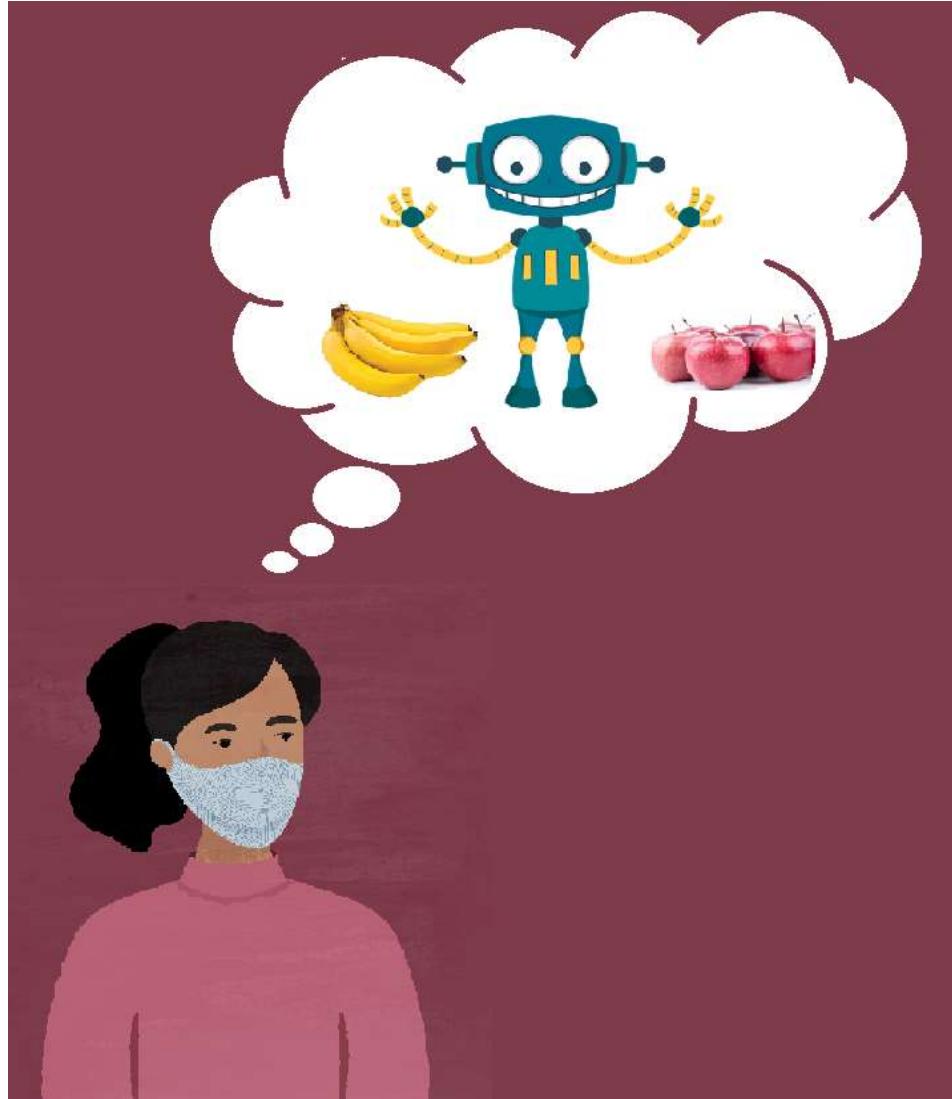
Como funciona?



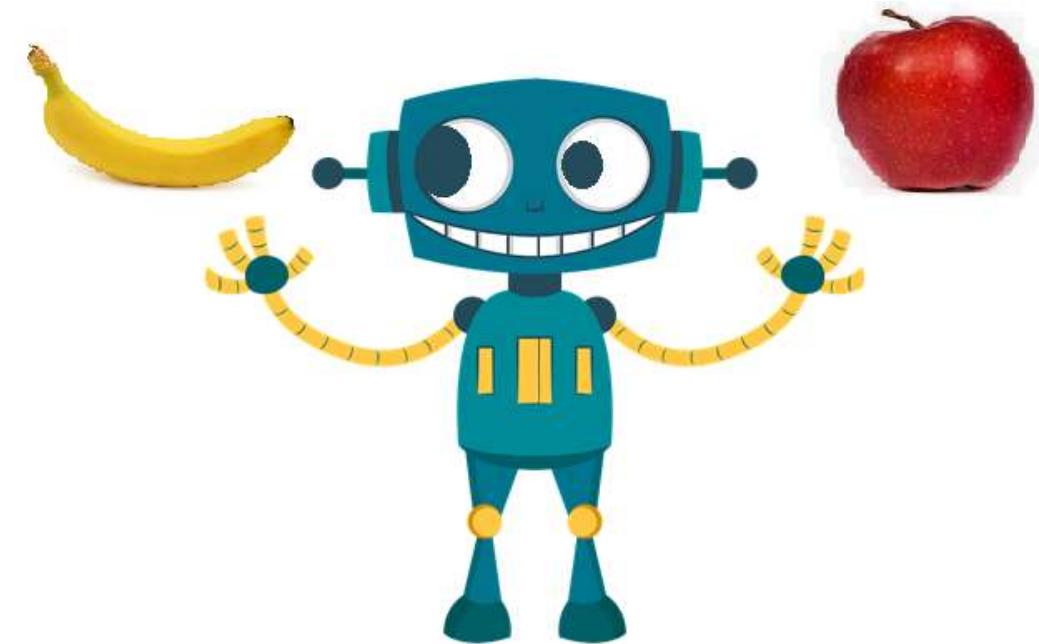
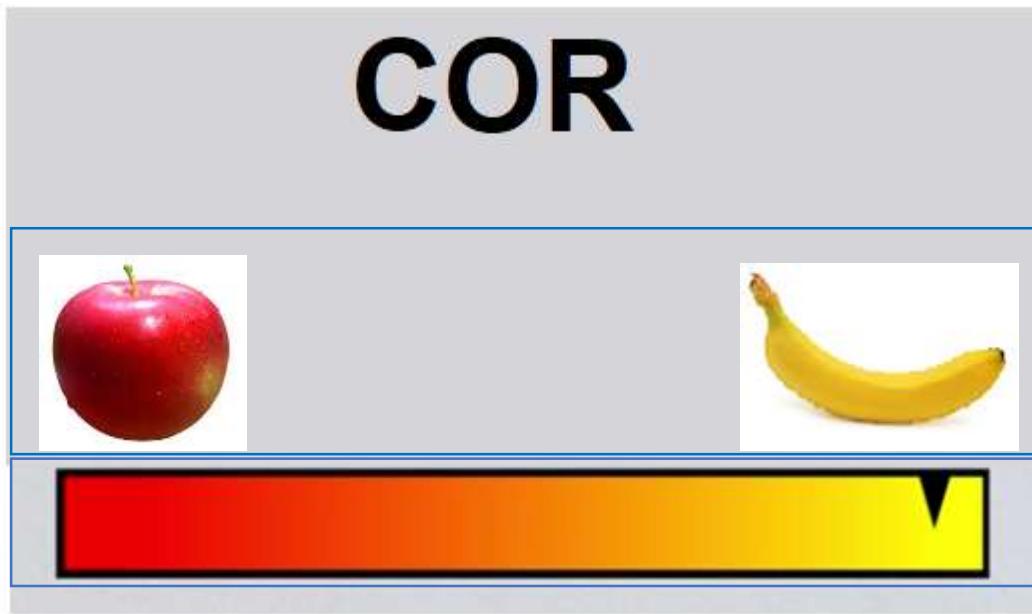
sistema **equilibrio** **espacio** **lugar**
equipo **cosa** **humildad** **nuevo** **desorden**
va **fin** **complementarias** **ESPERANZA** **ayuda** **tr** **energías** **diferentes**
va **creemos** **valor** **calma** **encanta** **inteligencia** **Emoción** **ser**
nos **nuevas** **aprendizaje** **pro** **organización** **cambios** **EQUPO**
aquel **Emprender** **ESFUERZO** **equilibrios** **lugar**
tener **opuestas** **fuerzas** **nadie** **paso** **temporada** **losimientos** **ESPIRITUALIDAD**
Buscar **miembros** **sistemas** **dejar** **comprender** **diligencia** **tras** **estrategias**
mejorar **ESTABILIDAD** **ENTUSIASMO** **entendimiento** **busqueda** **barrera** **horizontal**
universo **definición** **Ecología** **incomoda** **ECUANIMIDAD** **sistémicos** **sistémico** **volver** **dentro**
Empezar **interior** **primer** **Empiezo** **Entender** **Recoger** **lograr** **equipos** **EJE**
ESCALAR **Equidad** **pueden** **reponerse** **olvidar** **buscar**



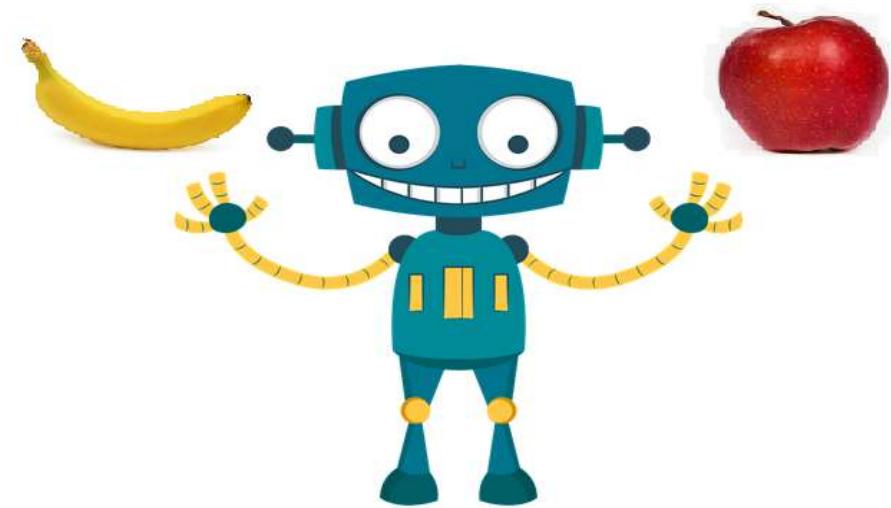
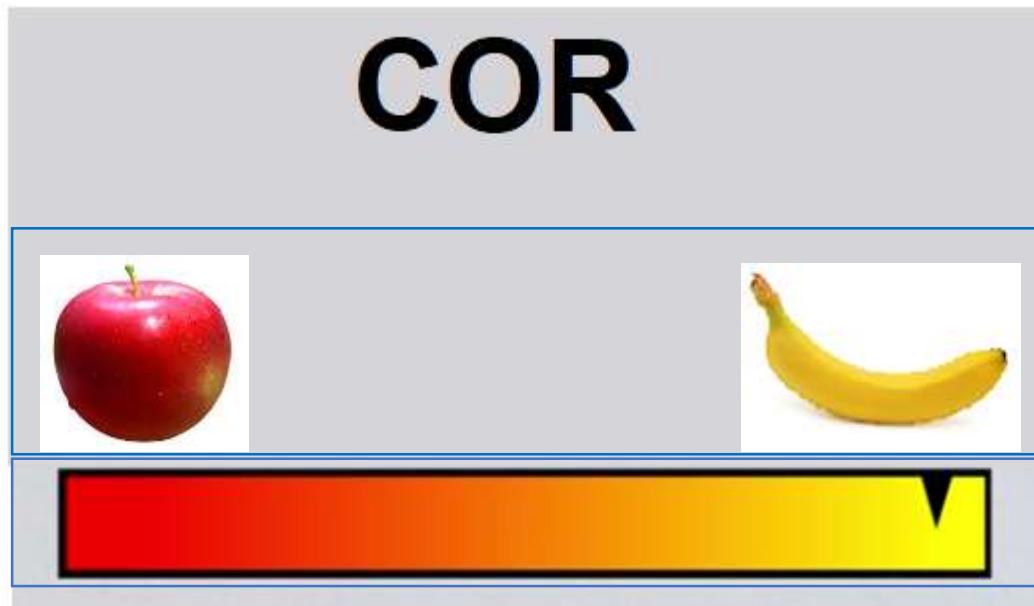
Como crear um modelo que aprende?



Selección de características

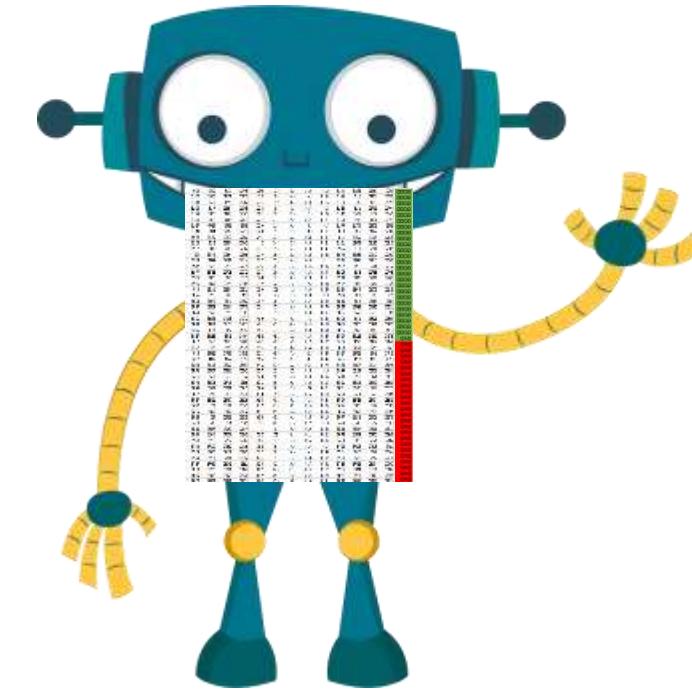


Selección de características



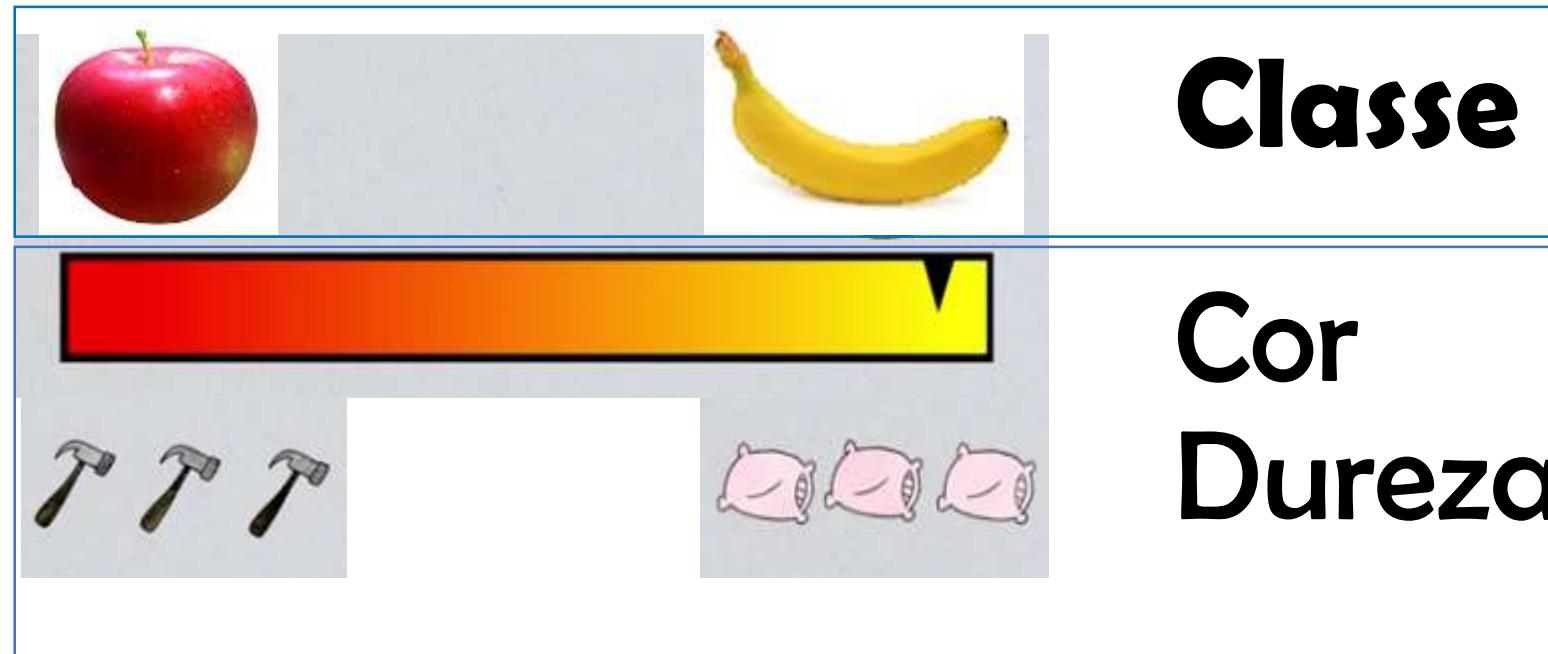
Resumen del proceso

Diversas fuentes de datos

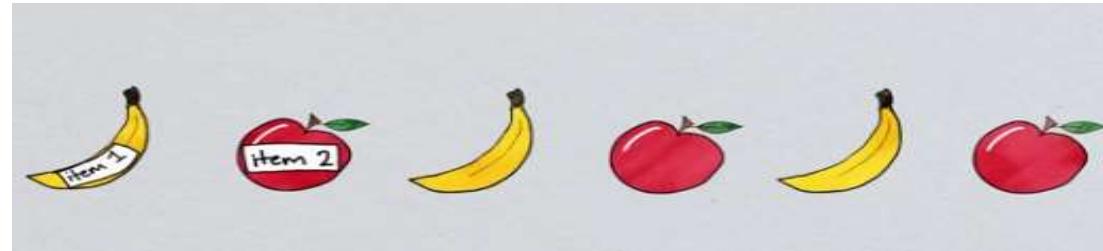


88.99	-44.99	319.7	317.45	317.5	317.26	318.86	314.85	313.2	313.44	313.37	314.67	-44.99
315.62	716.36	318.71	317.72	318.25	318.54	318.54	314.45	313.25	314.48	315.38	315.38	315.38
318.43	318.97	317.15	318.42	310.83	318.99	318.24	318.91	315.18	315.84	316.19	316.19	316.19
318.93	317.7	318.04	318.48	320.85	318.77	318.79	318.8	315.38	316.1	317.21	317.44	317.44
317.96	318.56	318.68	320.63	321.21	318.33	318.58	317.7	318.25	318.42	318.89	317.7	318.87
317.17	318.25	317.25	321.89	322.21	317.21	318.57	317.7	318.19	318.31	318.81	317.7	318.81
319.37	-94.99	318.99	321.49	321.49	319.46	318.7	316.79	317.79	318.71	319.99	319.99	319.99
318.66	320.84	320.84	322.12	321.36	321.87	321.39	321.46	317.81	317.1	318.87	318.42	318.42
320.62	321.59	322.38	322.87	320.21	323.76	322.39	322.37	318.64	318.7	319.79	321.38	321.38
322.05	322.5	323.04	324.42	324.42	324.09	322.95	325.82	319.21	319.21	320.72	321.98	321.18
322.37	323.32	322.89	325.62	325.62	324.36	324.19	325.82	320.98	321.2	321.2	321.2	321.2
325.59	324.42	323.48	324.48	321.34	324.49	324.49	323.87	321.18	321.48	321.48	321.48	321.48
325.03	325.39	326.87	326.14	326.27	327.48	326.35	324.69	323.1	323.16	323.88	325.13	325.66
326.17	326.86	327.18	327.78	326.82	328.57	327.34	323.66	323.36	323.37	324.8	326.21	326.32
328.77	327.83	327.75	329.72	329.87	329.39	328.08	328.32	328.93	329.18	329.5	327.93	327.93
329.99	328.96	328.96	328.5	328.48	329.37	328.37	328.31	327.91	327.18	328.94	329.08	329.08
324.18	323.86	323.3	323.21	323.21	323.36	323.6	323.36	323.26	323.48	323.26	323.13	323.13
323.4	323.41	322.04	323.21	323.21	323.31	323.06	323.26	323.2	320.34	320.21	321.08	322.13
322.79	322.26	323.5	328.58	328.87	324.34	323.03	323.34	323.34	323.25	323.25	323.13	323.13
322.93	323.62	323.7	326.57	326.74	326.27	324.93	322.75	321.99	321.78	322.4	322.4	322.4
328.21	328.44	328.86	328.45	328.28	328.28	327.72	326.05	322.81	323.86	323.86	323.86	323.86
325.20	325.28	325.28	325.17	325.17	325.17	325.17	325.17	325.17	325.17	325.17	325.17	325.17
329.23	347.47	347.47	342.51	342.51	342.51	342.51	342.51	339.69	336.88	336.88	339.93	339.93
342.75	342.75	343.1	344.84	345.32	343.09	342.26	343.06	339.66	339.09	339.28	340.44	341.13
343.7	344.5	345.26	347.28	347.43	348.79	345.4	343.26	341.07	341.25	342.99	342.78	342.78
344.2	344.2	344.2	344.2	344.2	344.2	344.2	344.2	344.2	344.2	344.2	344.2	344.2
348.97	347.86	349.18	349.21	349.49	347.47	349.8	349.8	349.39	349.17	349.44	349.9	349.13
348.02	348.47	349.42	351.89	351.84	351.25	349.32	348.1	346.45	346.26	347.81	348.45	349.61
351.43	351.75	352.25	352.55	354.22	353.79	352.39	352.45	349.72	348.88	350.27	351.34	352.48
352.76	353.07	353.68	353.42	355.87	355.13	353.3	351.87	349.9	350.29	351.2	352.91	352.91
353.86	354.24	355.38	356.82	357.18	356.23	355.82	354.91	353.18	353.82	354.21	354.19	354.19
354.21	354.21	354.21	354.21	354.21	354.21	354.21	354.21	354.21	354.21	354.21	354.21	354.21
355.98	355.72	357.81	358.15	358.15	358.25	357.02	357.02	353.01	353.21	354.18	355.27	355.27
357.16	359.38	359.46	360.28	359.8	357.57	359.52	353.69	353.39	353.39	355.8	357.18	357.18
359.37	359.91	359.91	361.18	361.48	360.95	359.95	357.48	359.84	359.84	357.58	359.54	359.54
360.97	361.31	361.64	361.45	361.79	361.26	361.3	359.46	359.08	359.08	360.7	360.49	360.49
362.45	362.55	364.5	364.24	364.24	364.24	364.24	364.24	359.4	360.8	361.4	361.8	361.8
363.18	363.84	364.28	366.79	365.42	364.47	362.31	363.18	360.47	360.47	362.29	363.76	363.76
365.33	366.15	367.31	368.81	368.87	368.87	365.77	363.9	364.23	365.48	366.27	366.92	366.92
368.15	368.87	369.68	371.14	371.14	371.38	368.27	368.93	366.63	366.11	366.47	366.01	366.31
368.36	370.32	371.68	371.42	371.7	371.12	368.12	368.62	366.82	366.71	368.21	368.9	368.9
371.26	371.12	372.87	374.62	374.62	373.7	371.62	369.37	369.9	369.39	371.24	373.32	373.32
372.77	373.77	373.77	373.77	373.77	373.77	373.77	373.77	373.77	373.77	373.77	373.77	373.77
374.06	374.63	374.71	374.49	374.28	374.19	374.02	374.3	372.99	374.51	374.39	375.7	375.69
374.76	377.37	378.61	380.52	380.43	378.47	377.39	378.86	376.27	376.88	377.47	377.36	377.36
378.37	379.69	380.41	382.1	382.28	382.13	382.05	378.71	376.42	376.32	380.34	379.37	379.37
381.39	382.05	382.64	384.42	384.49	384.38	382.29	380.47	379.67	379.58	380.14	381.74	382.84
382.49	383.66	384.27	385.26	385.26	385.87	384.39	381.78	380.73	380.51	382.69	383.93	383.93
383.01	383.21	384.21	384.21	384.21	384.21	384.21	384.21	384.21	384.21	384.21	384.21	384.21

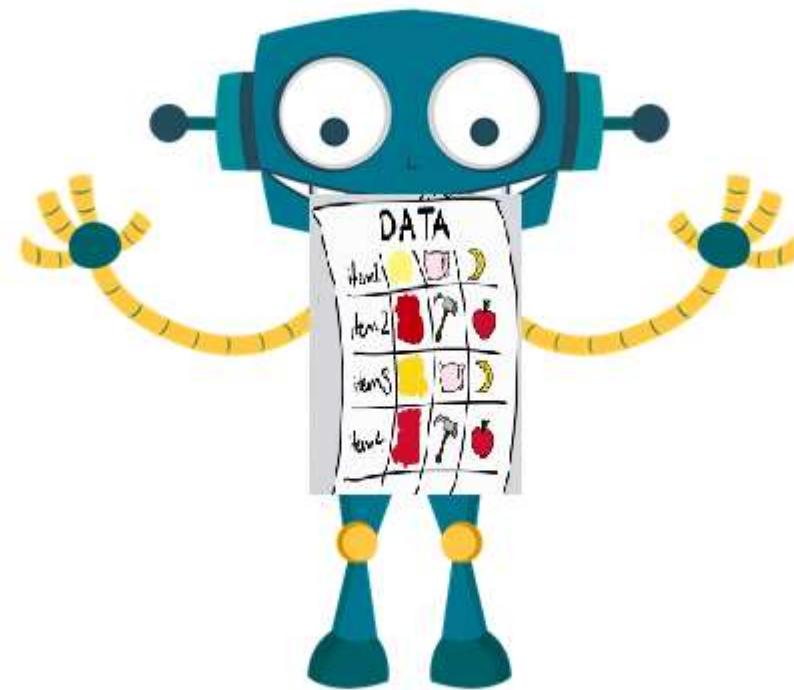
Selección de características



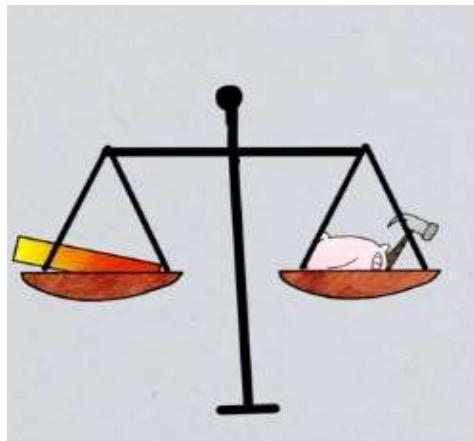
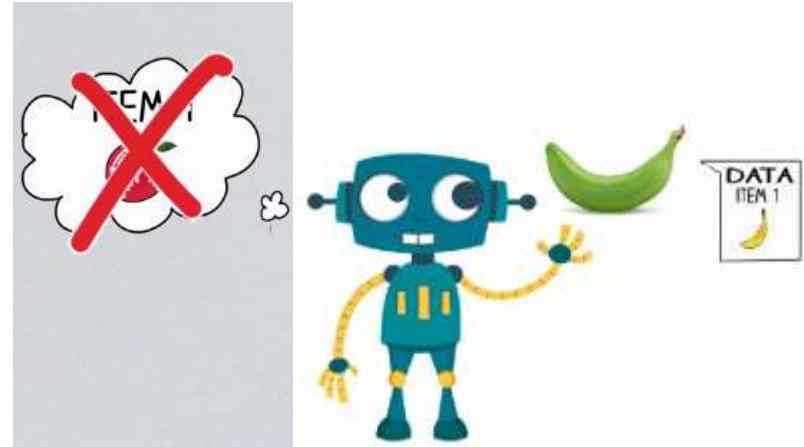
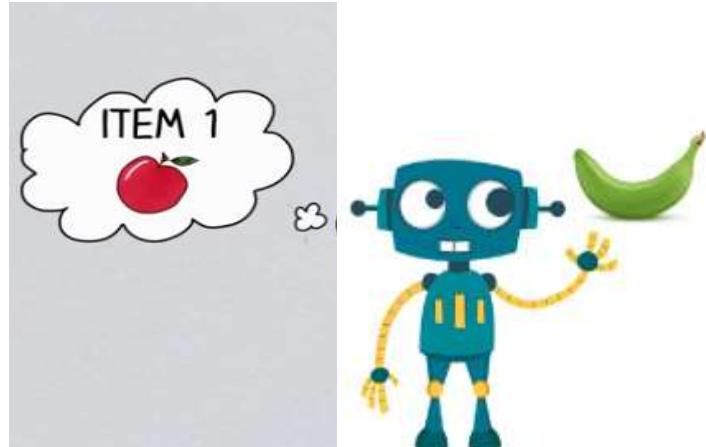
Entrenamiento



DATA			
item1	item2	item3	item4
yellow circle	red blob	yellow square	red blob
pink cat face	hammer	pink cat face	hammer
yellow crescent moon	apple	yellow crescent moon	apple

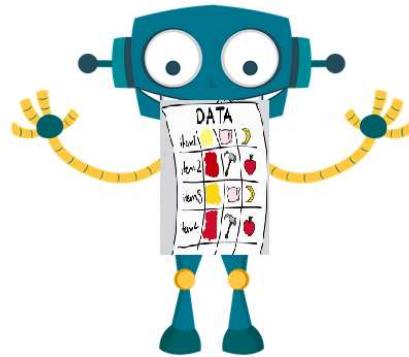
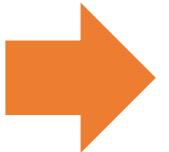


Entrenamiento

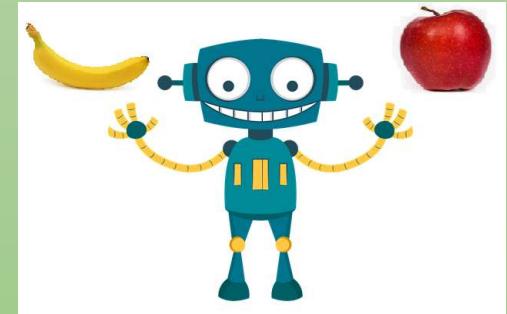


Entrenamiento

DATA			
item1	yellow circle	pink pig	yellow crescent moon
item2	red blob	grey hammer	red apple
item3	yellow blob	pink pig	yellow crescent moon
item4	red blob	grey hammer	red apple



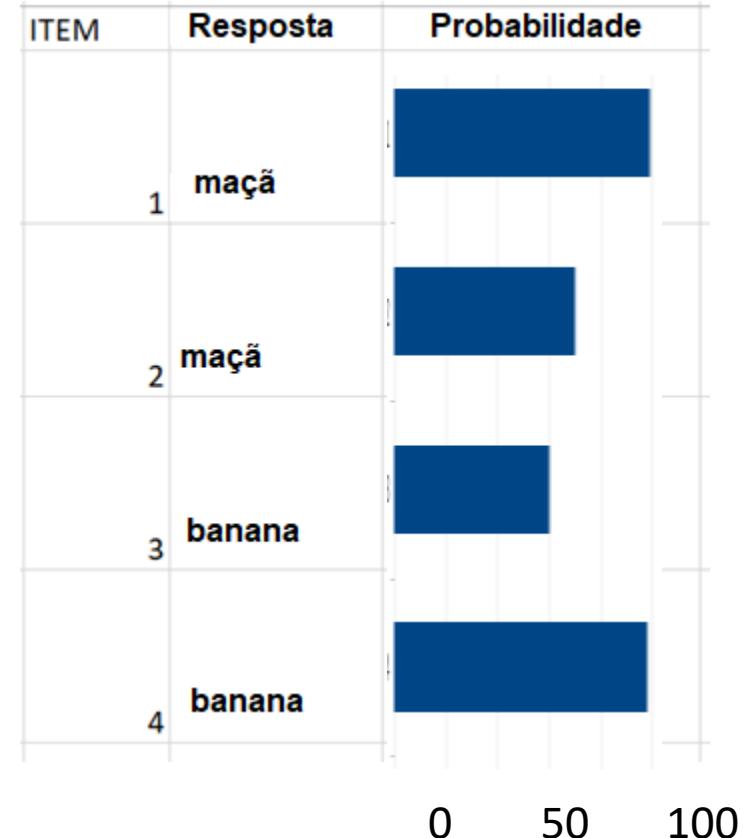
Modelo entrenado
color-dureza



Entrenamiento



ITEM	Cor	Dureza
1	● azul	🔨🔨🔨
2	● amarelo	🔨🔨🔨
3	● verde	🔨🔨🔨
4	● amarelo	Pussy



Como NO usar el modelo?



ITEM	Cor	Dureza
1		



ITEM	Comprimento	Dureza
1		



ITEM	Comprimento	Altura
1		

Modelo entrenado
color-dureza

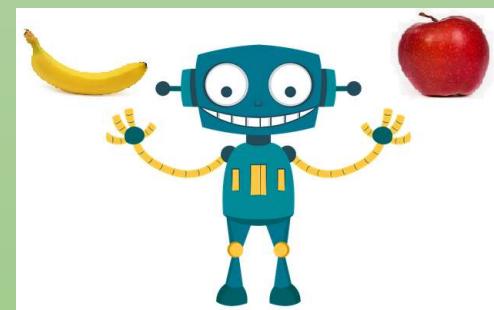


Tabla de confusión

Predicted Class	Actual Class
No Cancer	No Cancer
Has Cancer	Has Cancer
No Cancer	No Cancer
No Cancer	No Cancer
No Cancer	Has Cancer
Has Cancer	Has Cancer
No Cancer	No Cancer
Has Cancer	No Cancer
No Cancer	No Cancer
No Cancer	No Cancer
Has Cancer	No Cancer
No Cancer	No Cancer



Predicted Class
↑
Actual Class

Observación	Predicción	
	Positivos	Negativos
Positivos	Verdaderos Positivos (VP)	Falsos Negativos (FN)
Negativos	Falsos Positivos (FP)	Verdaderos Negativos (VN)

Observación	Predicción	
	Positivos	Negativos
Positivos	Verdaderos Positivos (VP)	Falsos Negativos (FN)
Negativos	Falsos Positivos (FP)	Verdaderos Negativos (VN)

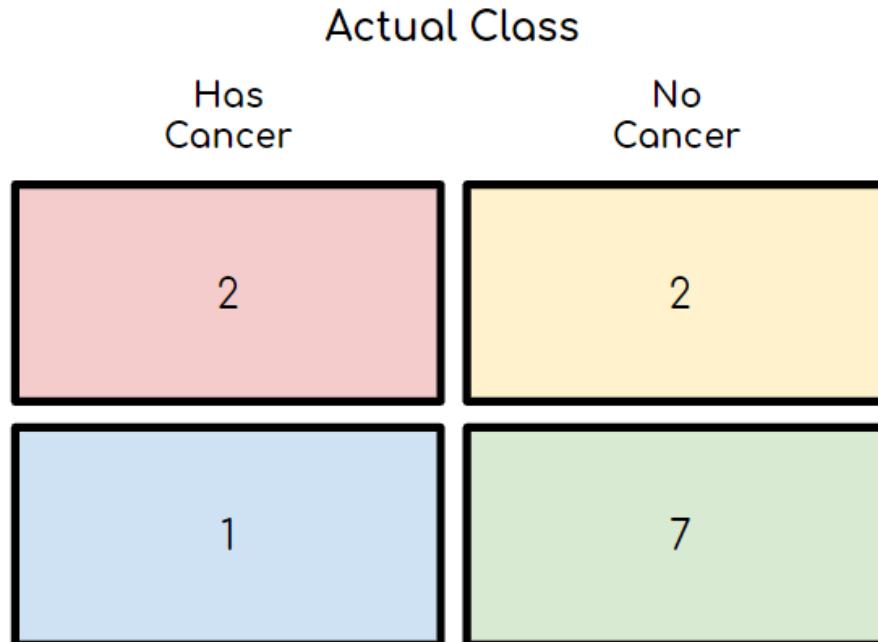


Tabla de confusión

Observación	Predicción	
	Positivos	Negativos
Positivos	Positivos	Verdaderos Positivos (VP)
	Negativos	Falsos Negativos (FN)
Negativos	Positivos	Falsos Positivos (FP)
	Negativos	Verdaderos Negativos (VN)

Observación	Predicción	
	Positivos	Negativos
Positivos	Positivos	Verdaderos Positivos (VP)
	Negativos	Falsos Negativos (FN)
Negativos	Positivos	Falsos Positivos (FP)
	Negativos	Verdaderos Negativos (VN)

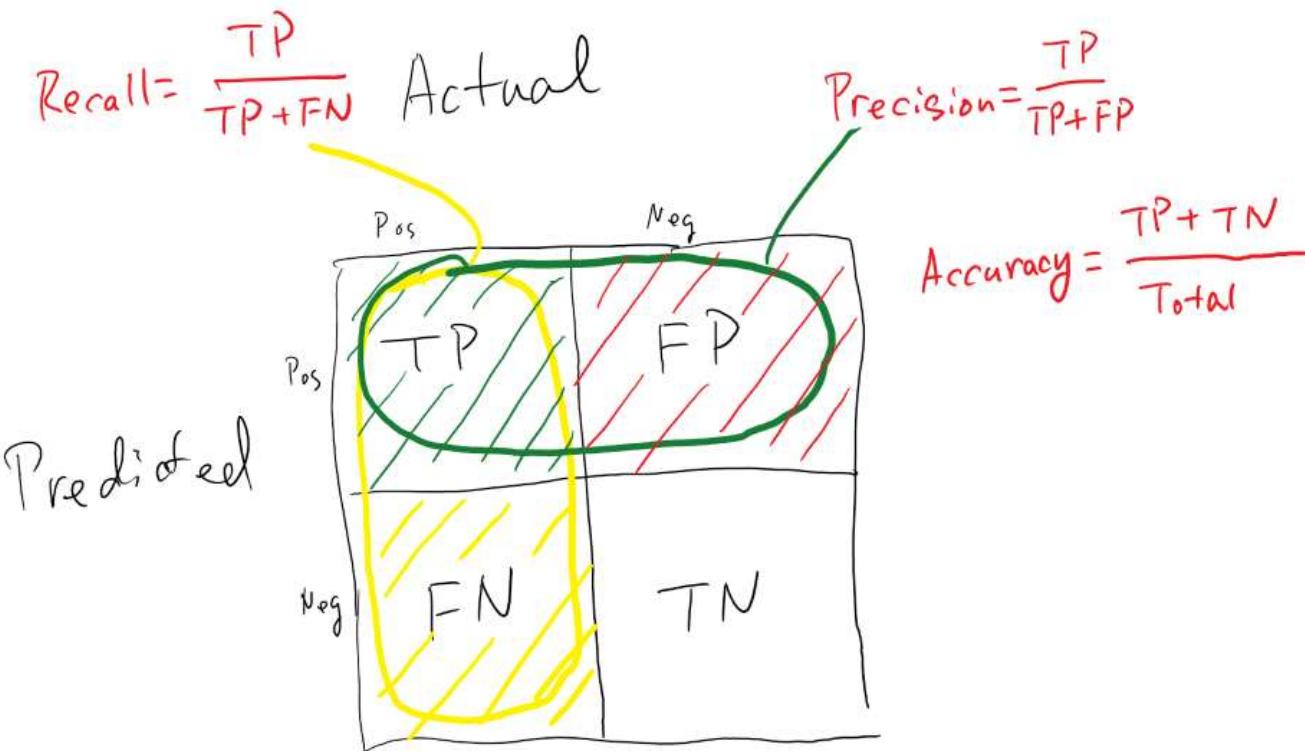
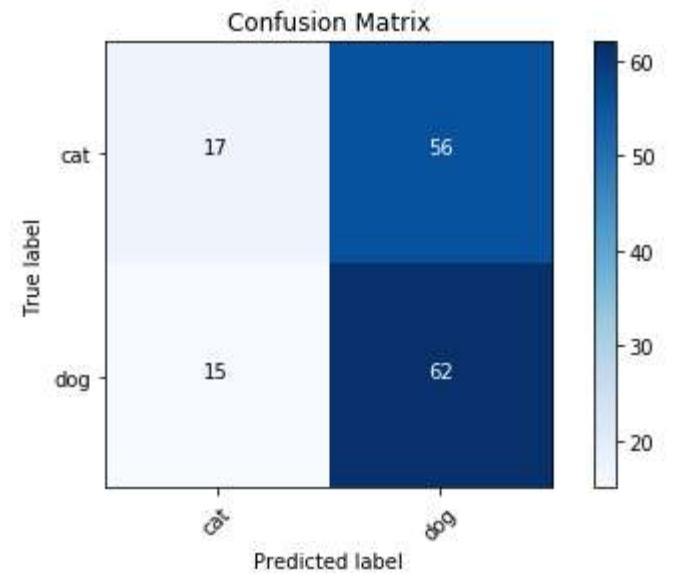
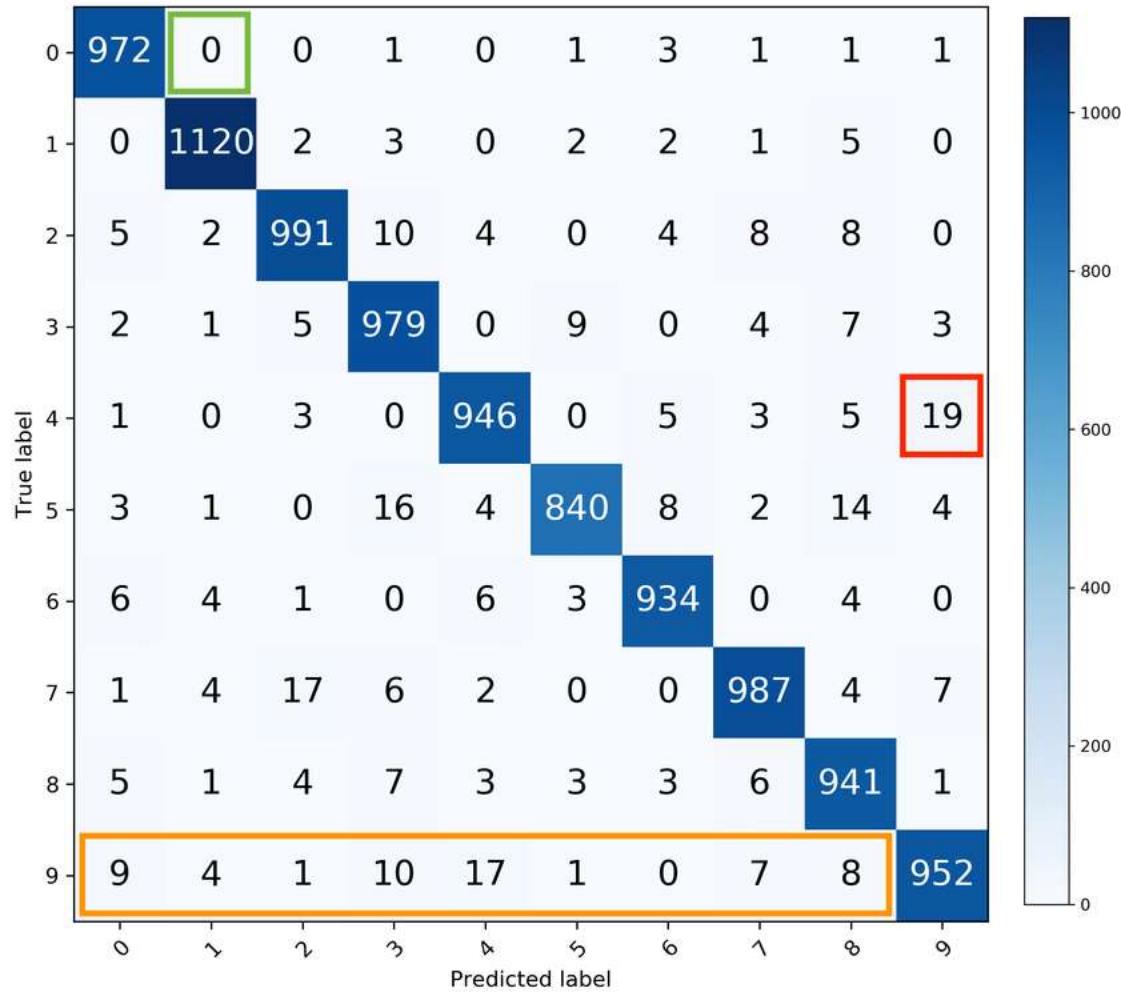
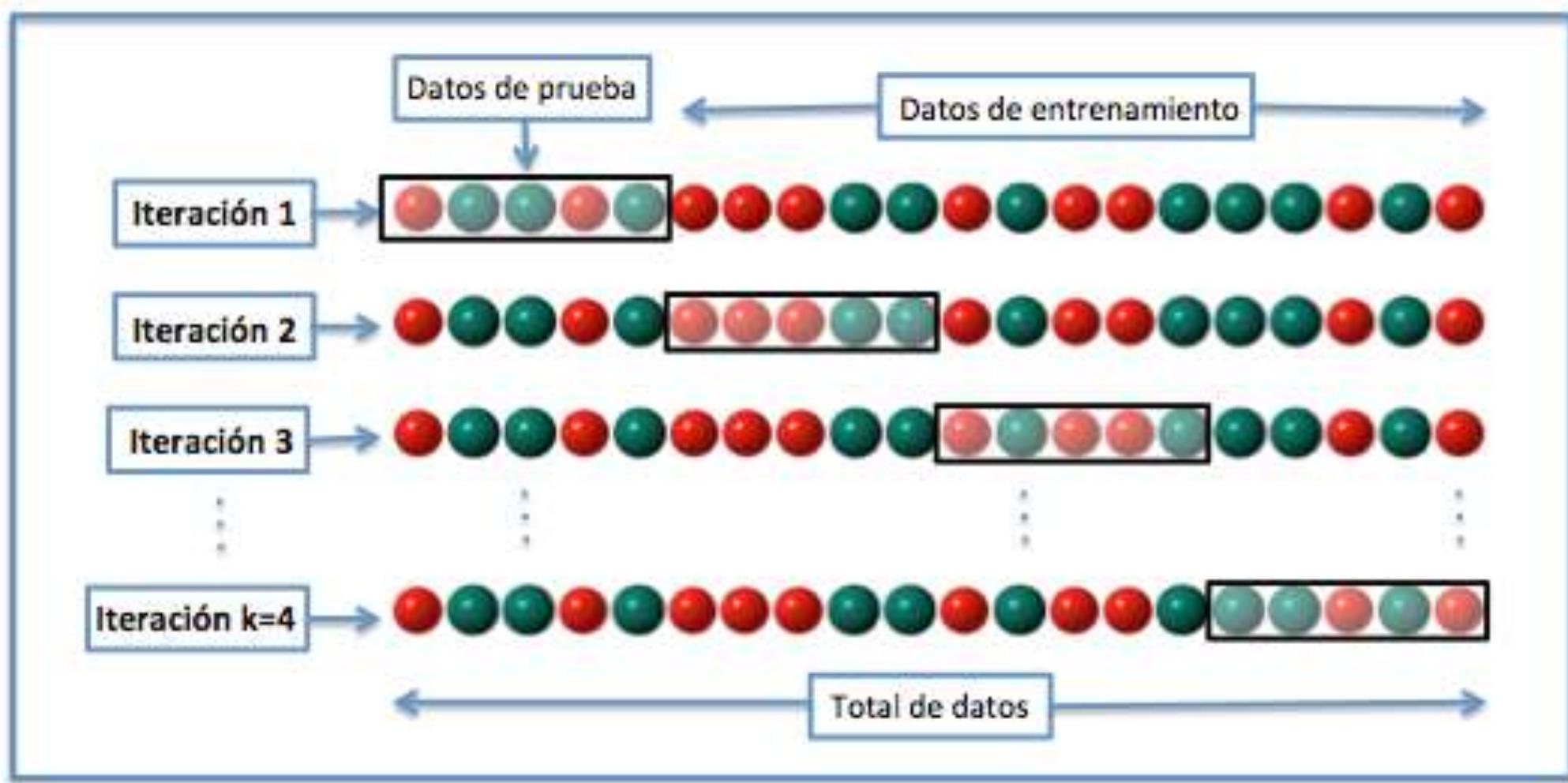


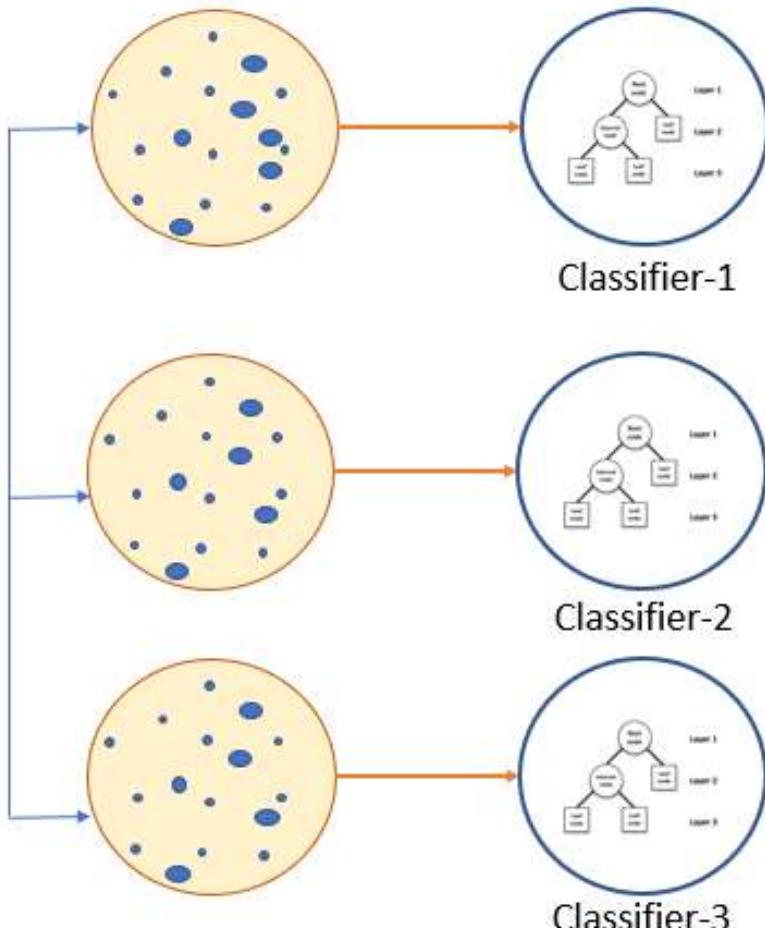
Tabla de confusión



Validación cruzada

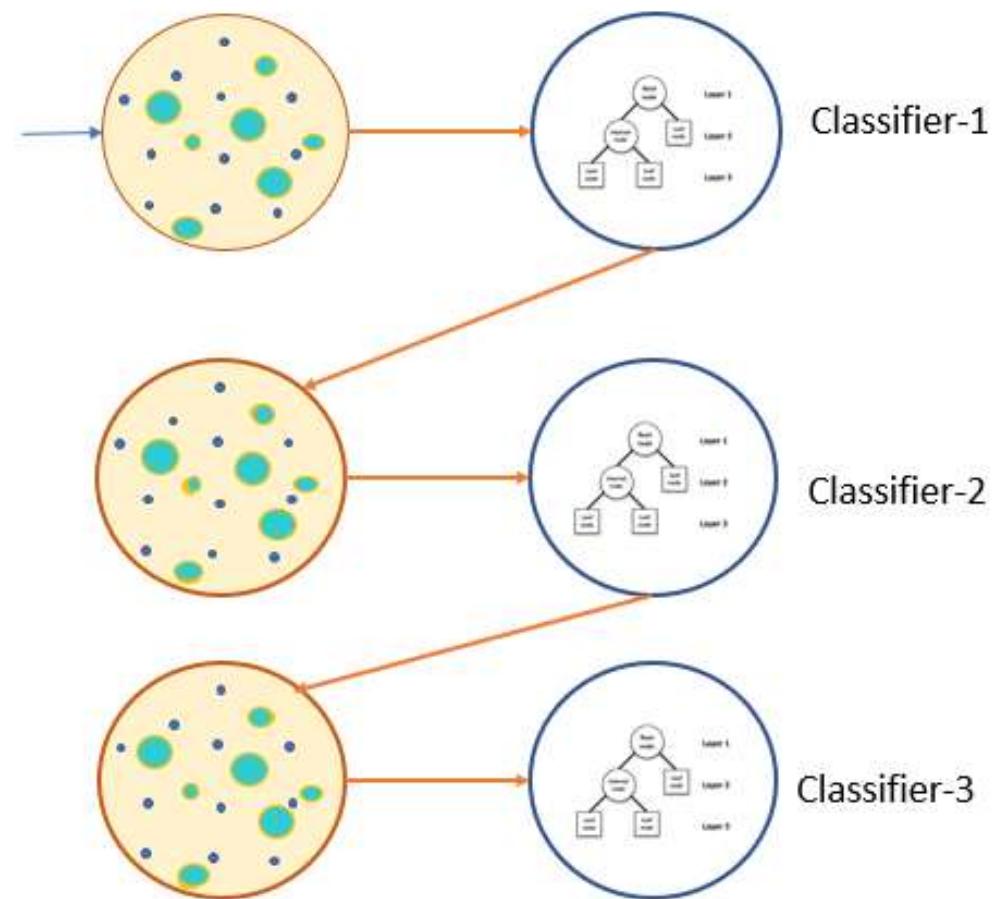


Bagging

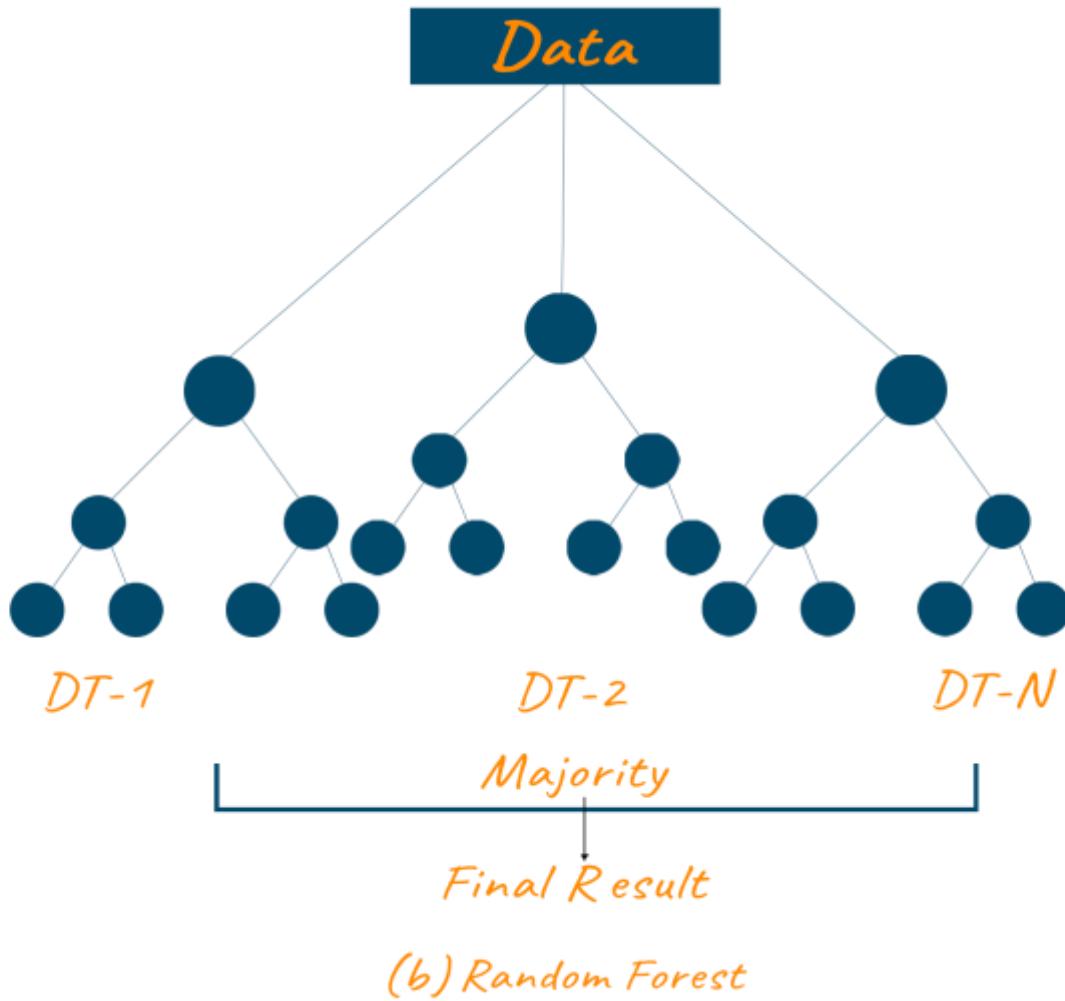


Parallel

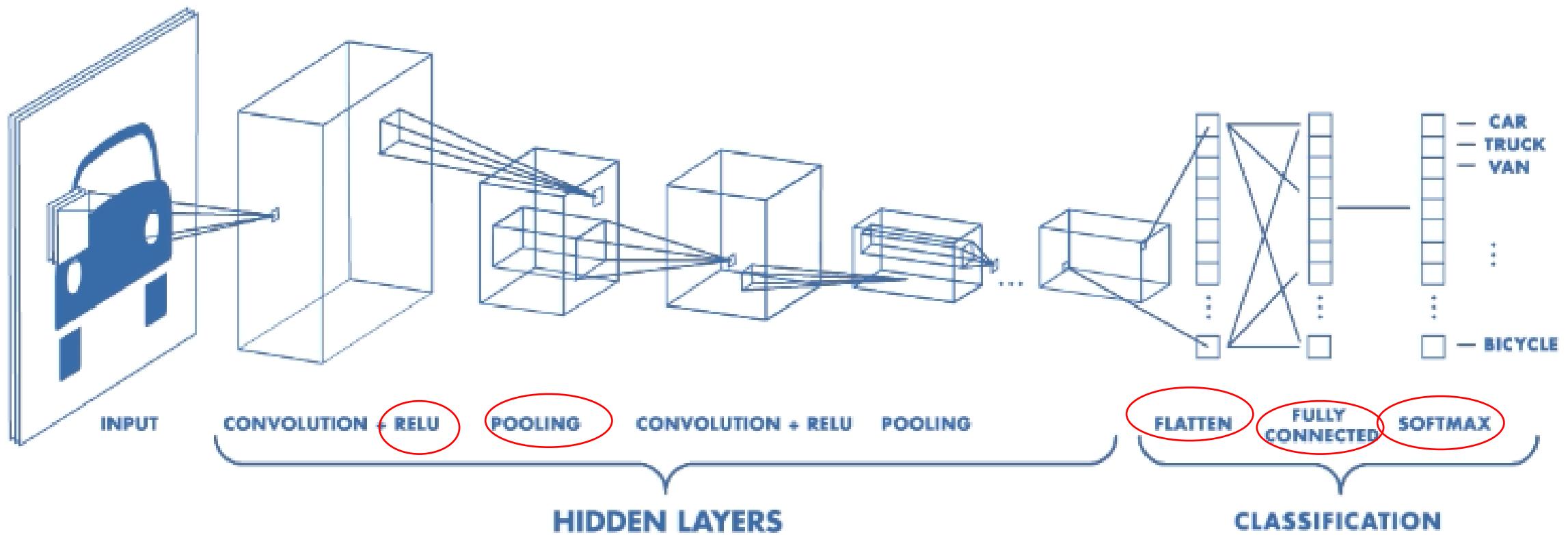
Boosting



Sequential

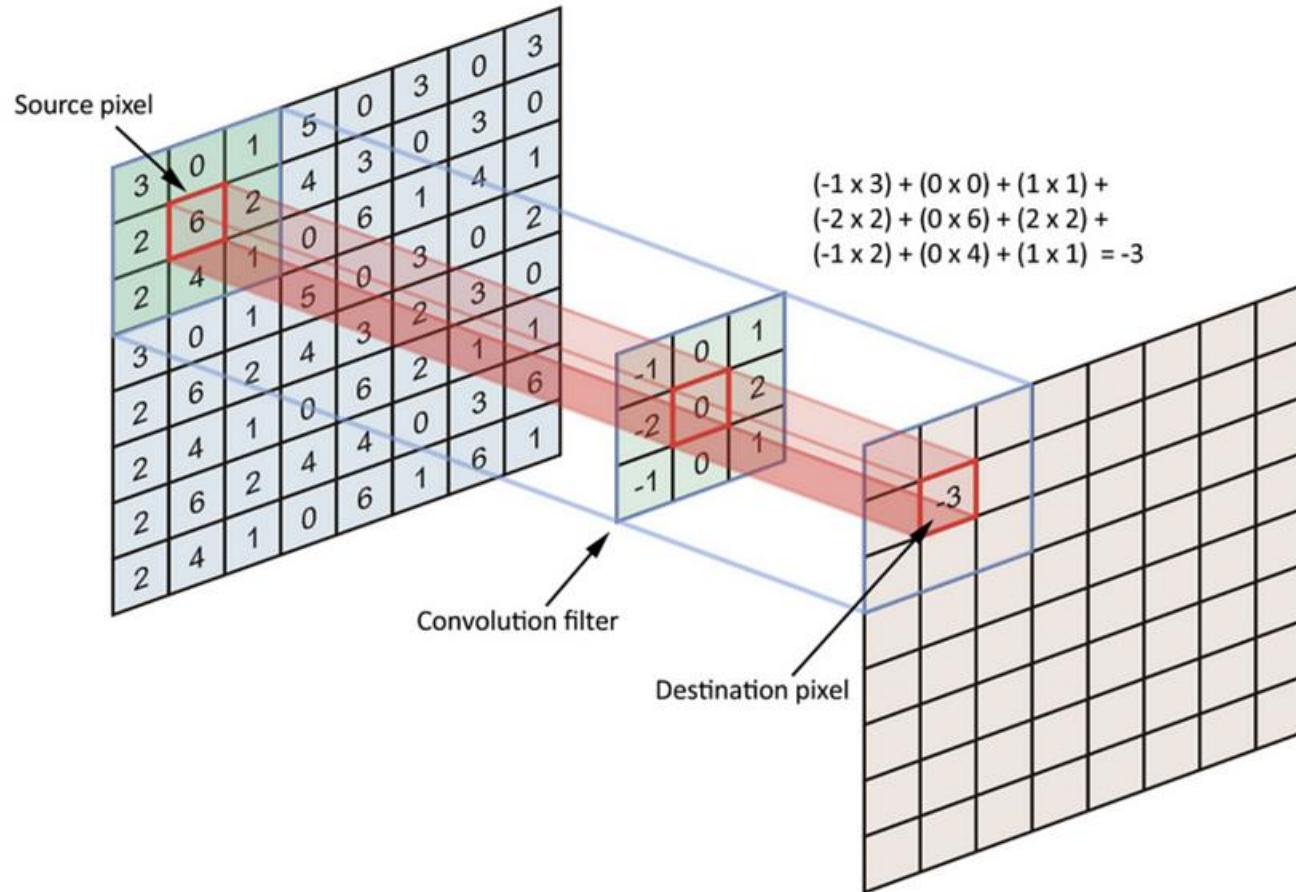


Previamente ...



Aula 3

Previamente ...



Fuente: <https://towardsdatascience.com/simple-introduction-to-convolutional-neural-networks-cdf8d3077bac>

Previamente ...

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

image

1	0	1
0	1	0
1	0	1

operador de convolución
(filtro, kernel)

1 _{x1}	1 _{x0}	1 _{x1}	0	0
0 _{x0}	1 _{x1}	1 _{x0}	1	0
0 _{x1}	0 _{x0}	1 _{x1}	1	1
0	0	1	1	0
0	1	1	0	0

Image

4		

Convolved
Feature

$$\begin{array}{ccccc}
 1 & 1 & 1 & 0 & 0 \\
 0 & 1 & 1 & 1 & 0 \\
 0 & 0 & 1 & 1 & 1 \\
 0 & 0 & 1 & 1 & 0 \\
 0 & 1 & 1 & 0 & 0
 \end{array} \times \otimes \begin{array}{ccc}
 1 & 0 & 1 \\
 0 & 1 & 0 \\
 1 & 0 & 1
 \end{array} = \begin{array}{ccc}
 4 & 3 & 4 \\
 2 & 4 & 3 \\
 2 & 3 & 4
 \end{array}$$

filter

feature map

Operación de convolución

- Dependiendo del filtro, se pueden extraer diferentes características de una misma imagen.



CNNs **aprende** los
filtros
durante el
entrenamiento!

Detección de bordes: ejercicio1

$$\begin{matrix} & \begin{matrix} 6 \times 6 \\ \hline \text{Input} \end{matrix} \\ \begin{matrix} 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \end{matrix} & \begin{matrix} * \\ = \end{matrix} & \begin{matrix} 3 \times 3 \\ \hline \text{Kernel} \end{matrix} \\ \begin{matrix} & \begin{matrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{matrix} \\ & \begin{matrix} \square & \square & \square \\ \square & \square & \square \\ \square & \square & \square \end{matrix} \end{matrix} & & \begin{matrix} & \begin{matrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{matrix} \\ & \begin{matrix} \square & \square & \square \\ \square & \square & \square \\ \square & \square & \square \end{matrix} \end{matrix} \\ \begin{matrix} & \begin{matrix} 0 & 0 & 0 & 10 & 10 & 10 \\ 0 & 0 & 0 & 10 & 10 & 10 \\ 0 & 0 & 0 & 10 & 10 & 10 \\ 0 & 0 & 0 & 10 & 10 & 10 \\ 0 & 0 & 0 & 10 & 10 & 10 \\ 0 & 0 & 0 & 10 & 10 & 10 \end{matrix} \\ & \begin{matrix} \square & \square \\ \square & \square \end{matrix} \end{matrix} & \begin{matrix} * \\ = \end{matrix} & \begin{matrix} & \begin{matrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{matrix} \\ & \begin{matrix} \square & \square & \square \\ \square & \square & \square \\ \square & \square & \square \end{matrix} \end{matrix} \end{matrix}$$

Andrew Ng

Detección de bordes

$$\begin{array}{c} \text{6} \times \text{6} \\ \begin{array}{|c|c|c|c|c|c|}\hline 10 & 10 & 10 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 \\ \hline \end{array} \end{array} * \begin{array}{c} \text{3} \times \text{3} \\ \begin{array}{|c|c|c|}\hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline \end{array} \end{array} = \begin{array}{|c|c|c|c|c|c|}\hline 0 & 0 & 0 & 10 & 10 & 10 \\ \hline 0 & 0 & 0 & 10 & 10 & 10 \\ \hline 0 & 0 & 0 & 10 & 10 & 10 \\ \hline 0 & 0 & 0 & 10 & 10 & 10 \\ \hline 0 & 0 & 0 & 10 & 10 & 10 \\ \hline 0 & 0 & 0 & 10 & 10 & 10 \\ \hline \end{array}$$

$$* \begin{array}{c} \text{3} \times \text{3} \\ \begin{array}{|c|c|c|}\hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline \end{array} \end{array} = \begin{array}{|c|c|c|c|c|c|}\hline 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ \hline \end{array}$$

Andrew Ng

Detección de bordes

$$\begin{array}{c} \text{6} \times \text{6} \\ \begin{array}{|c|c|c|c|c|c|}\hline 10 & 10 & 10 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 \\ \hline \end{array} \end{array} * \begin{array}{c} \text{3} \times \text{3} \\ \begin{array}{|c|c|c|}\hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline \end{array} \end{array} = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline \end{array}$$

$$* \begin{array}{c} \text{3} \times \text{3} \\ \begin{array}{|c|c|c|}\hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline \end{array} \end{array} = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline \end{array}$$

Andrew Ng

Detección de bordes

$$\begin{array}{c} \text{6 x 6} \\ \begin{array}{|c|c|c|c|c|c|} \hline 10 & 10 & 10 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 \\ \hline \end{array} \end{array} * \begin{array}{c} \text{3 x 3} \\ \begin{array}{|c|c|c|} \hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline \end{array} \end{array} = \begin{array}{|c|c|c|c|c|c|} \hline 0 & 0 & 0 & 10 & 10 & 10 \\ \hline 0 & 0 & 0 & 10 & 10 & 10 \\ \hline 0 & 0 & 0 & 10 & 10 & 10 \\ \hline 0 & 0 & 0 & 10 & 10 & 10 \\ \hline 0 & 0 & 0 & 10 & 10 & 10 \\ \hline 0 & 0 & 0 & 10 & 10 & 10 \\ \hline \end{array}$$

$$* \begin{array}{c} \text{3 x 3} \\ \begin{array}{|c|c|c|} \hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline \end{array} \end{array} = \begin{array}{|c|c|c|c|c|c|} \hline 0 & 0 & 0 & 10 & 10 & 10 \\ \hline 0 & 0 & 0 & 10 & 10 & 10 \\ \hline 0 & 0 & 0 & 10 & 10 & 10 \\ \hline 0 & 0 & 0 & 10 & 10 & 10 \\ \hline 0 & 0 & 0 & 10 & 10 & 10 \\ \hline 0 & 0 & 0 & 10 & 10 & 10 \\ \hline \end{array}$$

Andrew Ng

Detección de bordes

$$\begin{array}{c} \text{6 x 6} \\ \begin{array}{|c|c|c|c|c|c|} \hline 10 & 10 & 10 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 \\ \hline \end{array} \end{array} * \begin{array}{c} \text{3 x 3} \\ \begin{array}{|c|c|c|} \hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline \end{array} \end{array} = \begin{array}{|c|c|c|c|c|c|} \hline 0 & 0 & 0 & 10 & 10 & 10 \\ \hline 0 & 0 & 0 & 10 & 10 & 10 \\ \hline 0 & 0 & 0 & 10 & 10 & 10 \\ \hline 0 & 0 & 0 & 10 & 10 & 10 \\ \hline 0 & 0 & 0 & 10 & 10 & 10 \\ \hline 0 & 0 & 0 & 10 & 10 & 10 \\ \hline \end{array}$$

$$* \begin{array}{c} \text{3 x 3} \\ \begin{array}{|c|c|c|} \hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline \end{array} \end{array} = \begin{array}{|c|c|c|c|c|c|} \hline 0 & 0 & 0 & 10 & 10 & 10 \\ \hline 0 & 0 & 0 & 10 & 10 & 10 \\ \hline 0 & 0 & 0 & 10 & 10 & 10 \\ \hline 0 & 0 & 0 & 10 & 10 & 10 \\ \hline 0 & 0 & 0 & 10 & 10 & 10 \\ \hline 0 & 0 & 0 & 10 & 10 & 10 \\ \hline \end{array}$$

Andrew Ng

Detección de bordes

$$\begin{array}{c} \text{6} \times \text{6} \\ \begin{array}{|c|c|c|c|c|c|} \hline 10 & 10 & 10 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 \\ \hline \end{array} \end{array} * \begin{array}{c} \text{3} \times \text{3} \\ \begin{array}{|c|c|c|} \hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline \end{array} \end{array} = \begin{array}{|c|c|c|c|c|c|} \hline 0 & 0 & 0 & 10 & 10 & 10 \\ \hline 0 & 0 & 0 & 10 & 10 & 10 \\ \hline 0 & 0 & 0 & 10 & 10 & 10 \\ \hline 0 & 0 & 0 & 10 & 10 & 10 \\ \hline 0 & 0 & 0 & 10 & 10 & 10 \\ \hline 0 & 0 & 0 & 10 & 10 & 10 \\ \hline \end{array}$$

$$* \begin{array}{c} \text{3} \times \text{3} \\ \begin{array}{|c|c|c|} \hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline \end{array} \end{array} = \begin{array}{|c|c|c|c|c|c|} \hline 0 & 0 & 0 & 10 & 10 & 10 \\ \hline 0 & 0 & 0 & 10 & 10 & 10 \\ \hline 0 & 0 & 0 & 10 & 10 & 10 \\ \hline 0 & 0 & 0 & 10 & 10 & 10 \\ \hline 0 & 0 & 0 & 10 & 10 & 10 \\ \hline 0 & 0 & 0 & 10 & 10 & 10 \\ \hline \end{array}$$

Andrew Ng

Detección de bordes

$$\begin{matrix} & \begin{matrix} 6 \times 6 \\ \begin{matrix} 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \end{matrix} \end{matrix} & * & \begin{matrix} 3 \times 3 \\ \begin{matrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{matrix} \\ \begin{matrix} \square & \square & \square \end{matrix} \end{matrix} & = & \begin{matrix} \begin{matrix} 0 & 0 & 0 & 10 & 10 & 10 \\ 0 & 0 & 0 & 10 & 10 & 10 \\ 0 & 0 & 0 & 10 & 10 & 10 \\ 0 & 0 & 0 & 10 & 10 & 10 \\ 0 & 0 & 0 & 10 & 10 & 10 \\ 0 & 0 & 0 & 10 & 10 & 10 \end{matrix} \\ \begin{matrix} \square & \square \end{matrix} \end{matrix} \\ * & \begin{matrix} 3 \times 3 \\ \begin{matrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{matrix} \\ \begin{matrix} \square & \square & \square \end{matrix} \end{matrix} & = & \end{matrix}$$

Andrew Ng

Detección de bordes

$$\begin{array}{c} \text{6} \times 6 \\ \begin{array}{|c|c|c|c|c|c|} \hline 10 & 10 & 10 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 \\ \hline \end{array} \end{array} * \begin{array}{c} \text{3} \times 3 \\ \begin{array}{|c|c|c|} \hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline \end{array} \end{array} = \begin{array}{c} \text{4} \times 4 \\ \begin{array}{|c|c|c|c|} \hline & & & \\ \hline \end{array} \end{array}$$

$$\begin{array}{c} \text{6} \times 6 \\ \begin{array}{|c|c|c|c|c|c|} \hline 0 & 0 & 0 & 10 & 10 & 10 \\ \hline 0 & 0 & 0 & 10 & 10 & 10 \\ \hline 0 & 0 & 0 & 10 & 10 & 10 \\ \hline 0 & 0 & 0 & 10 & 10 & 10 \\ \hline 0 & 0 & 0 & 10 & 10 & 10 \\ \hline 0 & 0 & 0 & 10 & 10 & 10 \\ \hline \end{array} \end{array} * \begin{array}{c} \text{3} \times 3 \\ \begin{array}{|c|c|c|} \hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline \end{array} \end{array} = \begin{array}{c} \text{4} \times 4 \\ \begin{array}{|c|c|c|c|} \hline & & & \\ \hline \end{array} \end{array}$$

Andrew Ng

Detección de bordes

$$\begin{array}{|c|c|c|c|c|c|c|} \hline 10 & 10 & 10 & 0 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 & 0 \\ \hline \end{array} * \begin{array}{|c|c|c|} \hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline \end{array} = \begin{array}{|c|c|c|c|c|c|c|} \hline 0 & 30 & 30 & 0 & & & \\ \hline 0 & 30 & 30 & 0 & & & \\ \hline 0 & 30 & 30 & 0 & & & \\ \hline 0 & 30 & 30 & 0 & & & \\ \hline \end{array}$$

$$\begin{array}{|c|c|c|c|c|c|c|} \hline 0 & 0 & 0 & 10 & 10 & 10 & 10 \\ \hline 0 & 0 & 0 & 10 & 10 & 10 & 10 \\ \hline 0 & 0 & 0 & 10 & 10 & 10 & 10 \\ \hline 0 & 0 & 0 & 10 & 10 & 10 & 10 \\ \hline 0 & 0 & 0 & 10 & 10 & 10 & 10 \\ \hline 0 & 0 & 0 & 10 & 10 & 10 & 10 \\ \hline \end{array} * \begin{array}{|c|c|c|} \hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline \end{array} = \begin{array}{|c|c|c|c|c|c|c|} \hline 0 & -30 & -30 & 0 & & & \\ \hline 0 & -30 & -30 & 0 & & & \\ \hline 0 & -30 & -30 & 0 & & & \\ \hline 0 & -30 & -30 & 0 & & & \\ \hline \end{array}$$

Andrew Ng

Detección de bordes

1	0	-1
1	0	-1
1	0	-1

Vertical

1	1	1
0	0	0
-1	-1	-1

Horizontal

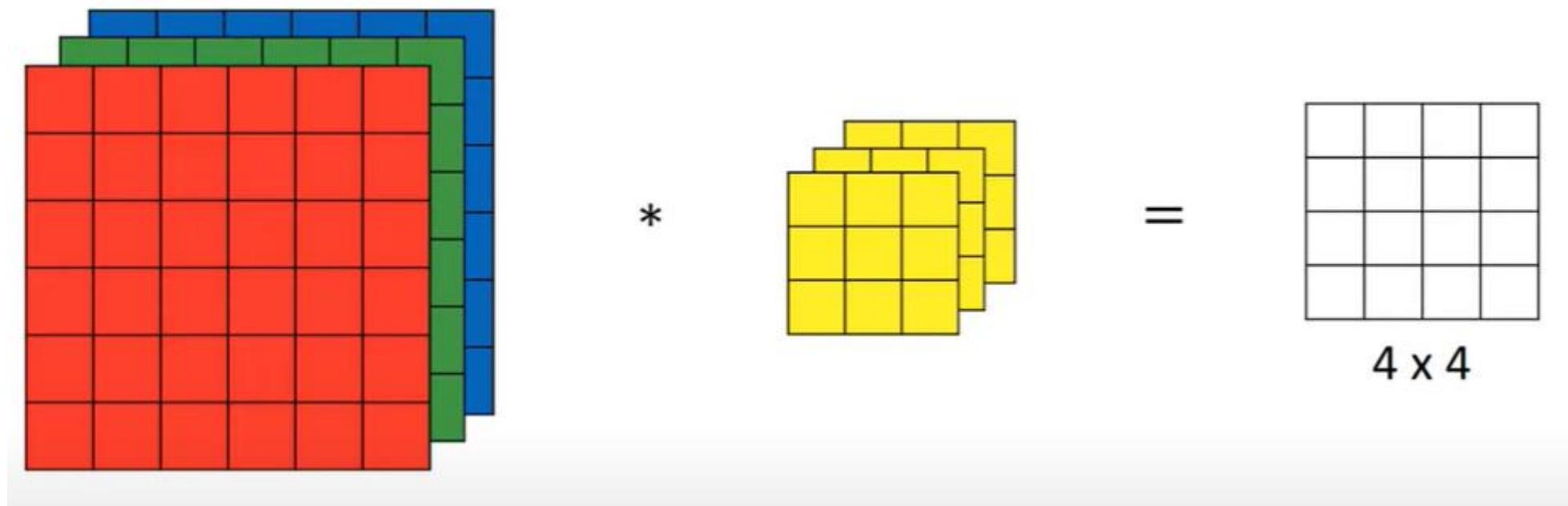
Detección de bordes: ejercicio 2

$$\begin{array}{c} A \\ \begin{array}{|c|c|c|c|c|c|c|} \hline 10 & 10 & 10 & 0 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 & 0 \\ \hline \end{array} \end{array} * \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -1 & -1 \\ \hline \end{array} =$$

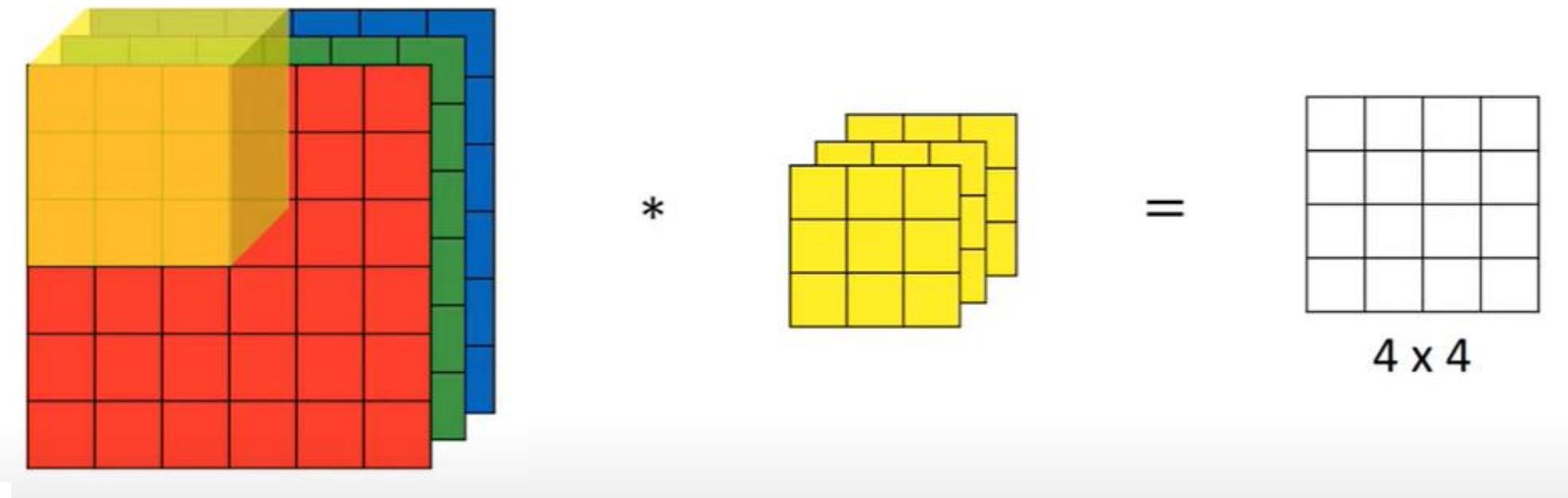
$$\begin{array}{c} B \\ \begin{array}{|c|c|c|c|c|c|c|} \hline 10 & 10 & 10 & 10 & 10 & 10 & 10 \\ \hline 10 & 10 & 10 & 10 & 10 & 10 & 10 \\ \hline 10 & 10 & 10 & 10 & 10 & 10 & 10 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline \end{array} \end{array} * \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -1 & -1 \\ \hline \end{array} =$$

Andrew Ng

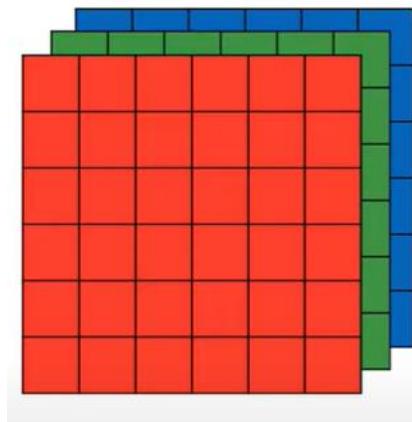
Convolución en imágenes RGB



Convolución en imágenes RGB



Convolución en imágenes RGB



Filtro
bordes
horizontales

*

=

4×4

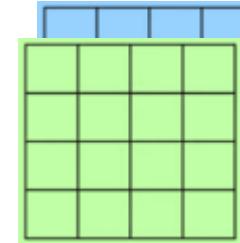
Filtro
bordes
verticales

*

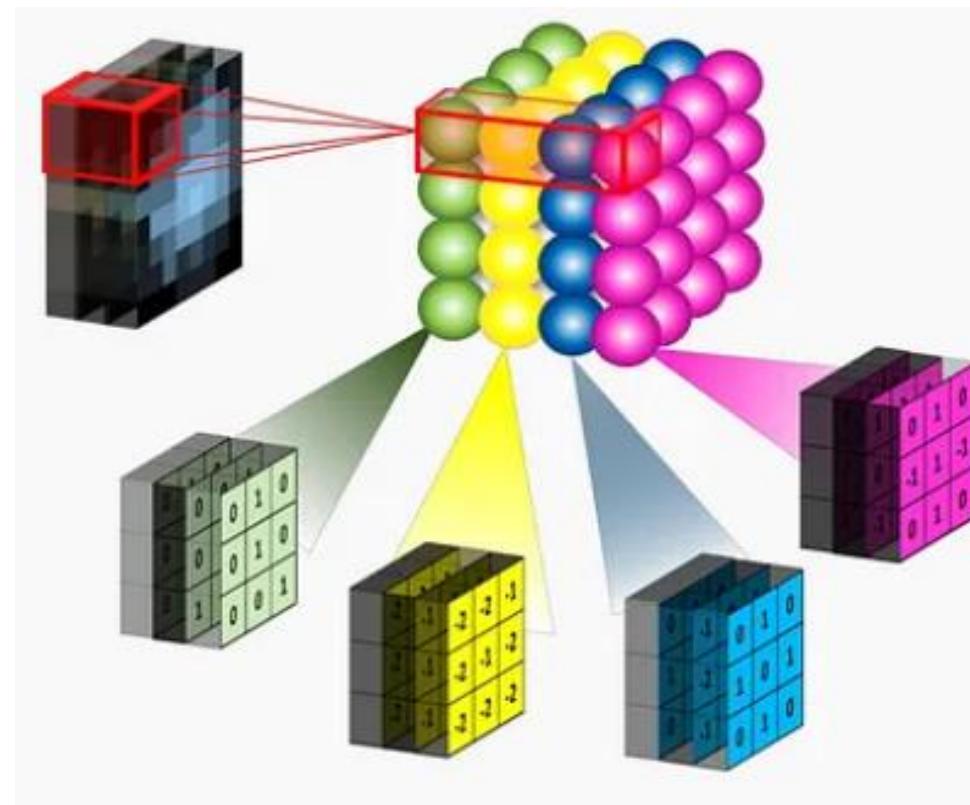
$3 \times 3 \times 3$

=

4×4



Convolución en imágenes RGB



Fuente: Curso Deep Learning, Andrew Ng (<https://cs230.stanford.edu/>)

Generalización de los filtros

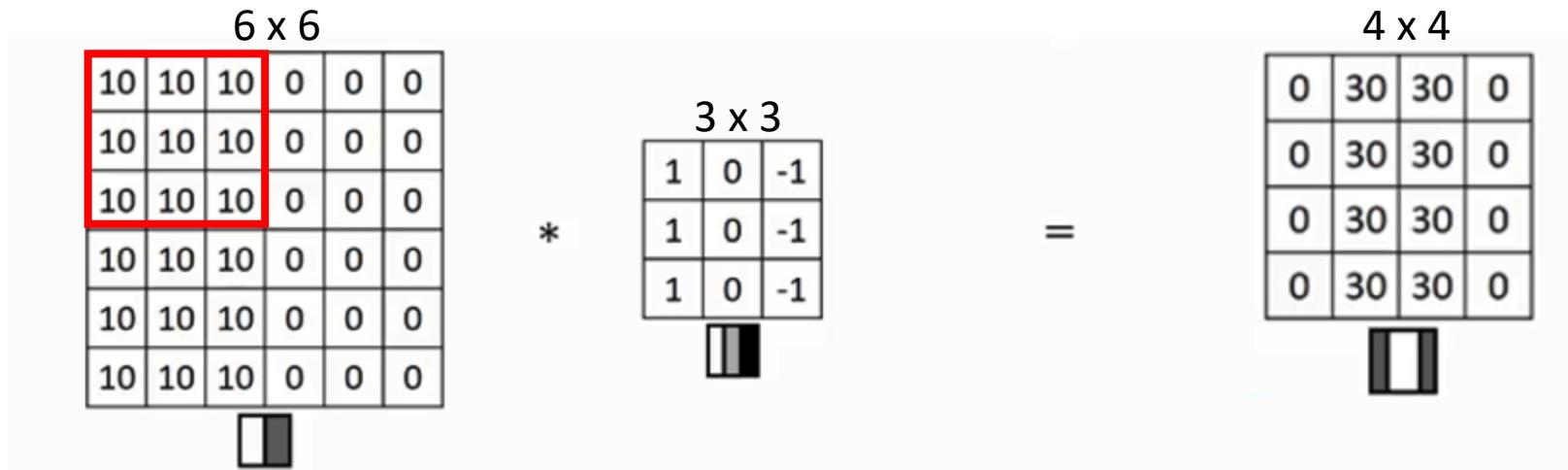


*

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

CNNs **aprende** los
filtros
durante el
entrenamiento!

Padding

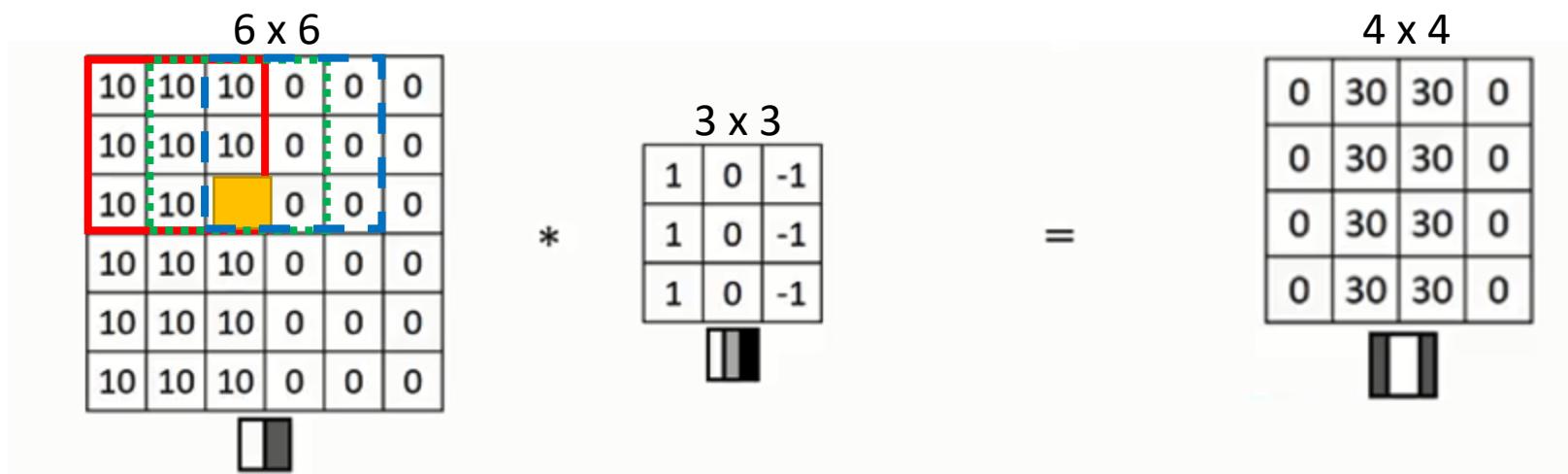


$$d = n-f+1$$

n: dimensión de la imagen

f: dimensión del filtro

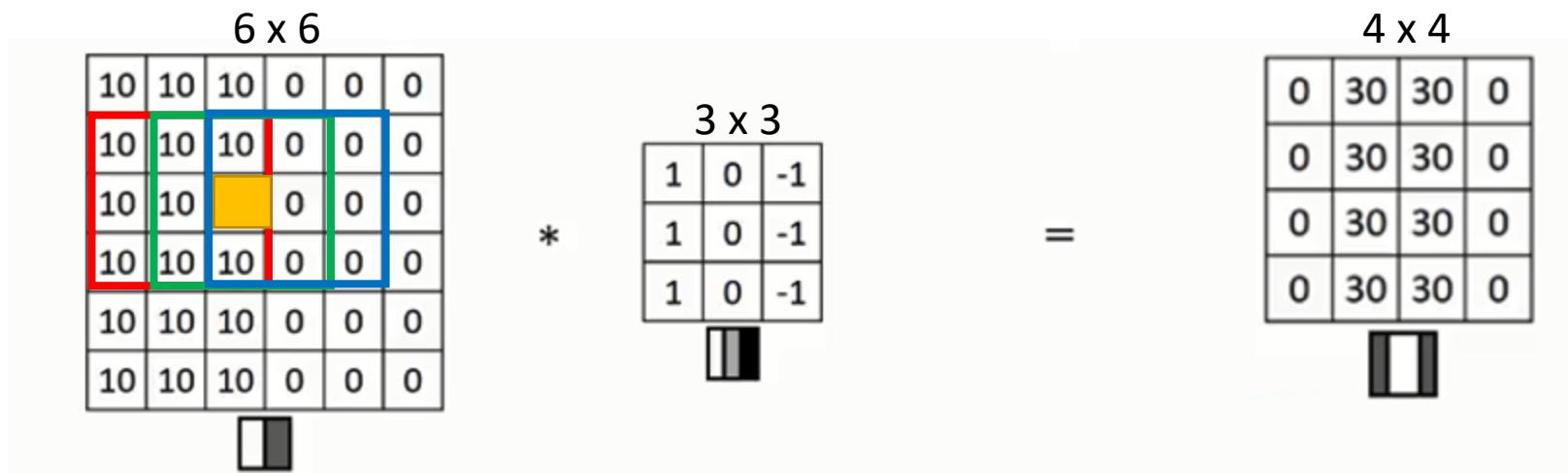
Padding



$$d = n-f+1$$

n: dimensión de la imagen
F: dimensión del filtro

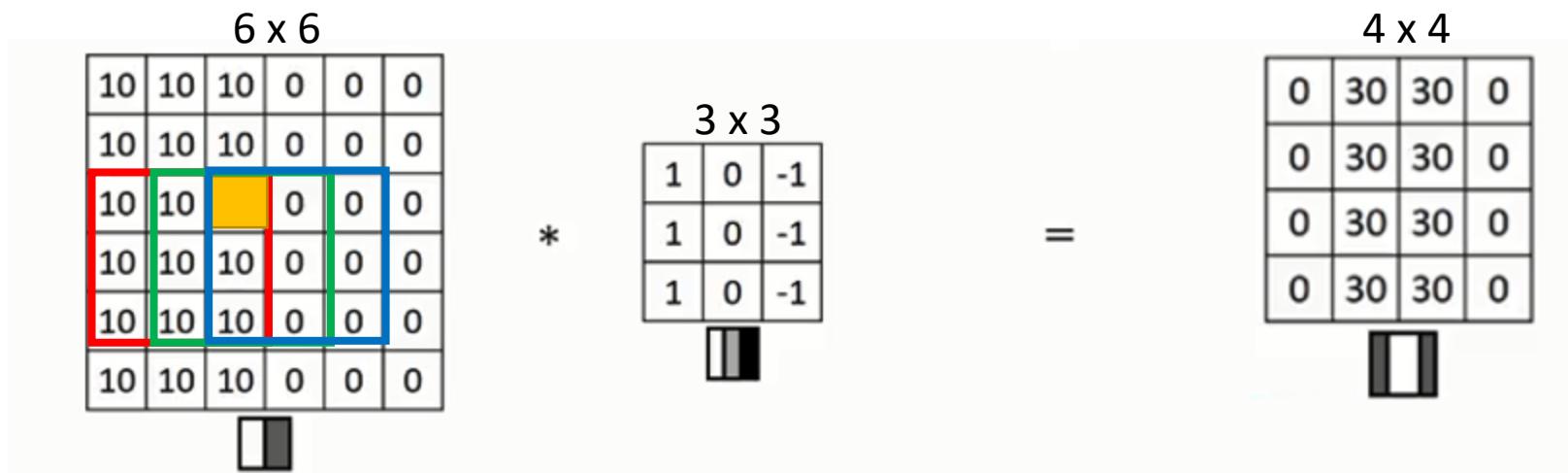
Padding



$$d = n-f+1$$

n: dimensión de la imagen
F: dimensión del filtro

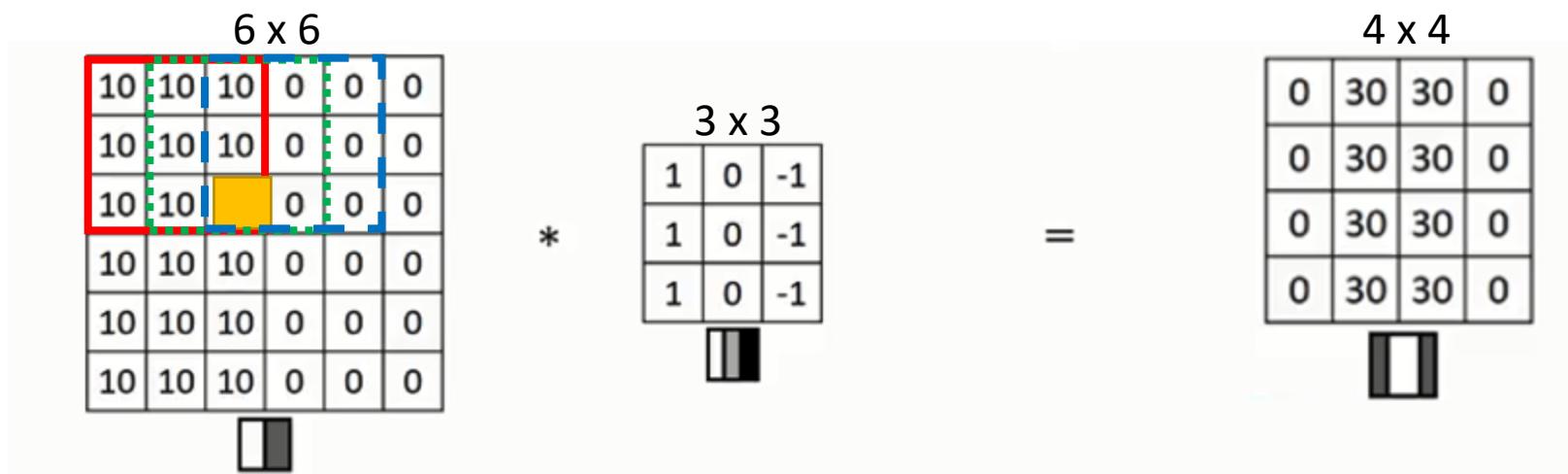
Padding



$$d = n-f+1$$

n: dimensión de la imagen
F: dimensión del filtro

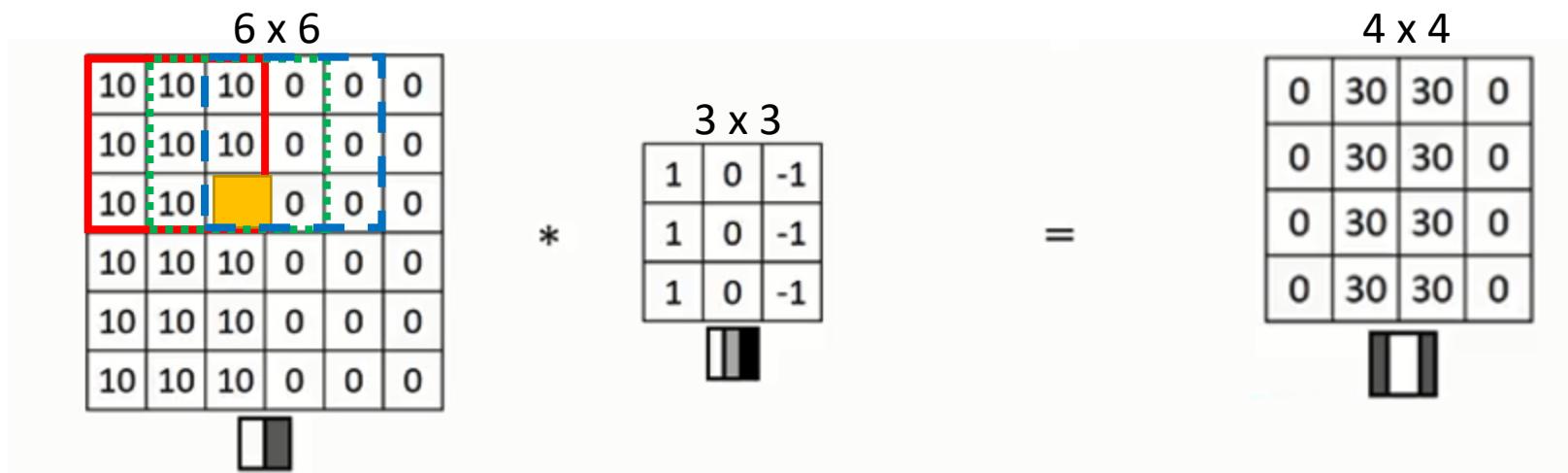
Padding



$$d = n-f+1$$

n: dimensión de la imagen
F: dimensión del filtro

Padding



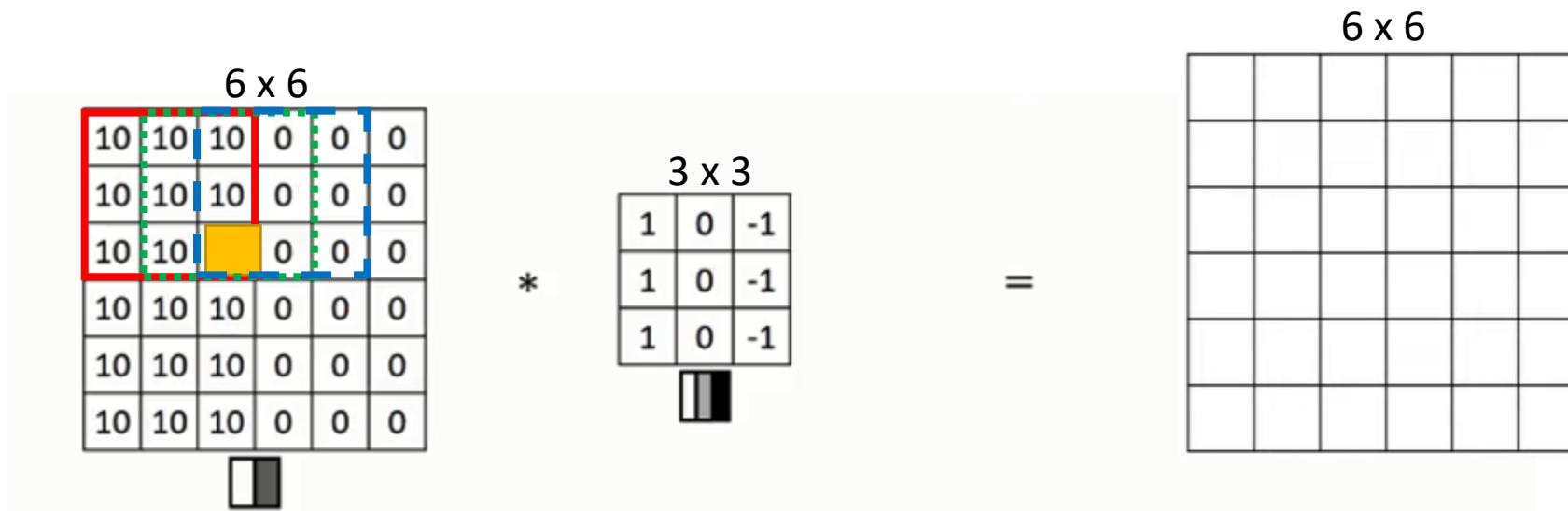
$$n-f+1+2p=n$$

$$2p=(f-1)$$

$$p=(f-1)/2$$

Fuente: Curso Deep Learning, Andrew Ng (<https://cs230.stanford.edu/>)

Padding



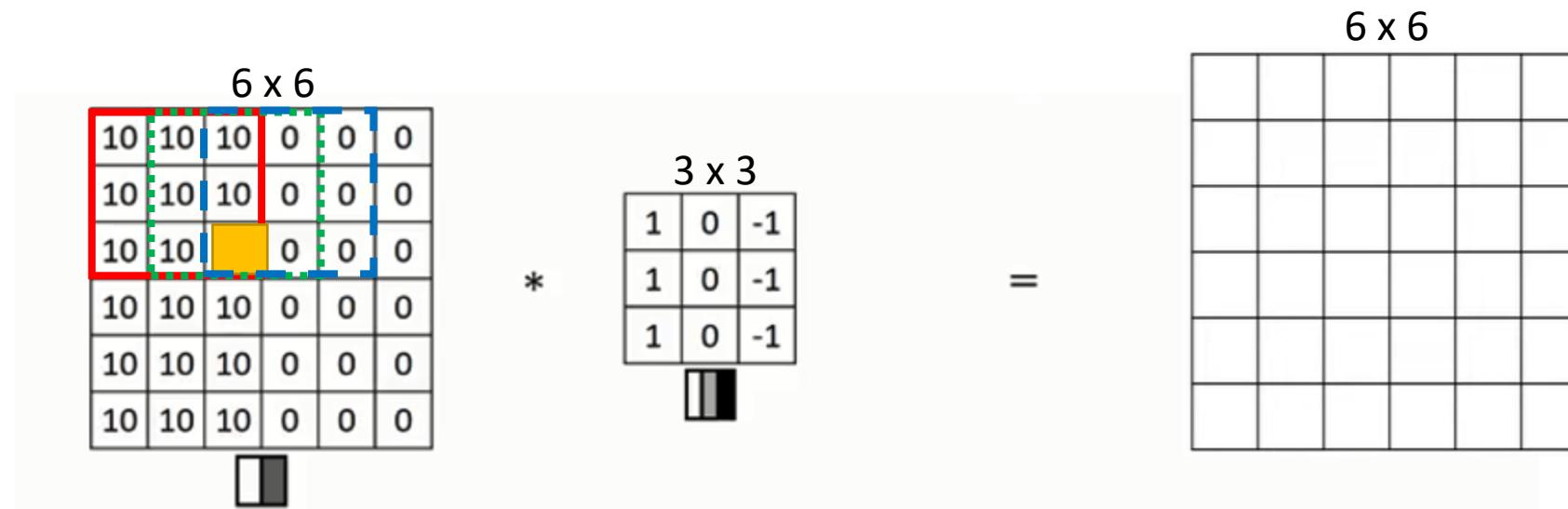
Padding = p

$$n-f+1+2p=n$$

$$2p=(f-1)$$

$$p=(f-1)/2$$

Padding



Padding = p

$$\begin{aligned}n-f+1+2p &= n \\2p &= (f-1) \\p &= (f-1)/2\end{aligned}$$

Stride o paso

2	3	7	4	6	2	9
6	6	9	8	7	4	3
3	4	8	3	8	9	7
7	8	3	6	6	3	4
4	2	1	8	3	4	6
3	2	4	1	9	8	3
0	1	3	9	2	1	4

*

3	4	4
1	0	2
-1	0	3

Stride = 2

Stride o paso

2	3	7	4	6	2	9
6	6	9	8	7	4	3
3	4	8	3	8	9	7
7	8	3	6	6	3	4
4	2	1	8	3	4	6
3	2	4	1	9	8	3
0	1	3	9	2	1	4

*

3	4	4
1	0	2
-1	0	3

Stride = 2

Stride o paso

2	3	7	4	6	2	9
6	6	9	8	7	4	3
3	4	8	3	8	9	7
7	8	3	6	6	3	4
4	2	1	8	3	4	6
3	2	4	1	9	8	3
0	1	3	9	2	1	4

*

3	4	4
1	0	2
-1	0	3

Stride = 2

Stride o paso

2	3	7	4	6	2	9
6	6	9	8	7	4	3
3	4	8	3	8	9	7
7	8	3	6	6	3	4
4	2	1	8	3	4	6
3	2	4	1	9	8	3
0	1	3	9	2	1	4

*

3	4	4
1	0	2
-1	0	3

$$d = 1 + (n+2p-f)/s$$

Padding = p

Stride = s

Stride o paso

2	3	7	4	6	2	9
6	6	9	8	7	4	3
3	4	8	3	8	9	7
7	8	3	6	6	3	4
4	2	1	8	3	4	6
3	2	4	1	9	8	3
0	1	3	9	2	1	4

*

3	4	4
1	0	2
-1	0	3

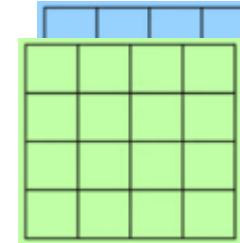
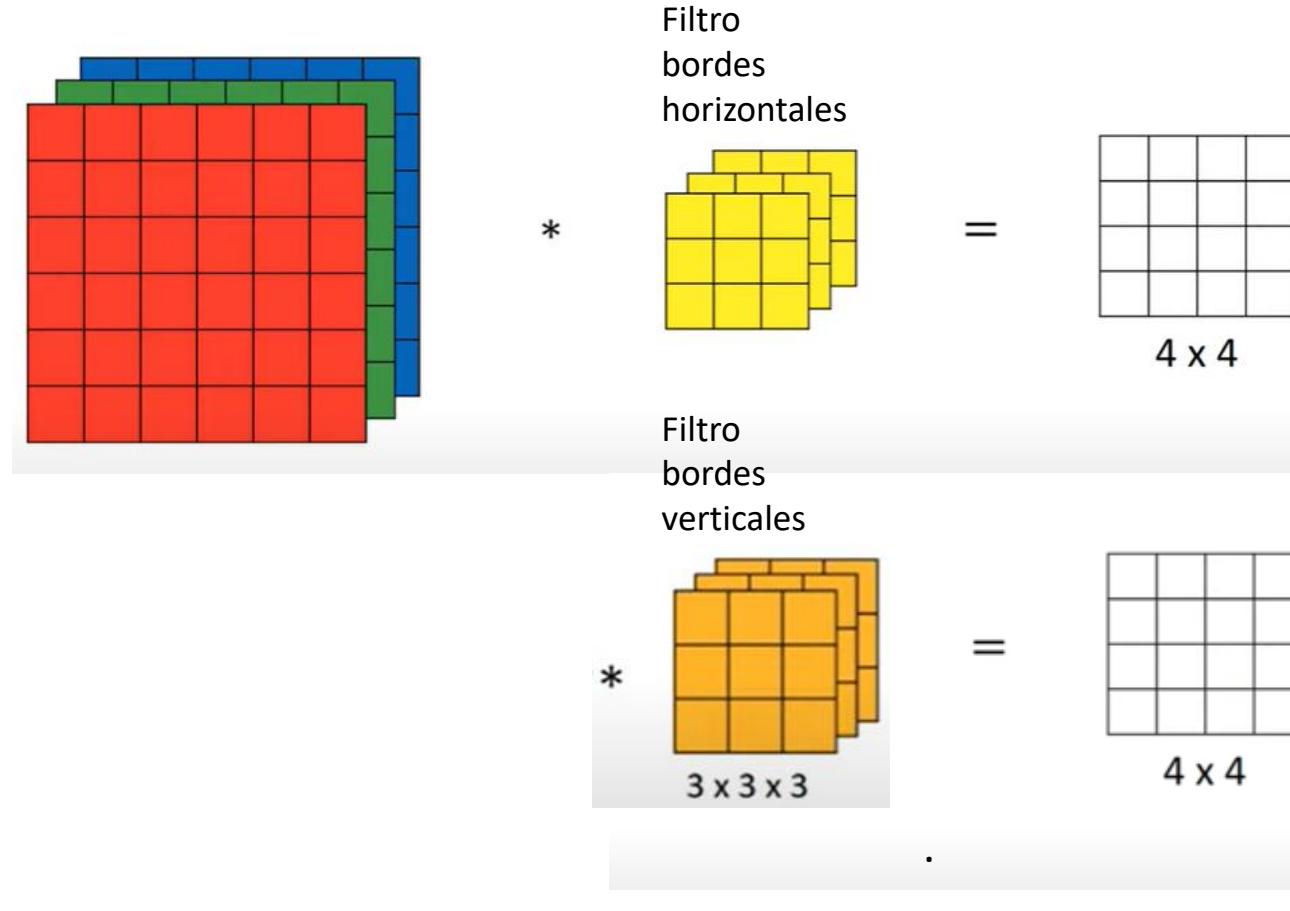
$$d = 1 + (n+2p-f)/s$$

$$d = 1 + (7+2 \times 0 - 3)/2$$

$$d = 1 + 2$$

$$d = 3$$

Convolución en imágenes RGB

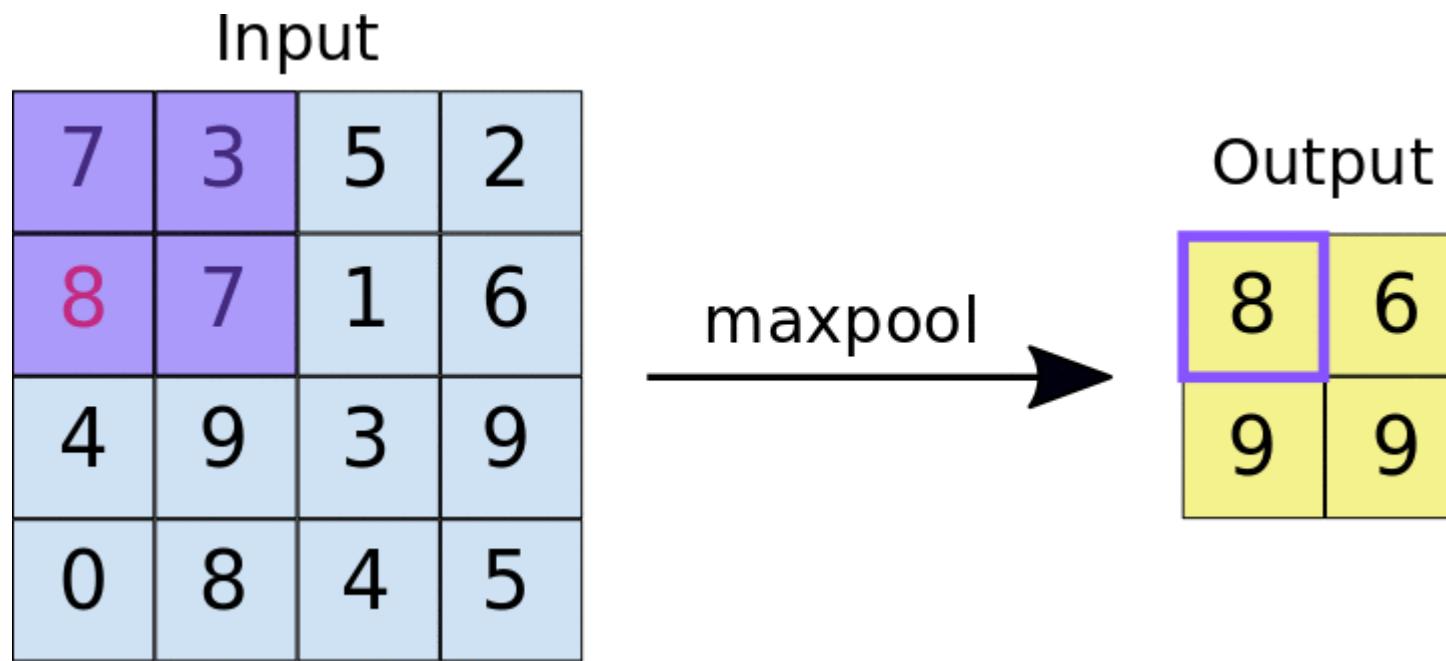


Dimension = $d \times d \times n_f$
Donde:
 n_f = número de filtros

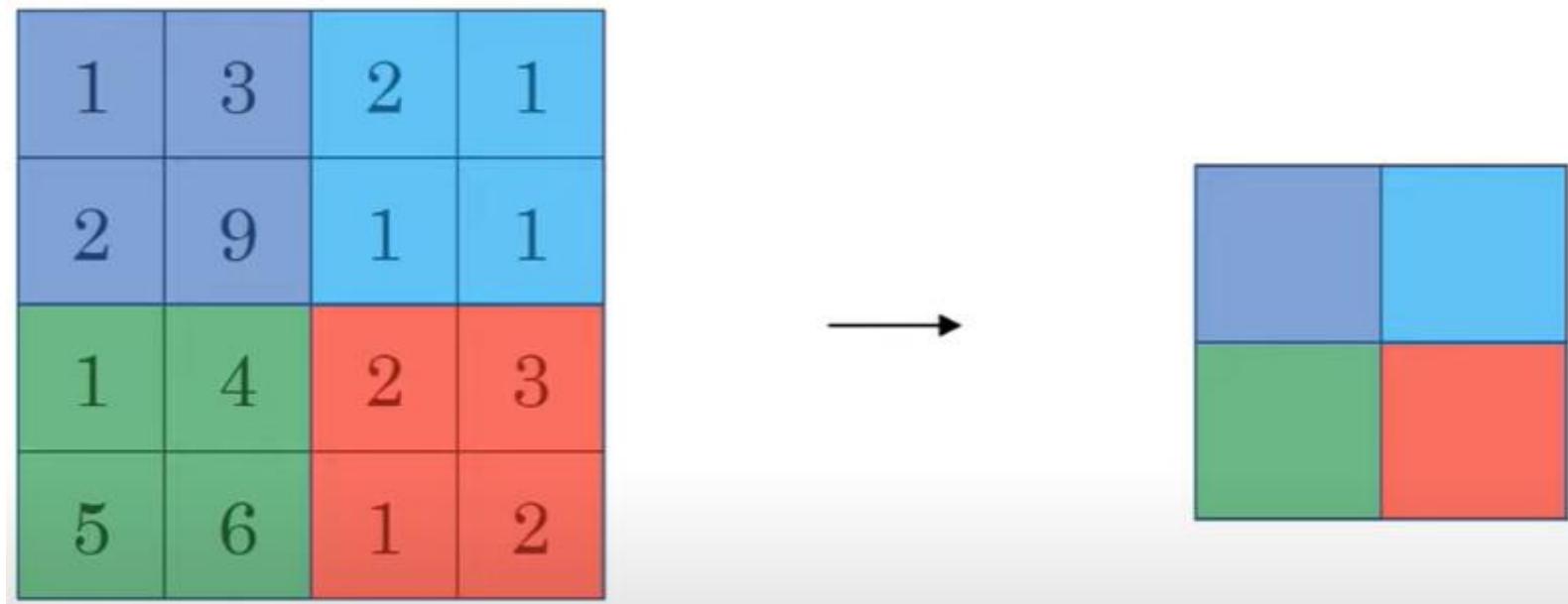
Demo

<https://tinymilky.github.io/demos/conv/index.html>

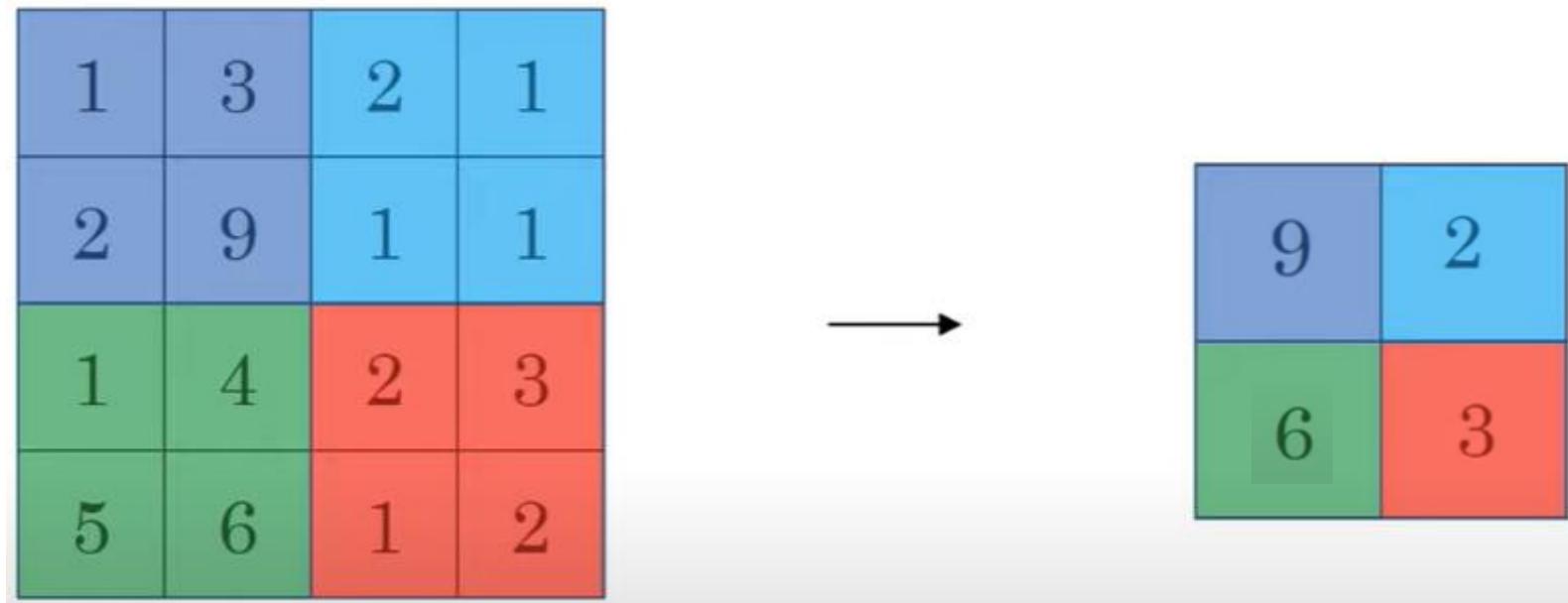
Pooling



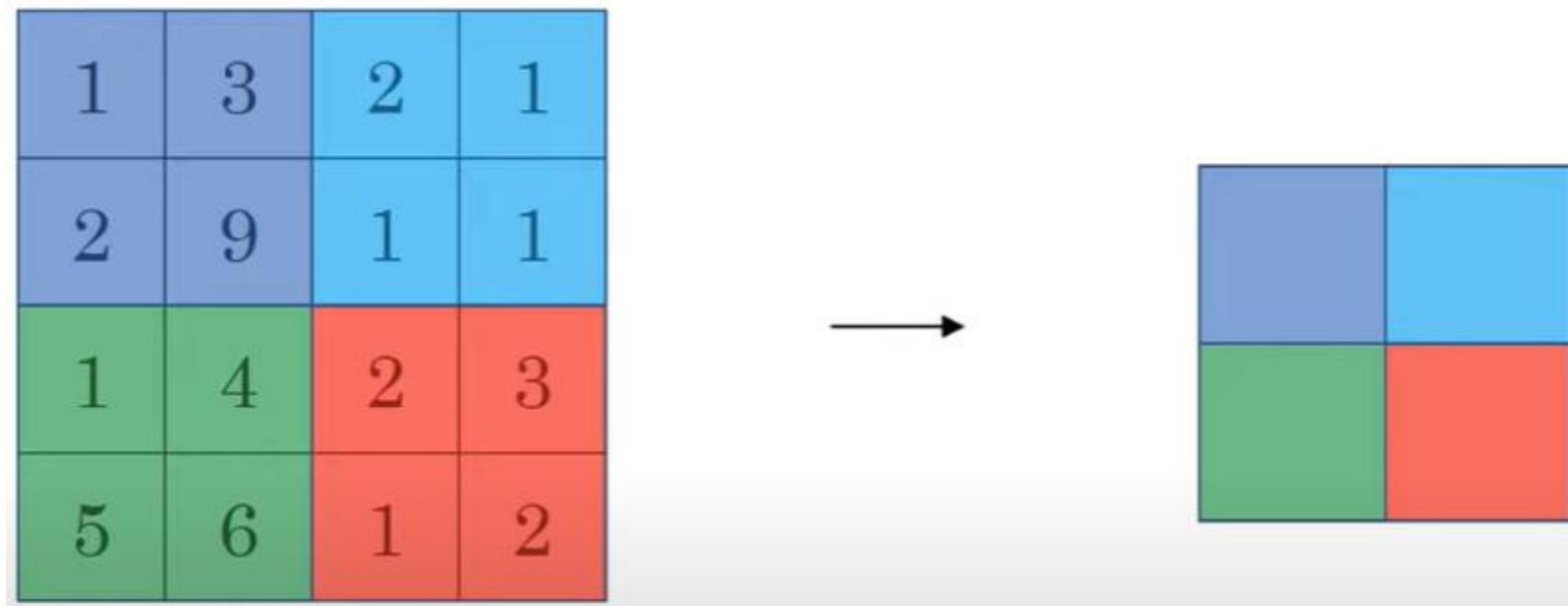
Max Pooling:



Max Pooling



Avg Pooling: Ejercicio



Pooling

1	3	2	1	3
2	9	1	1	5
1	3	2	3	2
8	3	5	1	0
5	6	1	2	9

$$d = 1 + (n+2p-f)/s$$

$$n=5$$

$$f=3$$

$$s=1$$

$$p=0$$

Pooling:Ejercicio

1	3	2	1	3
2	9	1	1	5
1	3	2	3	2
8	3	5	1	0
5	6	1	2	9

$$n=5$$

$$f=3$$

$$s=1$$

$$p=0$$

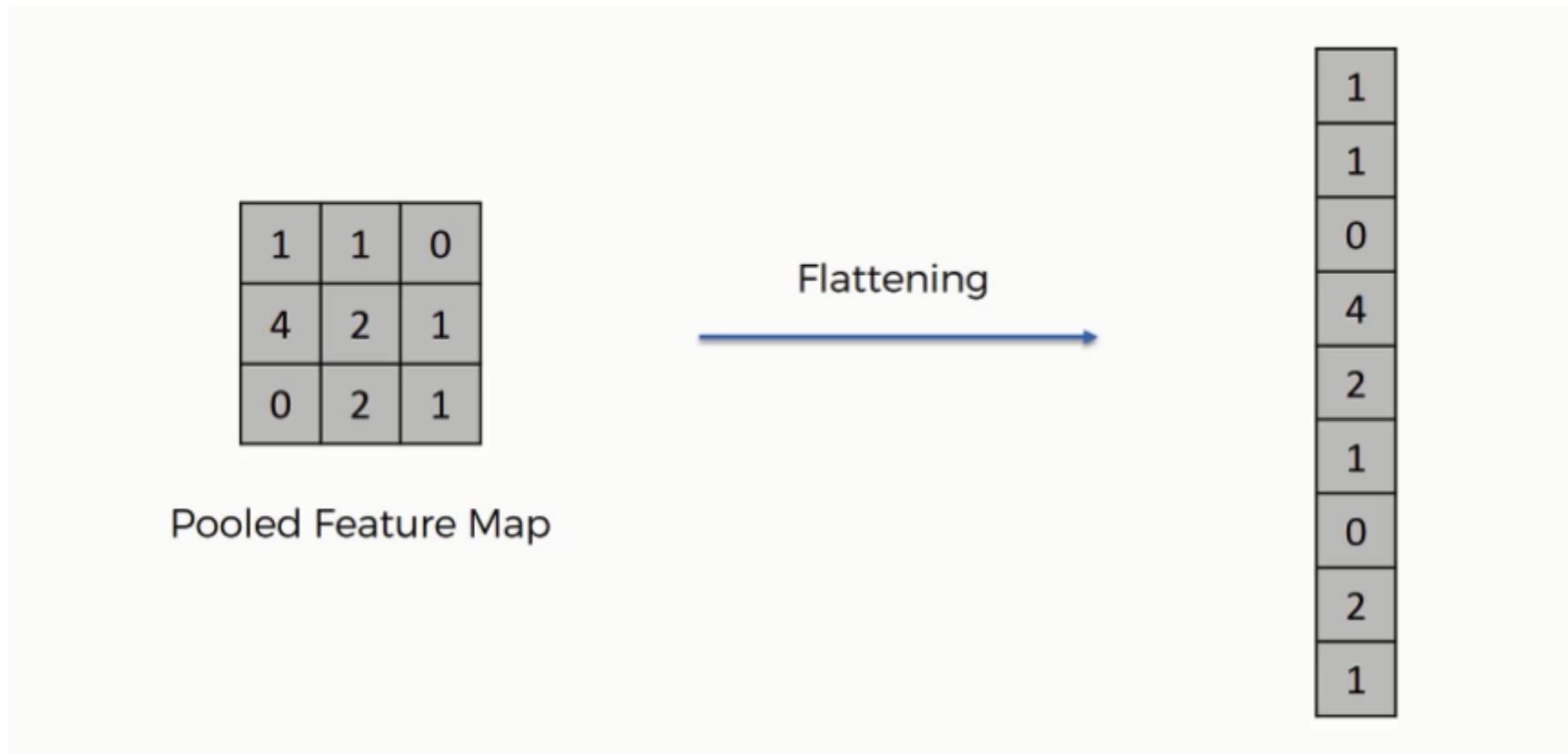
$$d = 1 + (n-f)/s$$

$$d = 1 + (5-3)/1$$

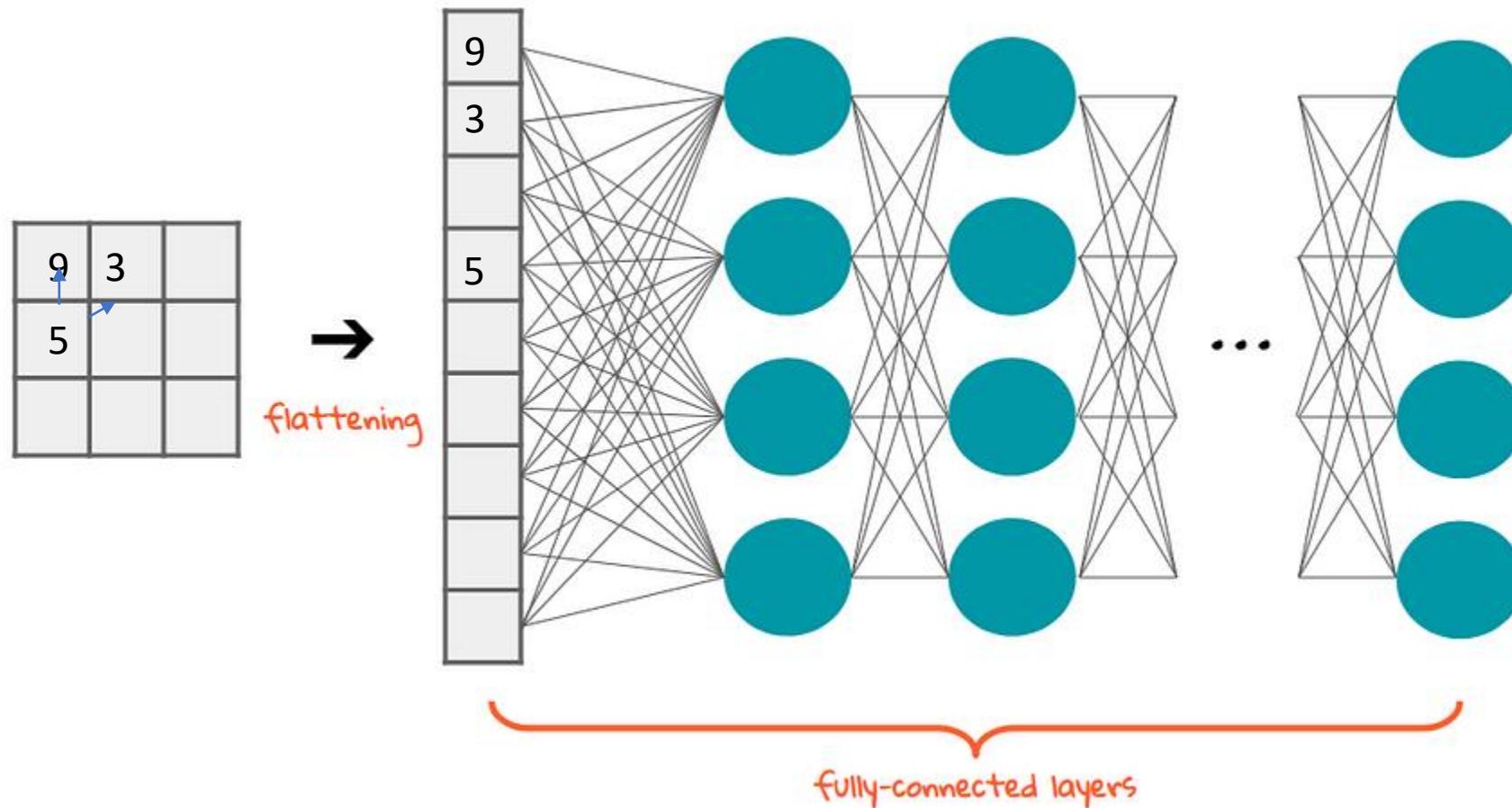
$$d = 1 + 2$$

$$d = 3$$

Flatten

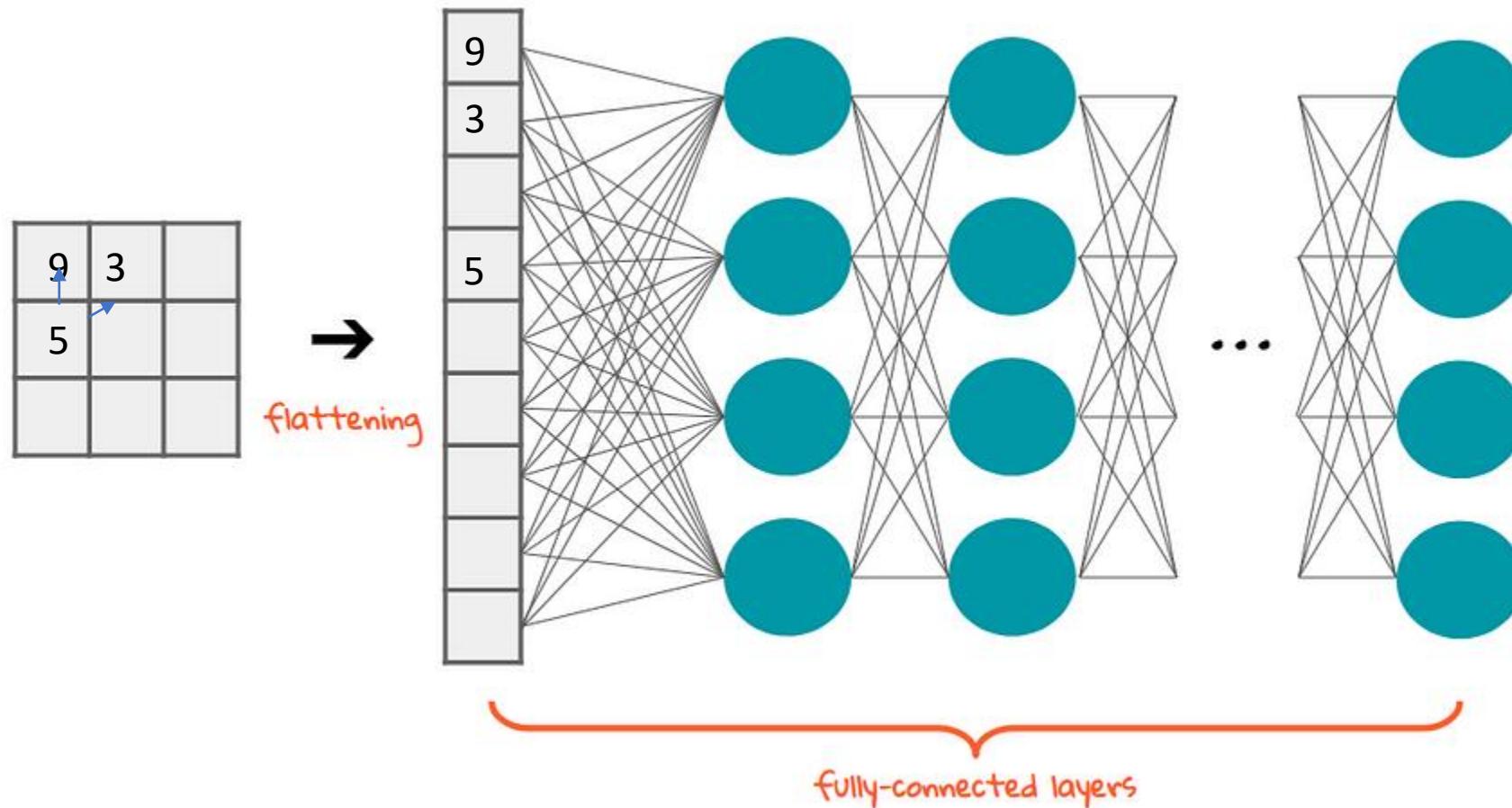


Flatten



Fuente: <https://towardsdatascience.com/the-most-intuitive-and-easiest-guide-for-convolutional-neural-network-3607be47480>

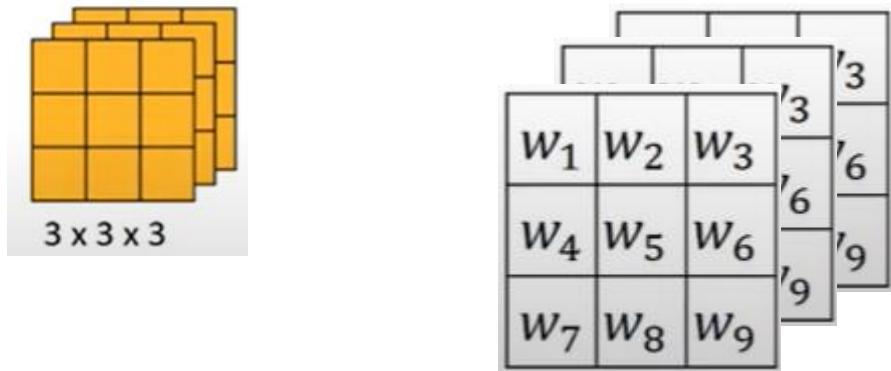
Flatten



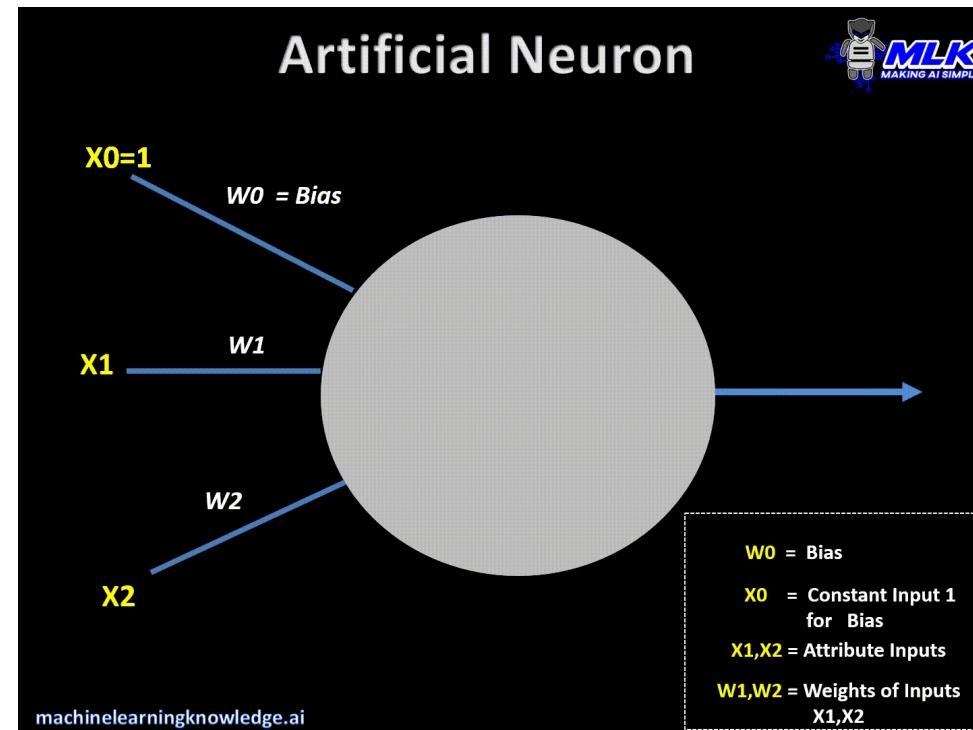
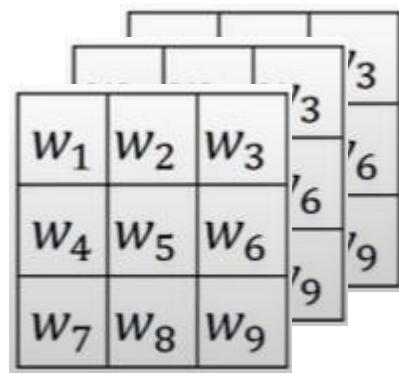
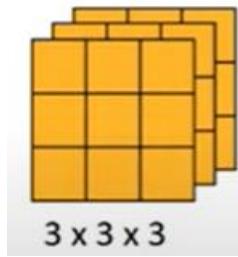
Fuente: <https://towardsdatascience.com/the-most-intuitive-and-easiest-guide-for-convolutional-neural-network-3607be47480>

Ejercicio: En una capa de convolución de 10 filtros de dimensión 3x3x3 cuantos parámetros podemos identificar

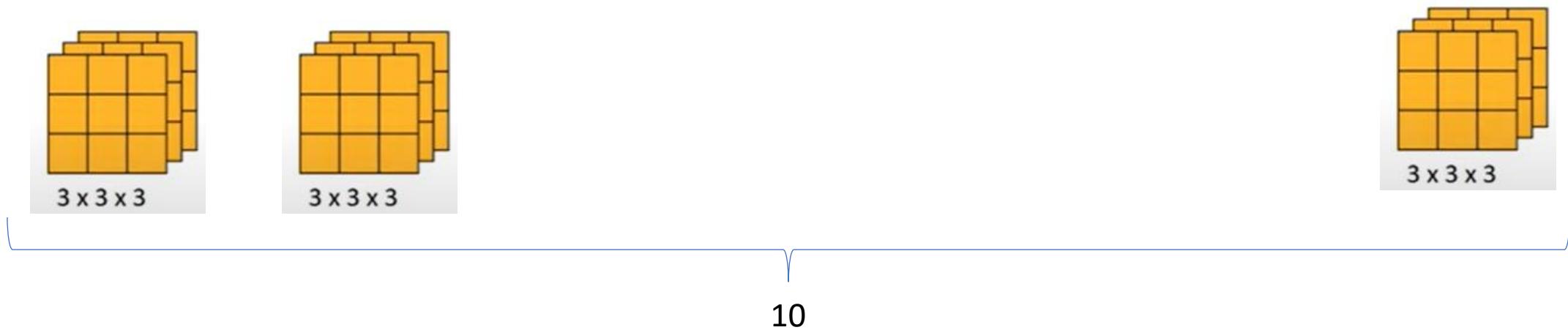
Ejercicio: En una capa de convolución de 10 filtros de dimensión 3x3x3 cuantos parámetros podemos identificar



Ejercicio: En una capa de convolución de 10 filtros de dimensión 3x3x3 cuantos parámetros podemos identificar



Ejercicio: En una capa de convolución de 10 filtros de dimensión 3x3x3 cuantos parámetros podemos identificar

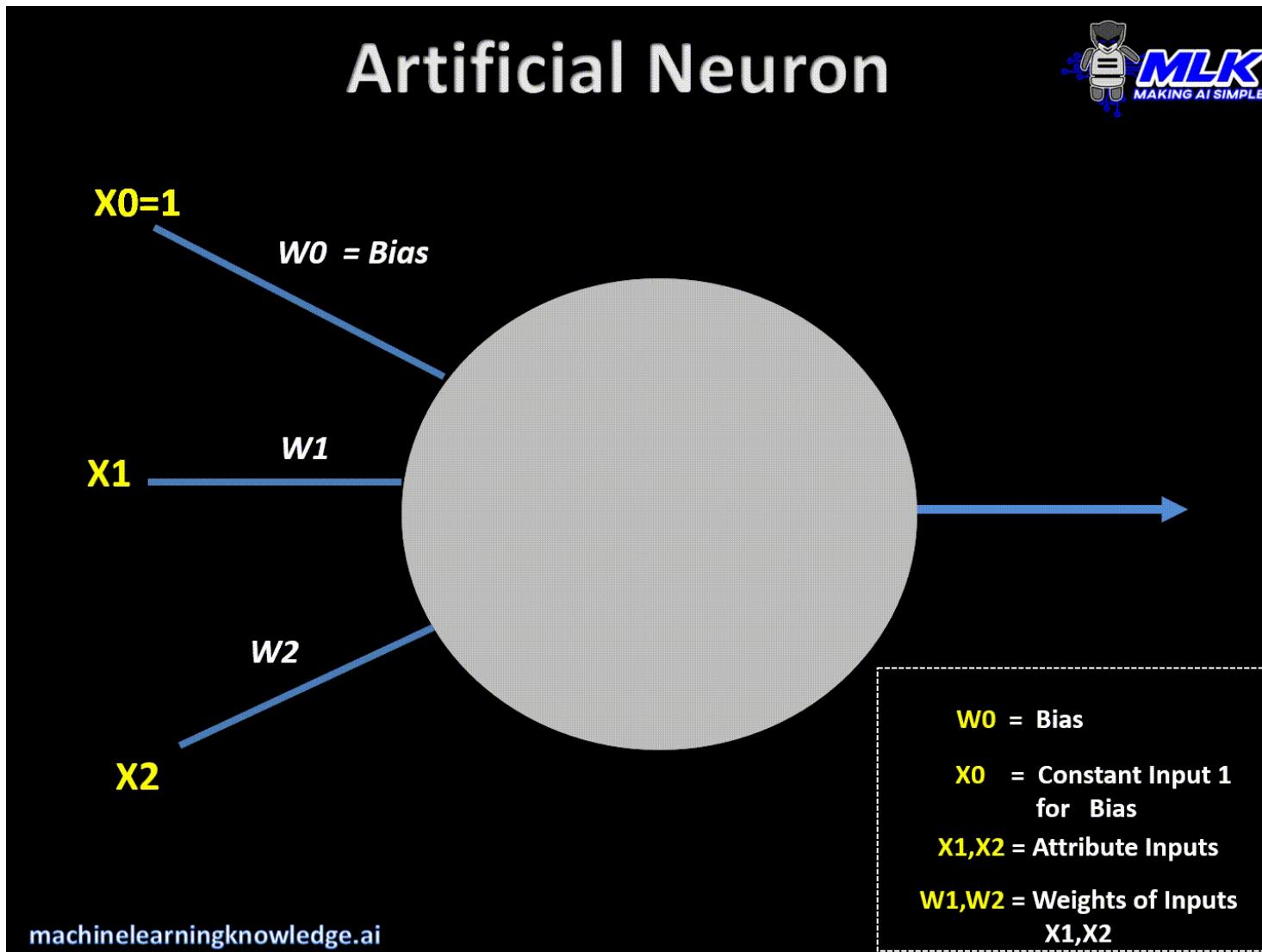


Para cada filtro

$$\text{Dimensión + bias} = 3 \cdot 3 \cdot 3 + 1 = 28$$

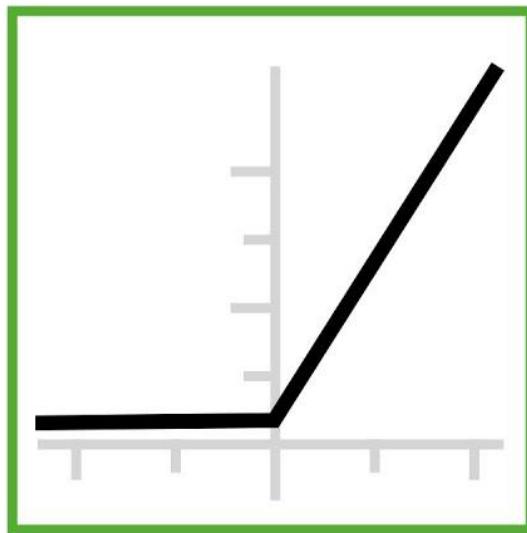
$$\text{Por 10 filtros} = 280$$

Funciones de activación



ReLU

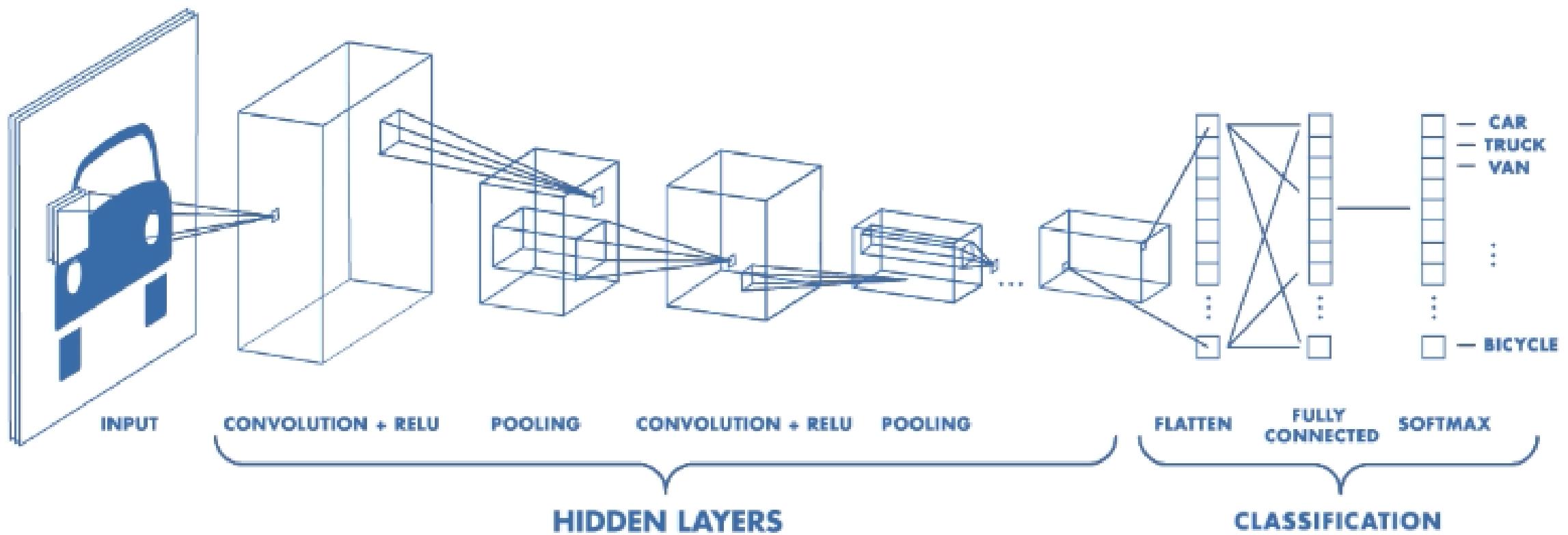
The ReLU Activation Function...



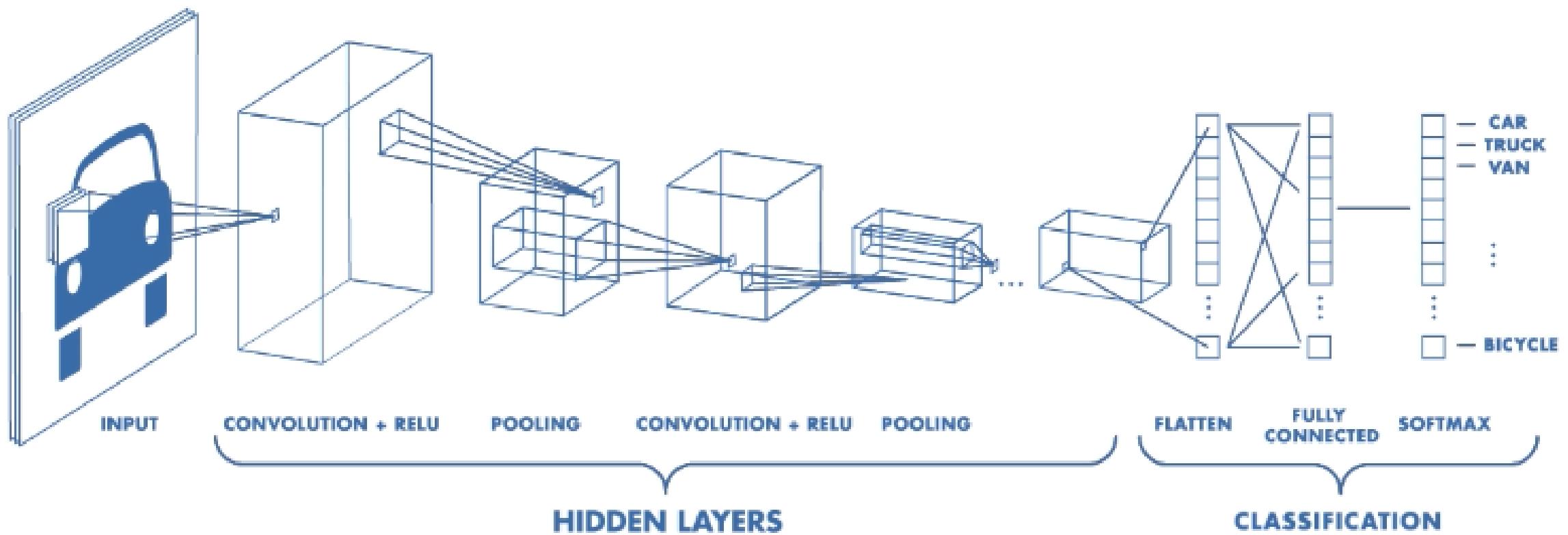
$$f(x) = \max(0, x)$$

Aula 4

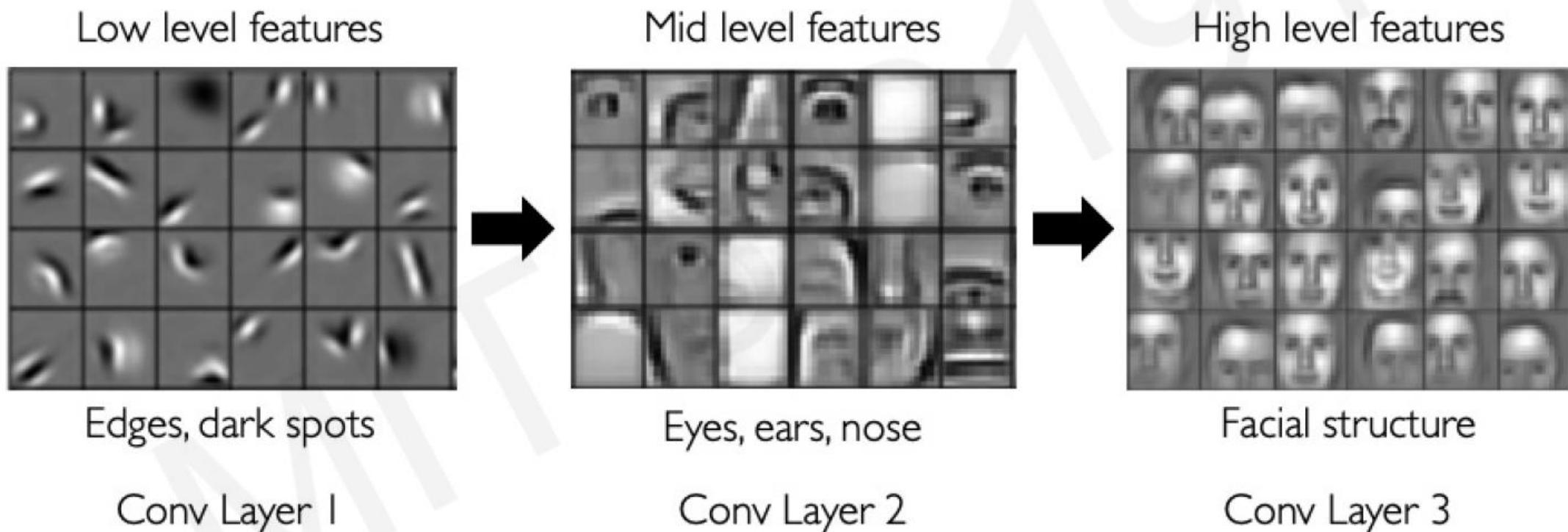
Previamente ...



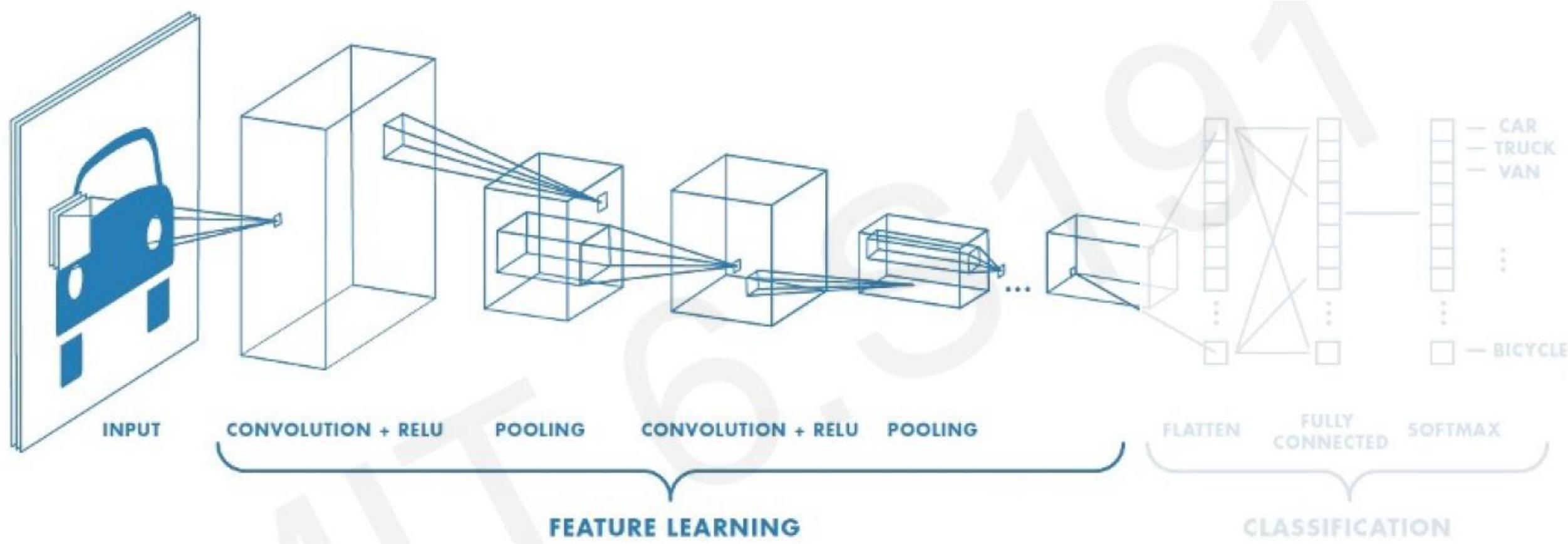
Previamente ...



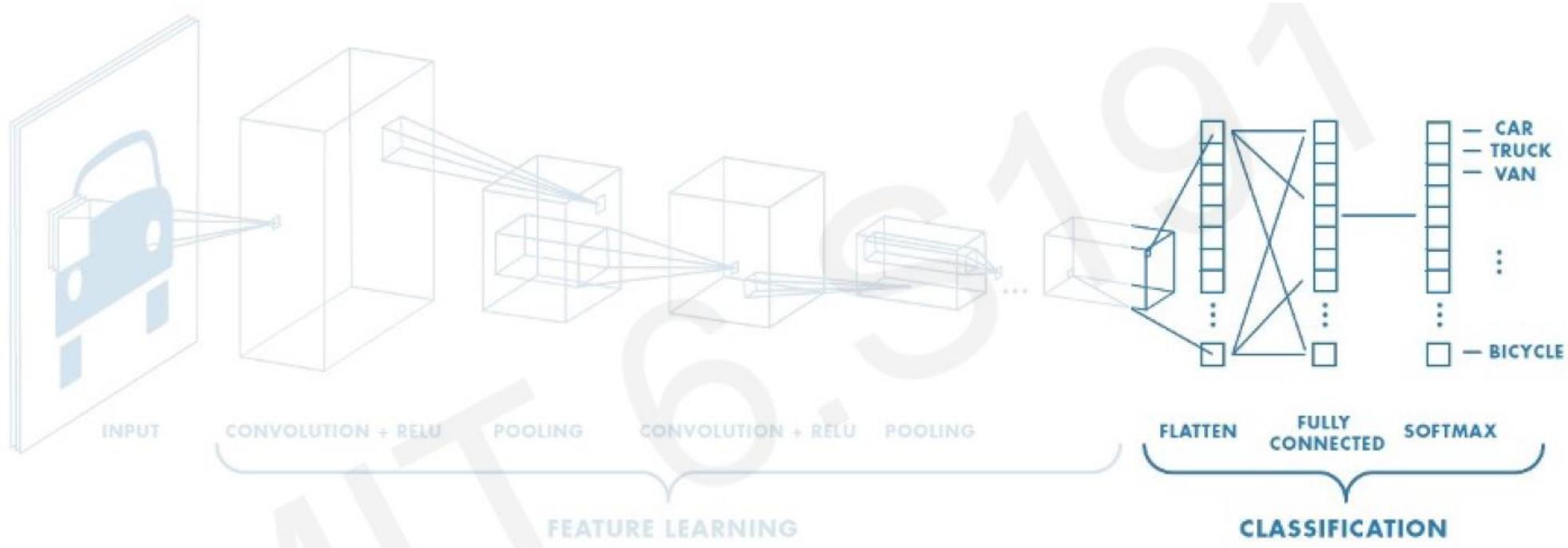
Como se organiza una CNN?



Como se organiza una CNN?



Como se organiza una CNN?



$$\text{softmax}(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}}$$

Softmax

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}$$

$$\begin{bmatrix} 8 \\ 5 \\ 0 \end{bmatrix}$$

K Número de clases

e^{z_i} Exponencial de Z_i

z_i i-esima entrada de la capa

\vec{z} Vector de entrada

Softmax

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}$$

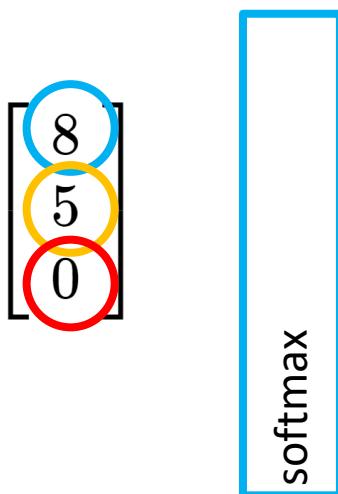
K	Número de clases
e^{z_i}	Exponencial de Z_i
z_i	i-esima entrada de la capa
\vec{z}	Vector de entrada



Softmax

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}$$

K Número de clases
 e^{z_i} Exponencial de Z_i
 z_i i-esima entrada de la capa



$$e^{z_1} = e^8 = 2981.0$$

$$e^{z_2} = e^5 = 148.4$$

$$e^{z_3} = e^0 = 1.0$$

Softmax

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}$$

K Número de clases
 e^{z_i} Exponencial de Z_i
 z_i i-esima entrada de la capa



softmax

$$e^{z_1} = e^8 = 2981.0$$

$$e^{z_2} = e^5 = 148.4$$

$$e^{z_3} = e^0 = 1.0$$

$$\sum_{j=1}^K e^{z_j} = e^{z_1} + e^{z_2} + e^{z_3} = 2981.0 + 148.4 + 1.0 = 3130.4$$

Softmax

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}$$

K Número de clases

e^{z_i} Exponencial de Z_i

z_i i-esima entrada de la capa



softmax

$$\sum_{j=1}^K e^{z_j} = e^{z_1} + e^{z_2} + e^{z_3} = 2981.0 + 148.4 + 1.0 = 3130.4$$

$$e^{z_1} = e^8 = 2981.0$$

$$e^{z_2} = e^5 = 148.4$$

$$e^{z_3} = e^0 = 1.0$$

$$\sigma(\vec{z})_1 = \frac{2981.0}{3130.4} = 0.9523$$

$$\sigma(\vec{z})_2 = \frac{148.4}{3130.4} = 0.0474$$

$$\sigma(\vec{z})_3 = \frac{1.0}{3130.4} = 0.0003$$

Argmax

Softmax

$$\sigma(\vec{z})_1 = \sigma(3) = \frac{e^3}{e^3 + e^0} = 0.953$$
$$\sigma(\vec{z})_2 = \sigma(0) = \frac{e^0}{e^3 + e^0} = 0.0474$$

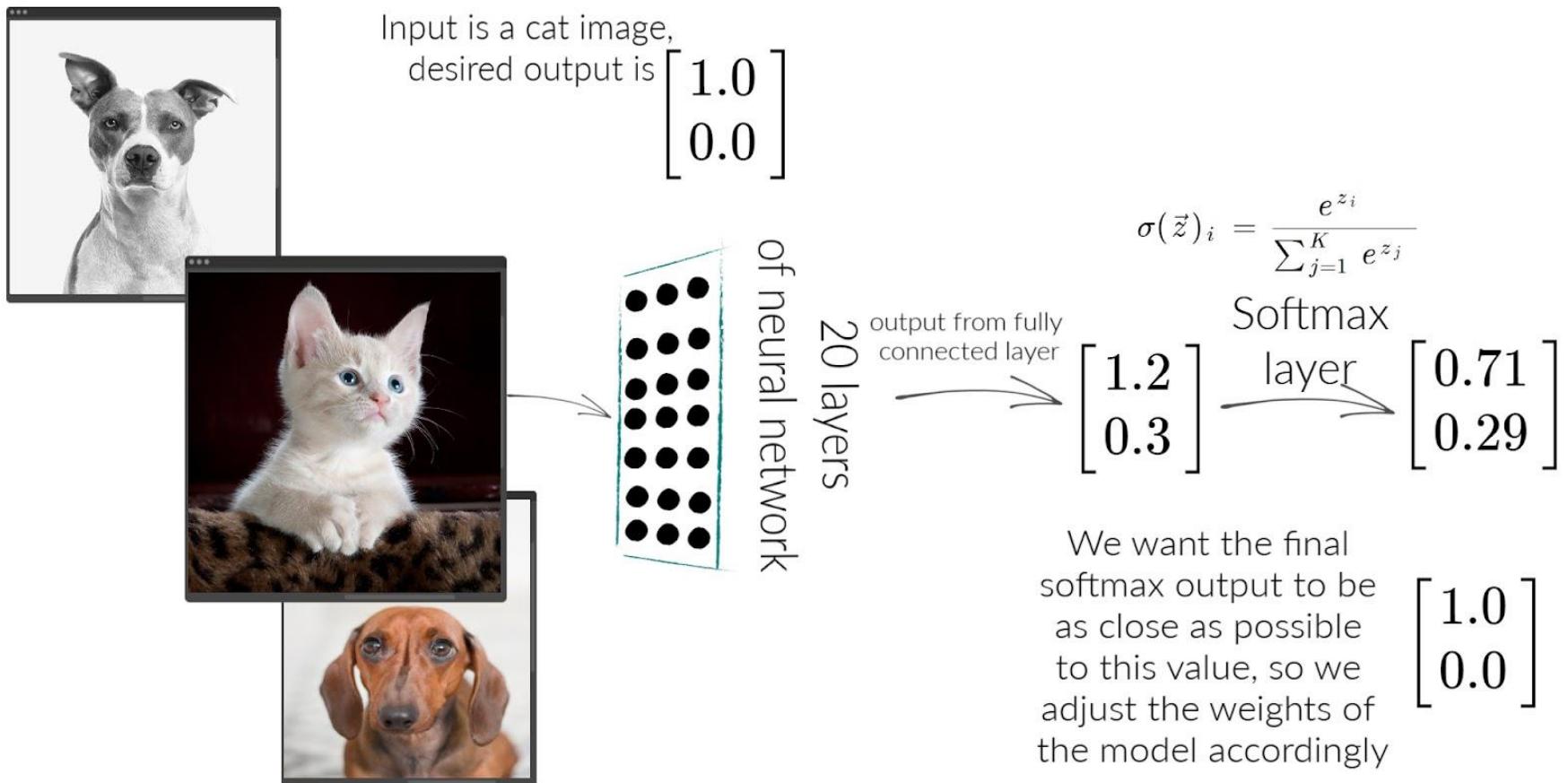
Argmax

$$\text{argmax}(\vec{z}) = \text{argmax}\left(\begin{bmatrix} 3 \\ 0 \end{bmatrix}\right) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\text{argmax}(\vec{z})_1 = 1$$

$$\text{argmax}(\vec{z})_2 = 0$$

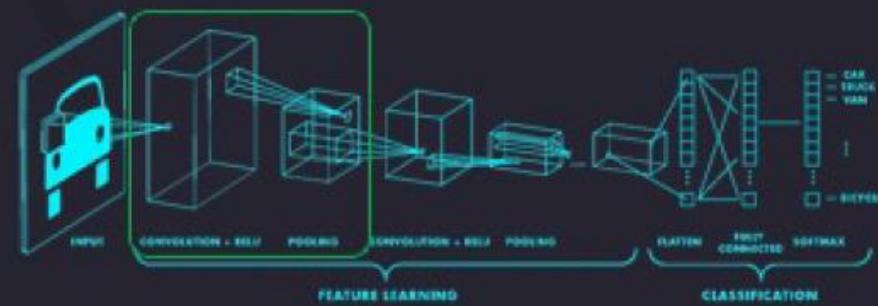
Resumen softmax



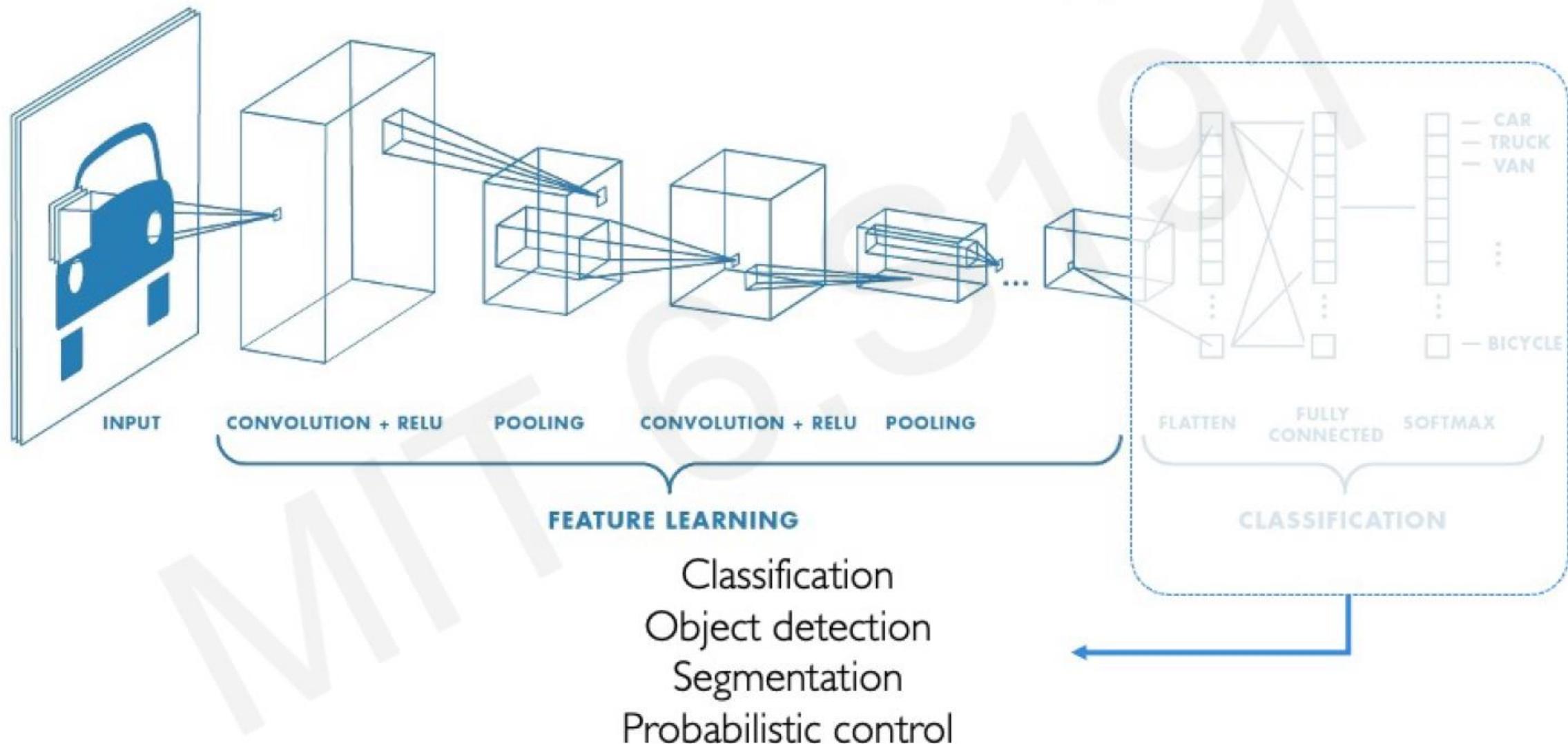
Como se organiza una CNN?

```
import tensorflow as tf

def generate_model():
    model = tf.keras.Sequential([
        # first convolutional layer
        tf.keras.layers.Conv2D(32, filter_size=3, activation='relu'),
        tf.keras.layers.MaxPool2D(pool_size=2, strides=2),
        # second convolutional layer
        tf.keras.layers.Conv2D(64, filter_size=3, activation='relu'),
        tf.keras.layers.MaxPool2D(pool_size=2, strides=2),
        # fully connected classifier
        tf.keras.layers.Flatten(),
        tf.keras.layers.Dense(1024, activation='relu'),
        tf.keras.layers.Dense(10, activation='softmax')  # 10 outputs
    ])
    return model
```



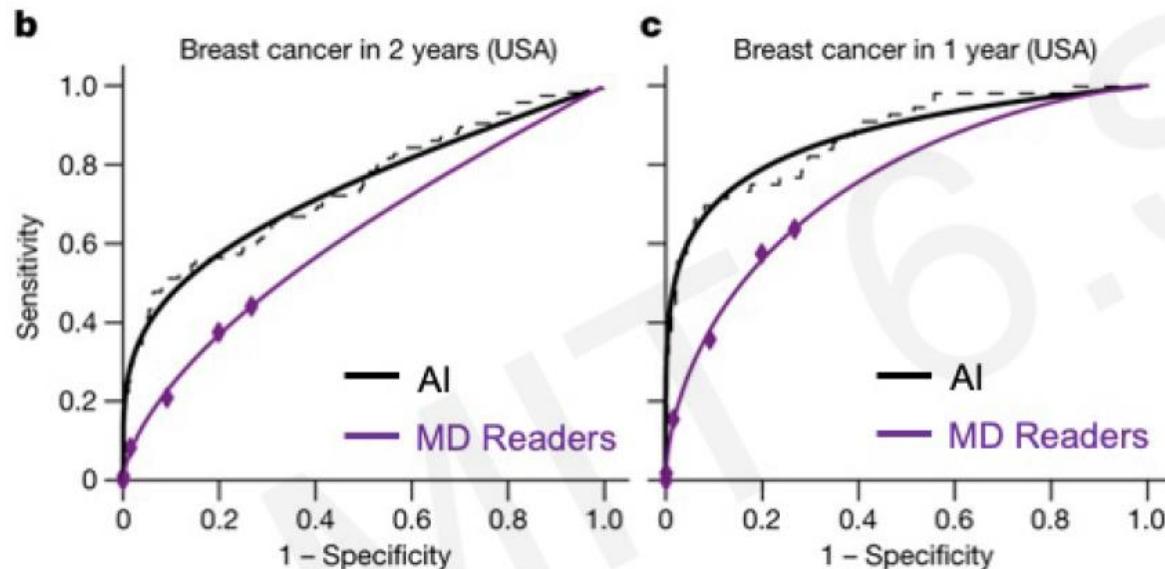
Como se organiza una CNN?



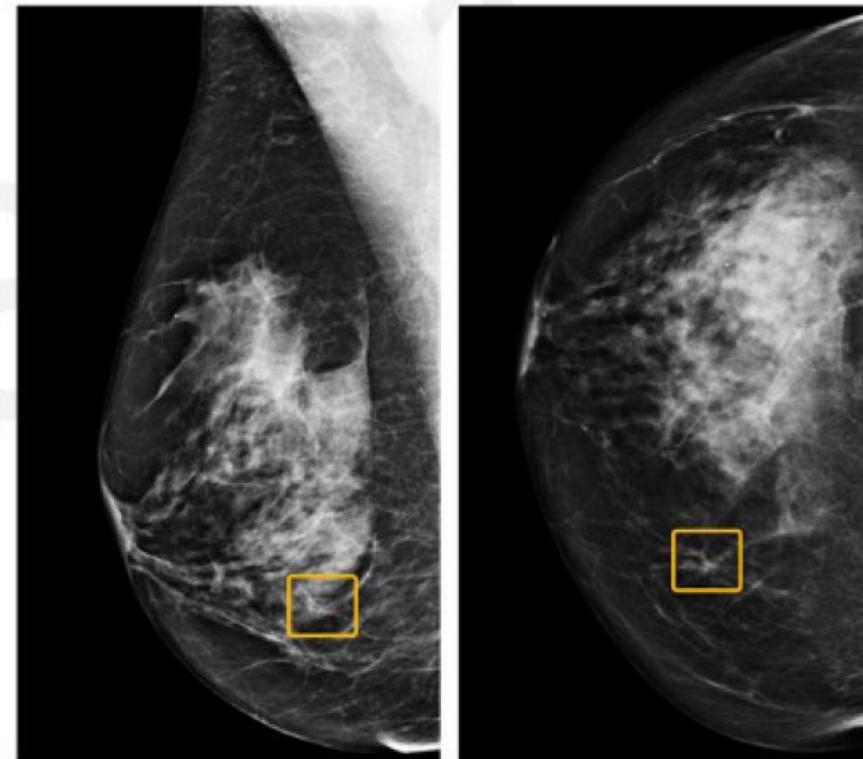
Aplicaciones

Classification: Breast Cancer Screening

International evaluation of an AI system for breast cancer screening



CNN-based system outperformed expert radiologists at detecting breast cancer from mammograms



Breast cancer case missed by radiologist but detected by AI

Aplicaciones

Object Detection

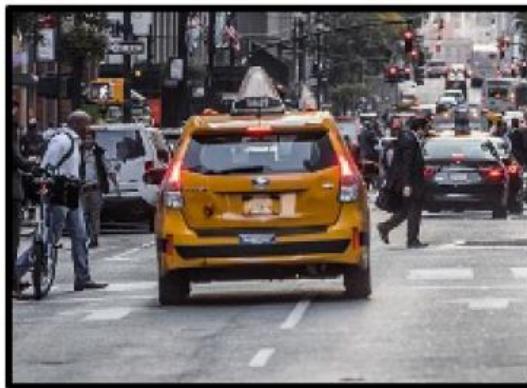


Image X

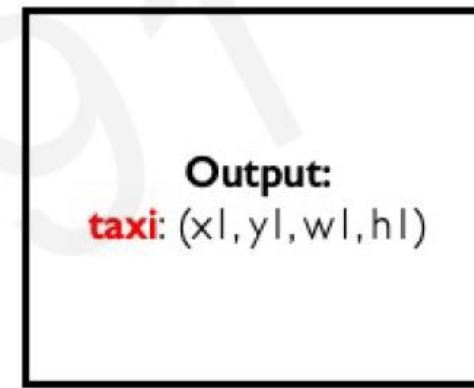
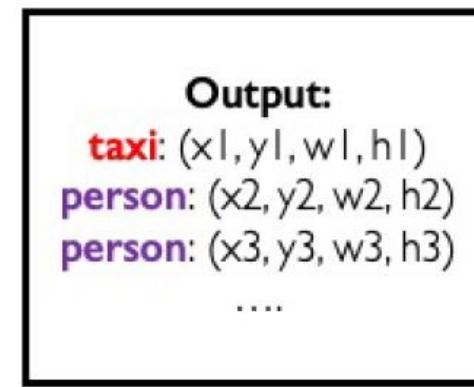


Image X



Aplicaciones

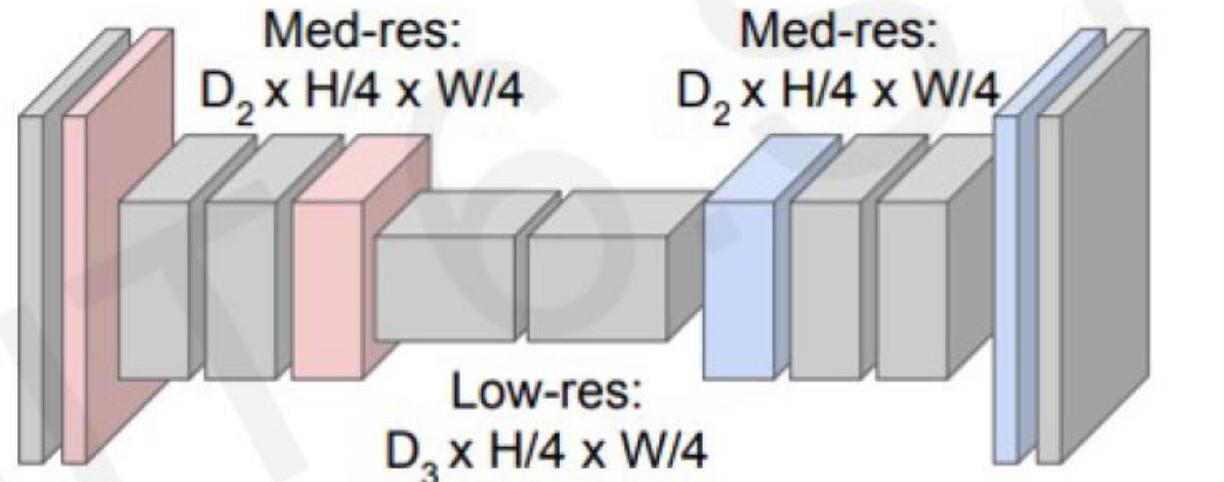
FCN: Fully Convolutional Network.

Network designed with all convolutional layers,
with **downsampling** and **upsampling** operations



Input:
 $3 \times H \times W$

High-res:
 $D_1 \times H/2 \times W/2$

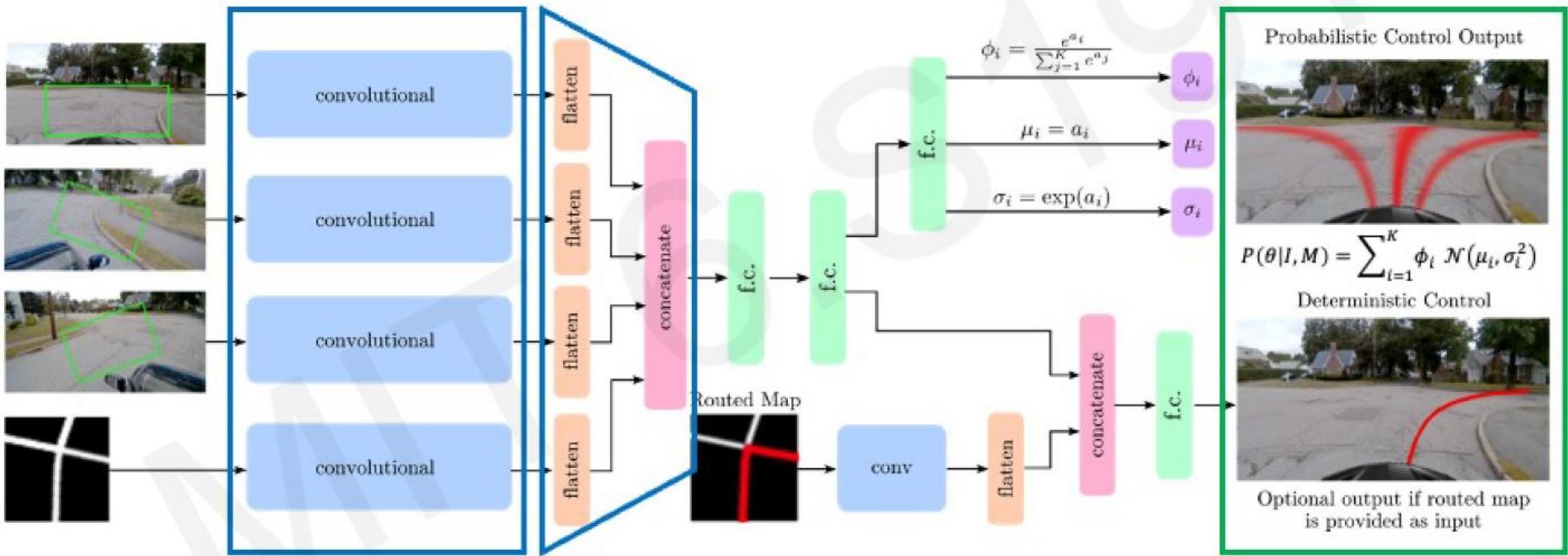


Predictions:
 $H \times W$



Aplicaciones

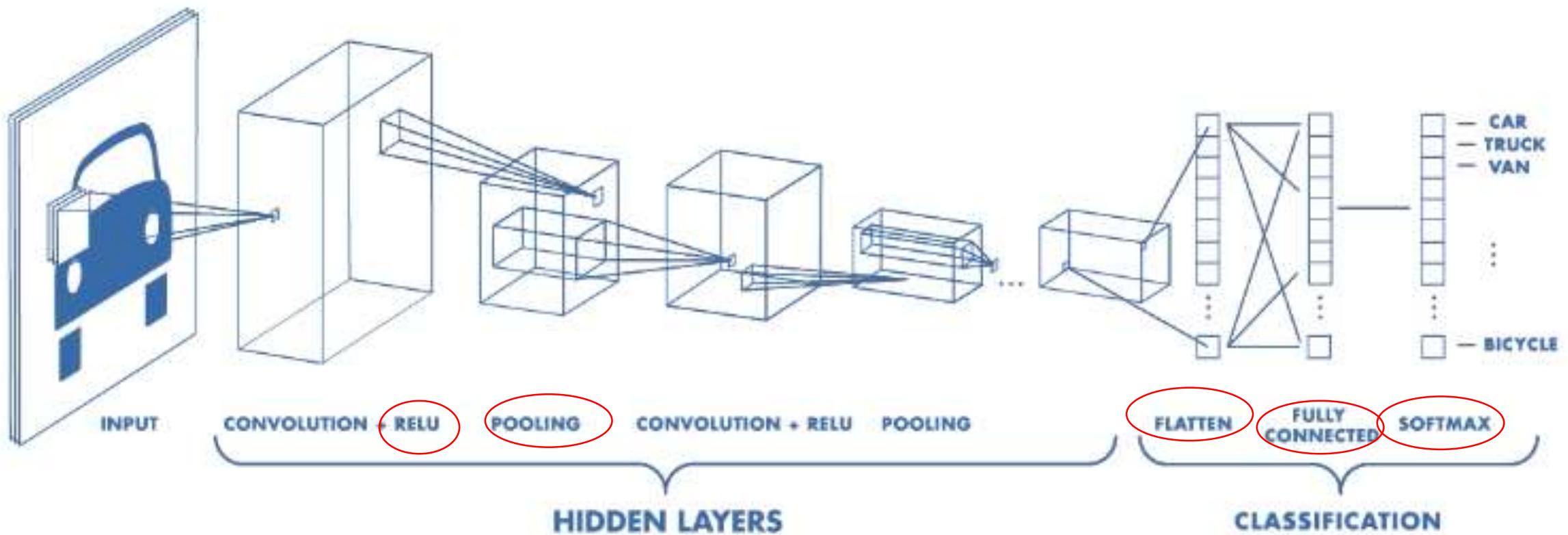
Entire model is trained end-to-end **without any human labelling or annotations**



El laboratorio ya está disponible en el classroom.

Hora máxima de entrega:
20:15 de hoy 22/06/23

Previamente ...



Aula 4

Redes Neuronales Recurrentes

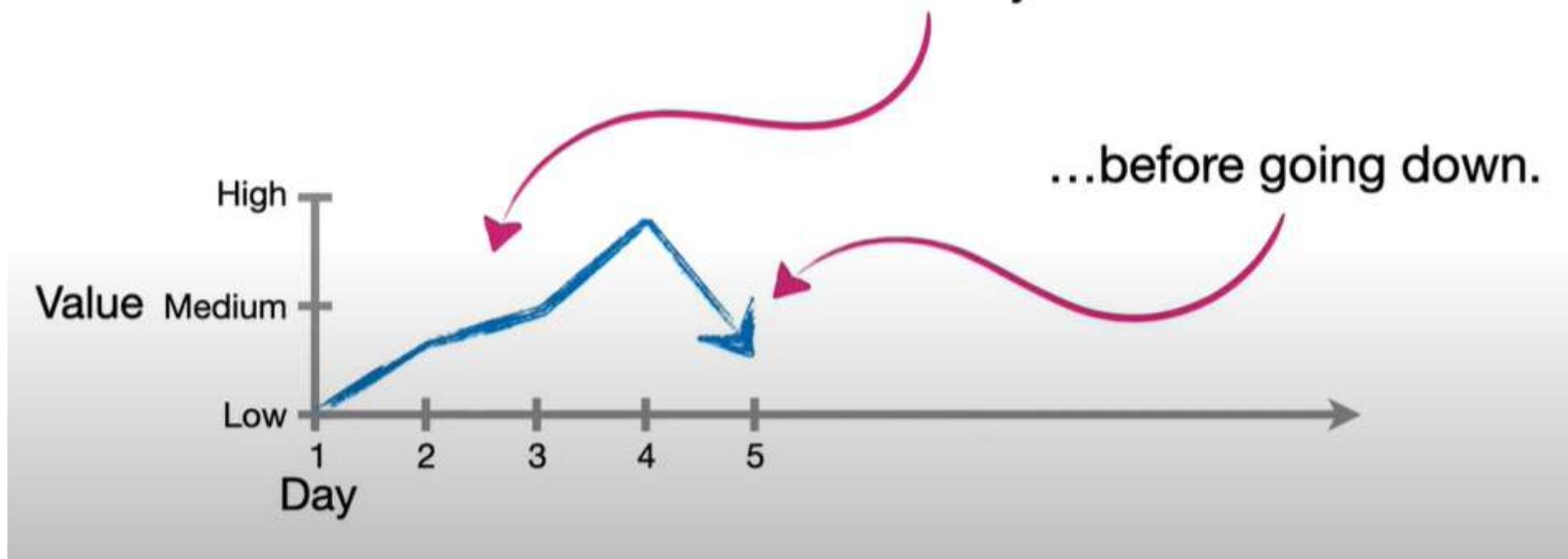
Motivación

When we look at stock prices, they tend to change over time.



Motivación

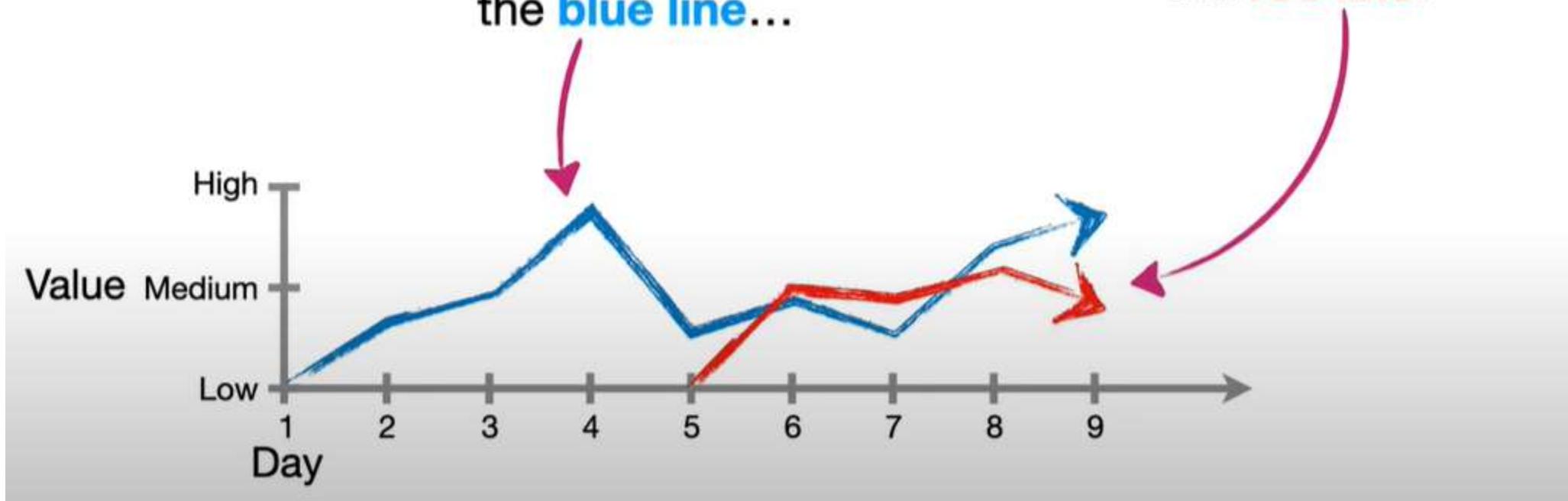
For example, the price
of this stock went up
for **4** days...



Motivación

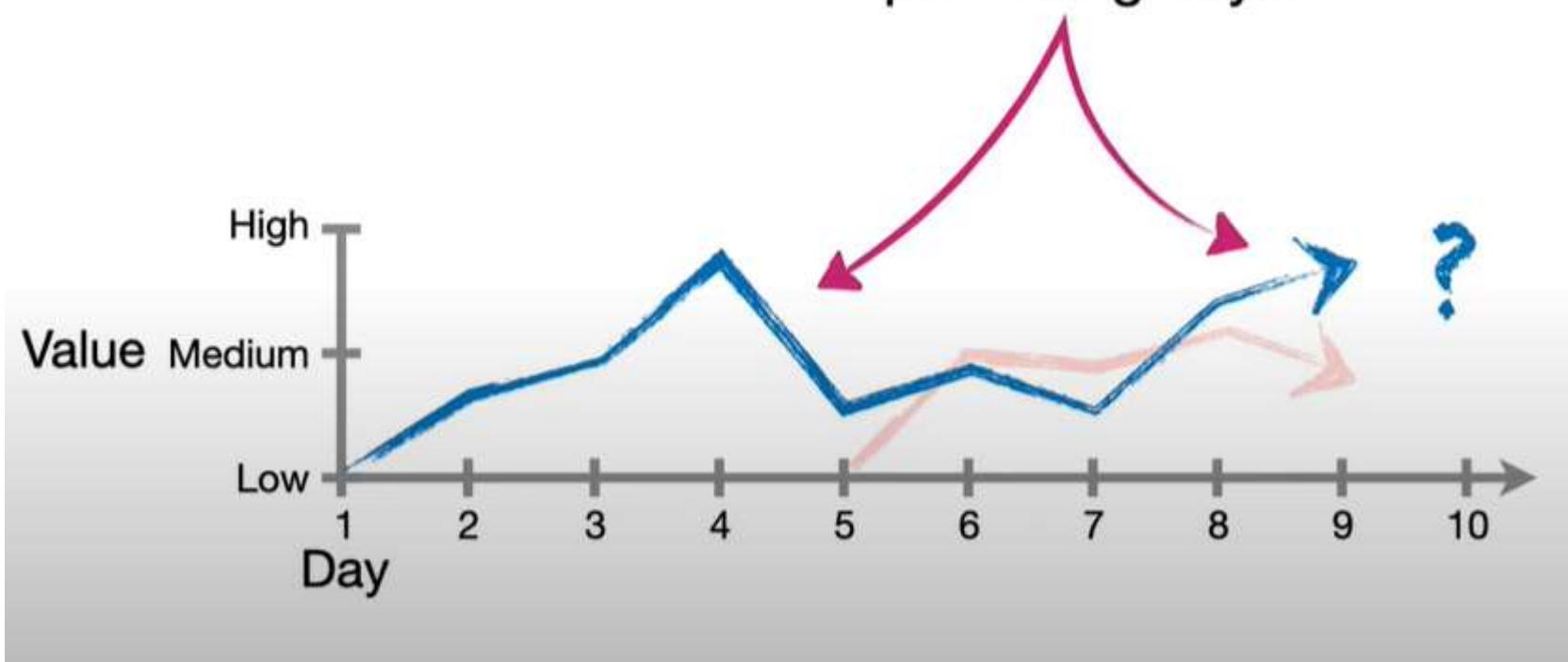
For example, we have more time points for the company represented by the **blue line**...

...then we have for the company represented by the **red line**.



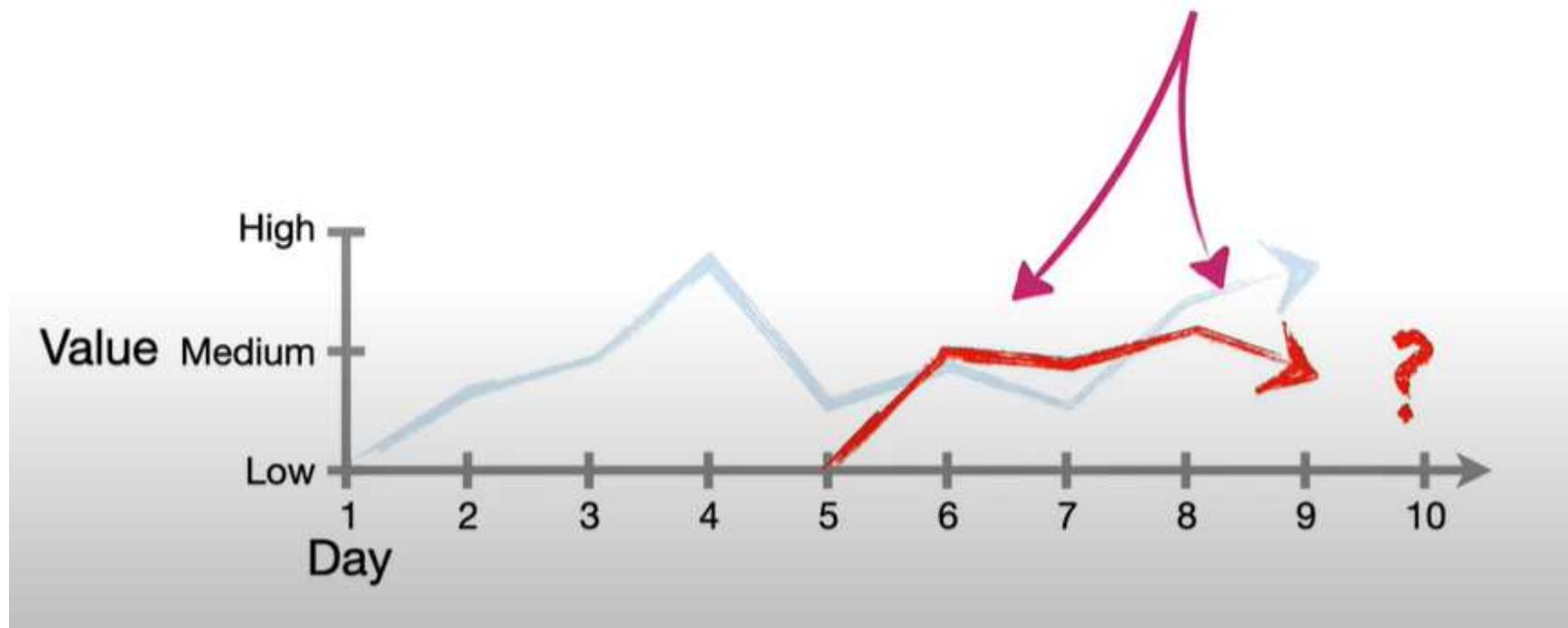
Motivación

...then we might want to use
the data from all **9** of the
preceding days.



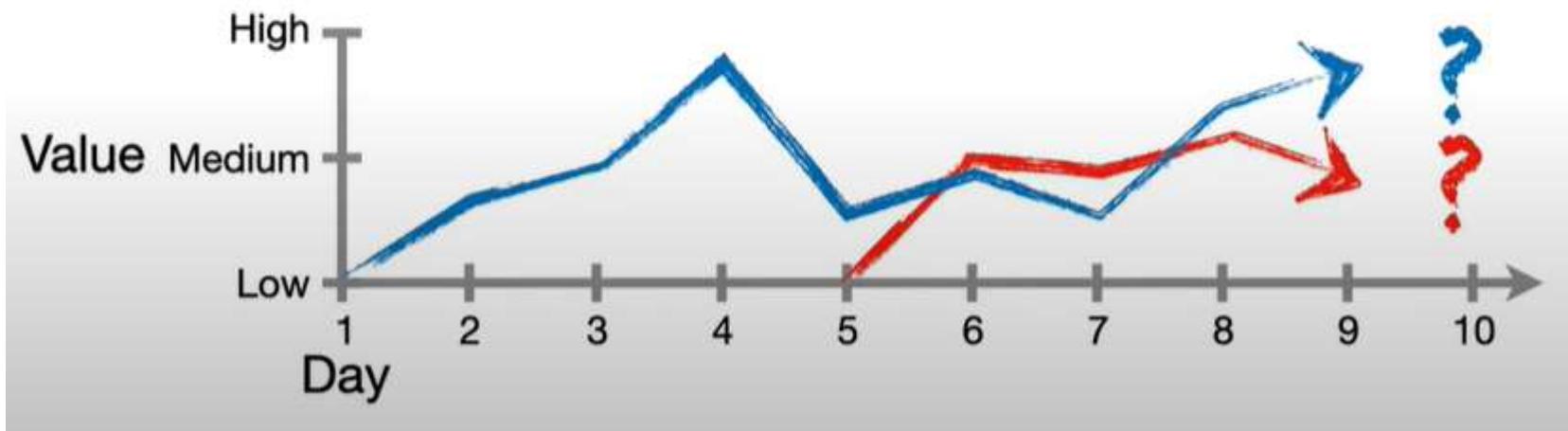
Motivación

...then we would only have data
for the preceding **5** days.



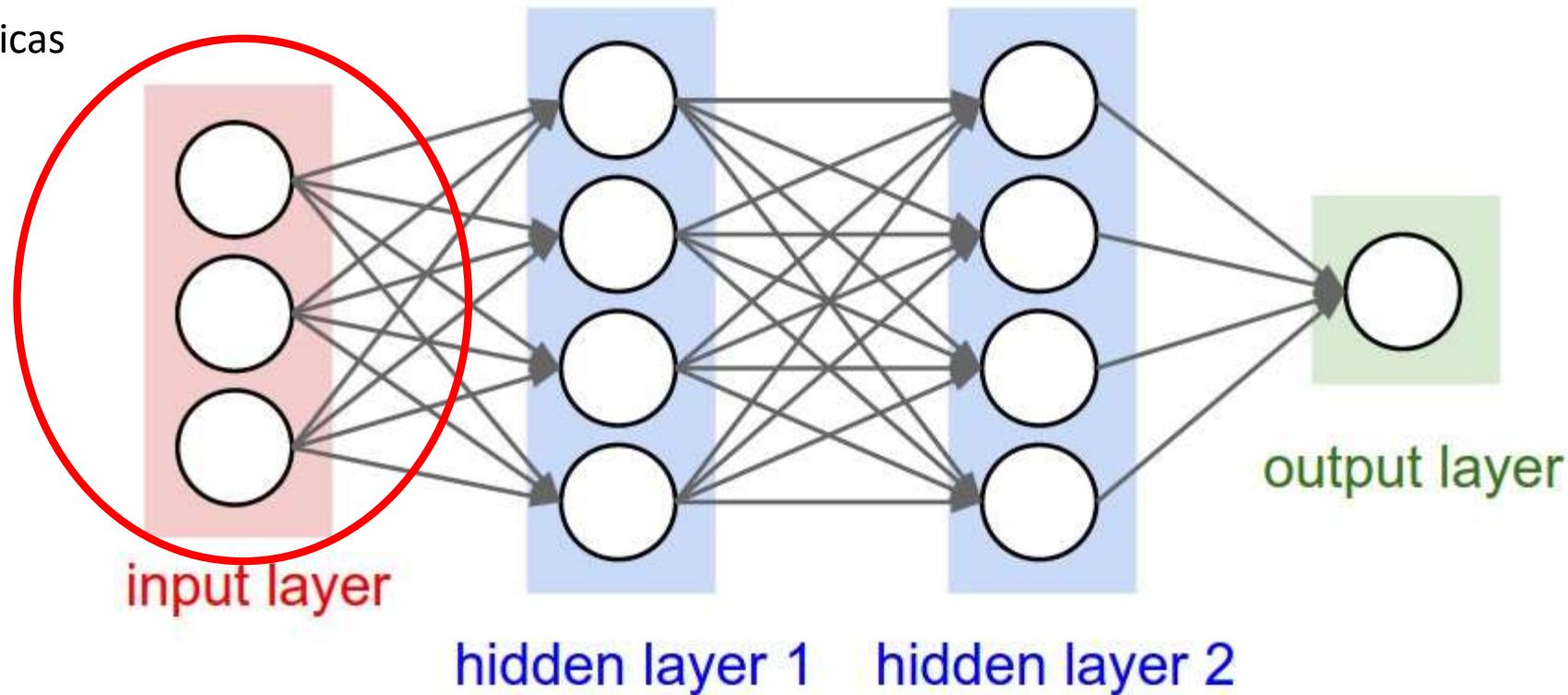
Motivación

So we need the neural network
to be flexible in terms of how
much sequential data we use
to make a prediction.



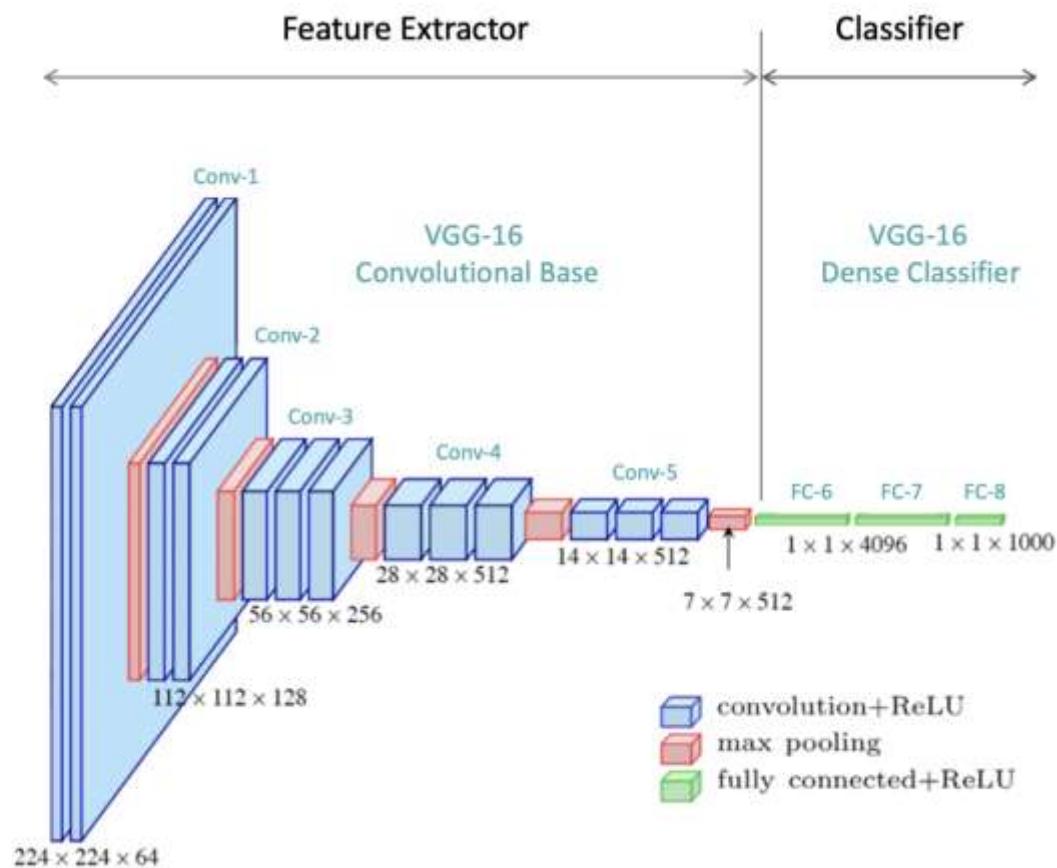
Motivación

Características

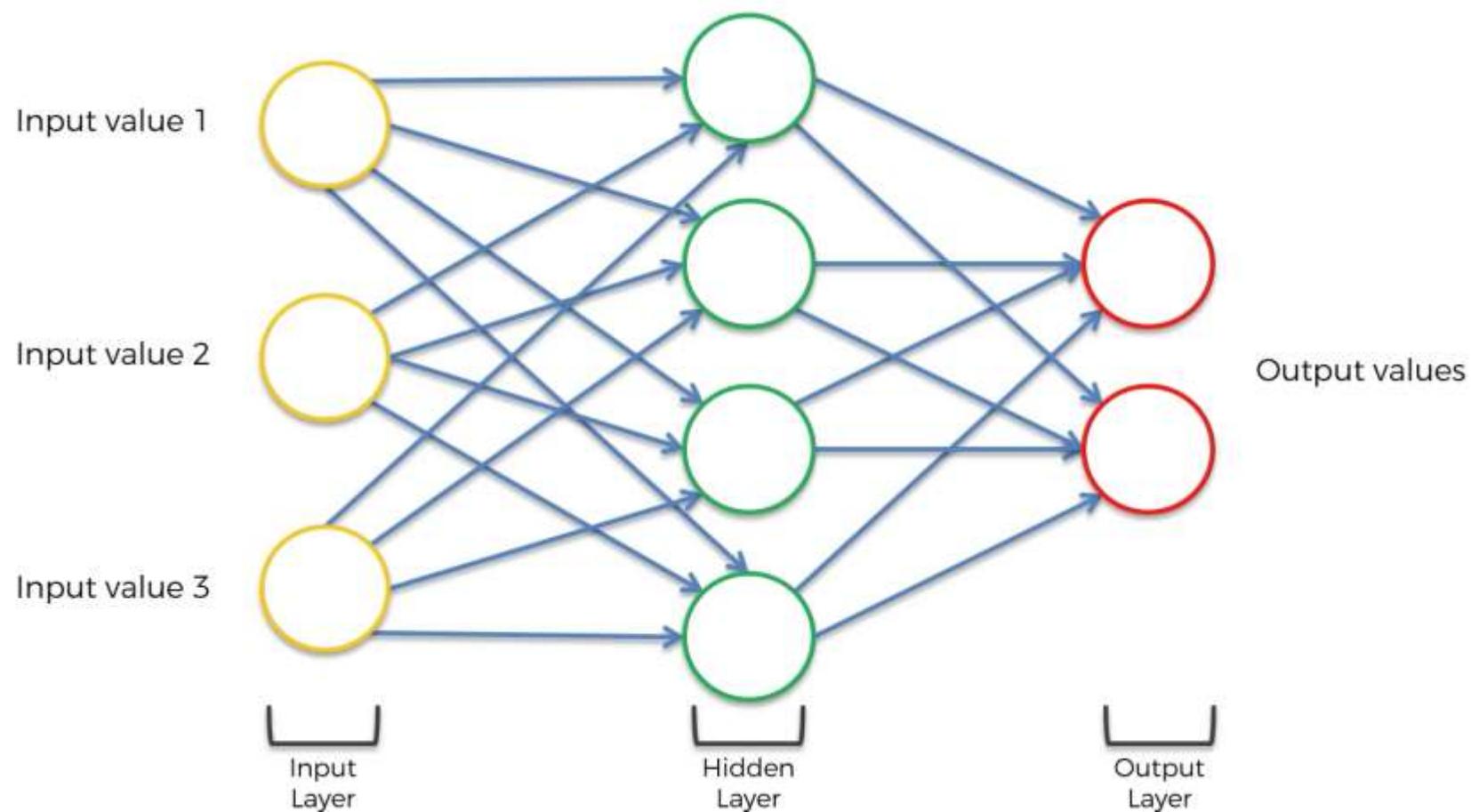


Motivación

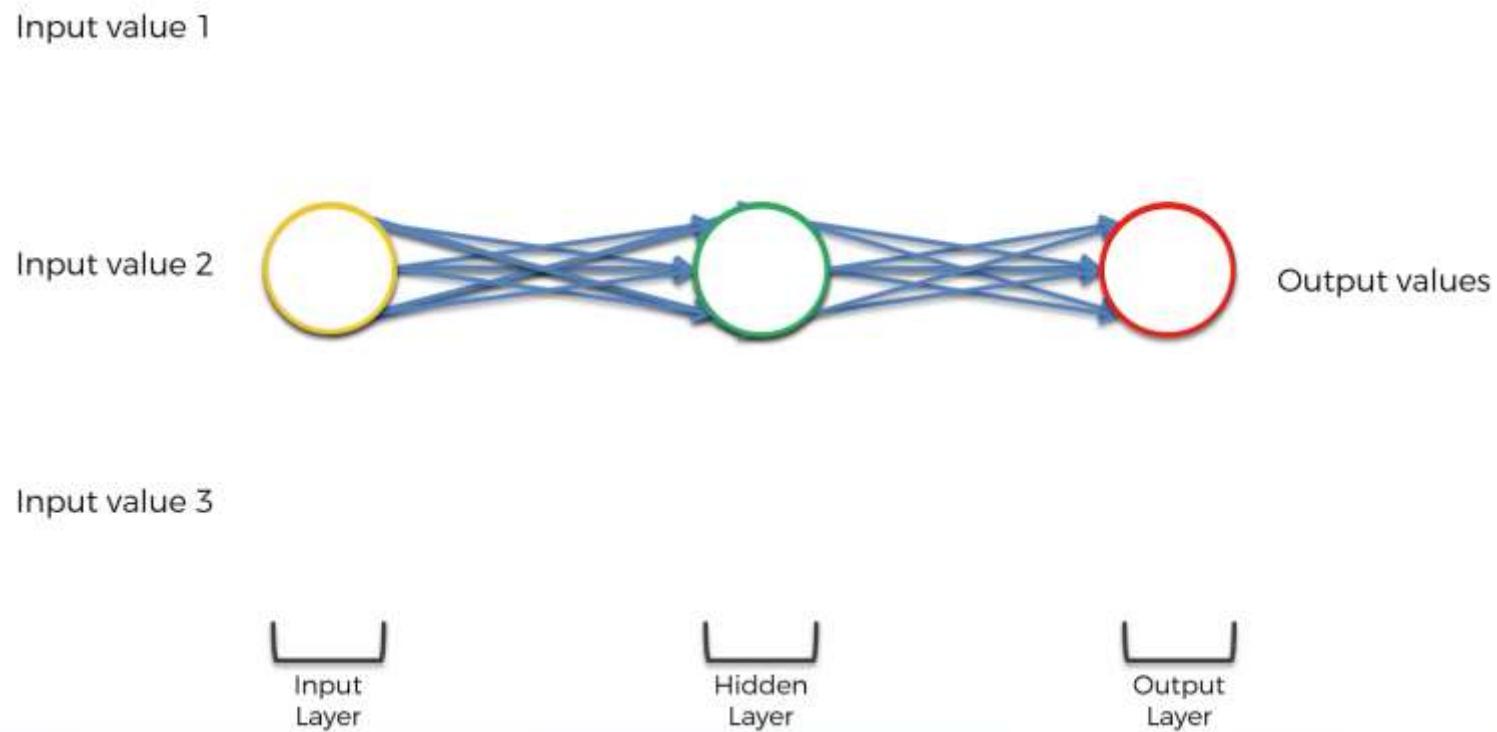
Dimensión de las imágenes



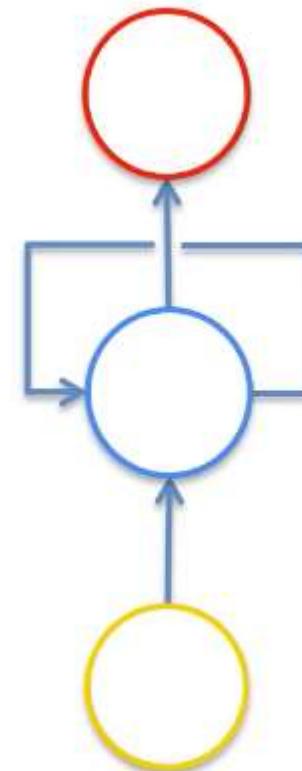
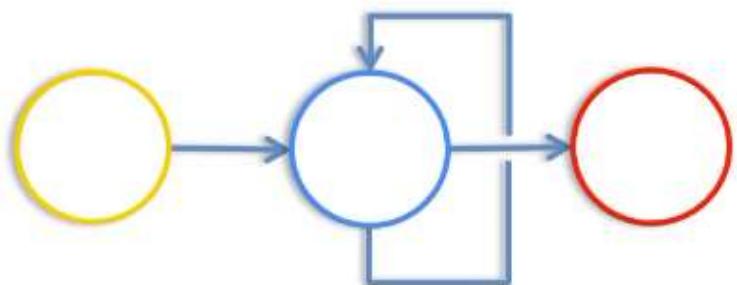
Redes neuronales recurrentes



Redes neuronales recurrentes

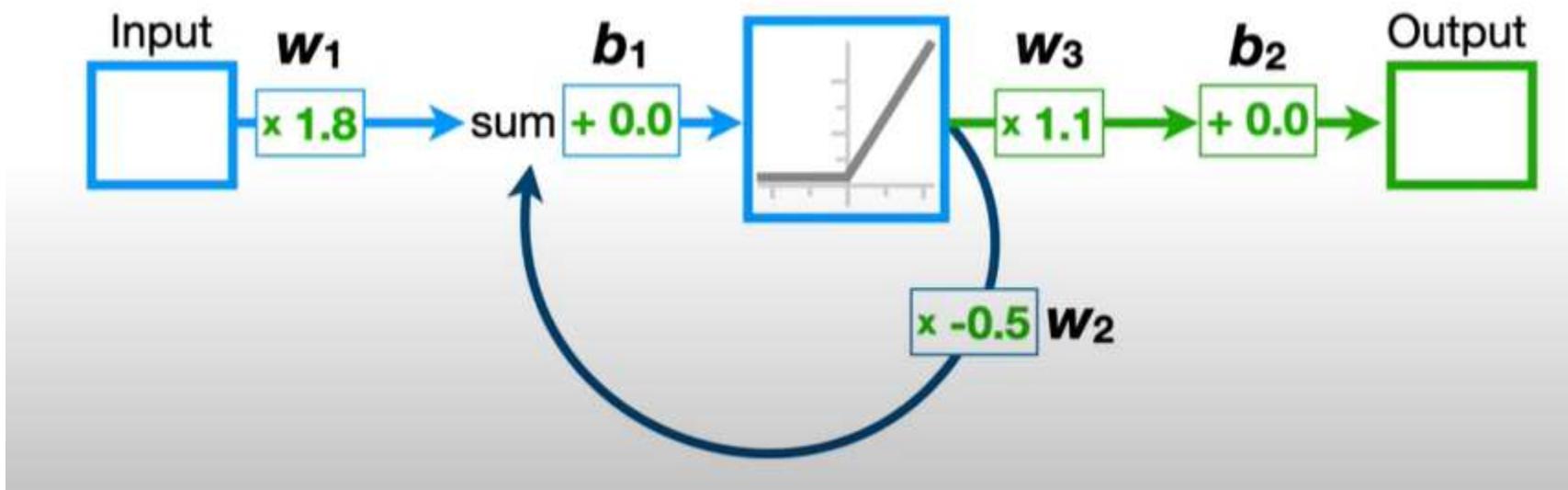


Redes neuronales recurrentes



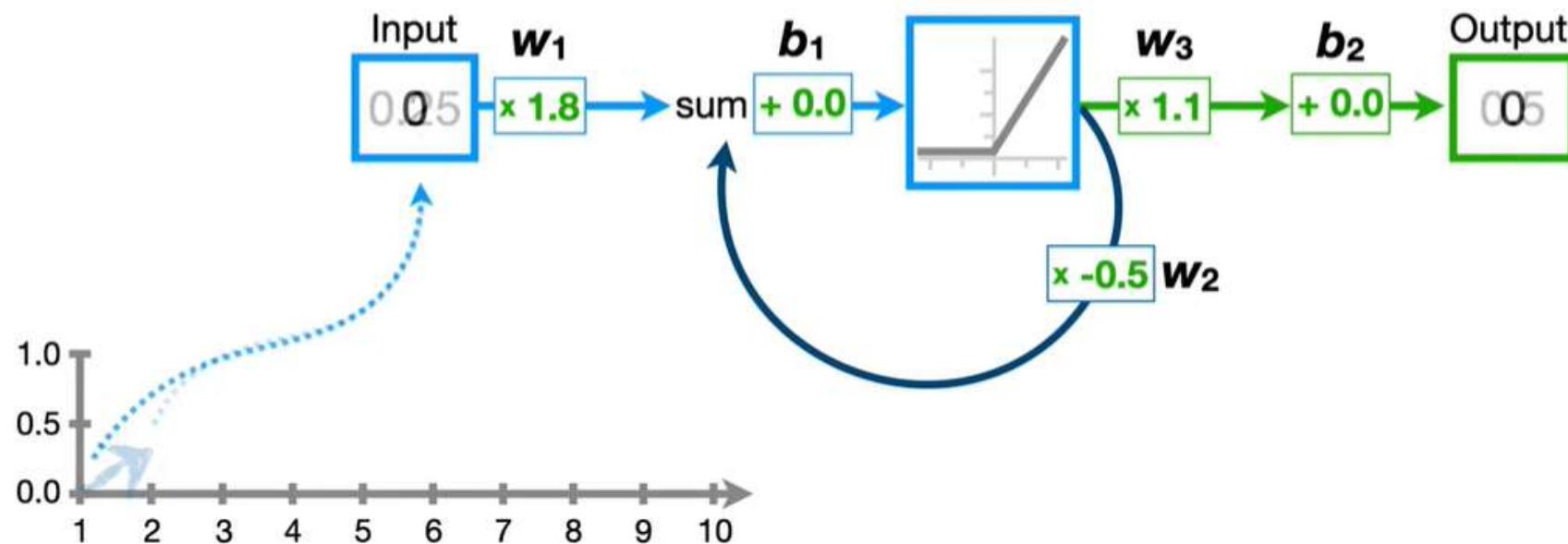
Redes neuronales recurrentes

Just like the other neural networks that we've seen before, **Recurrent Neural Networks have weights, biases,**

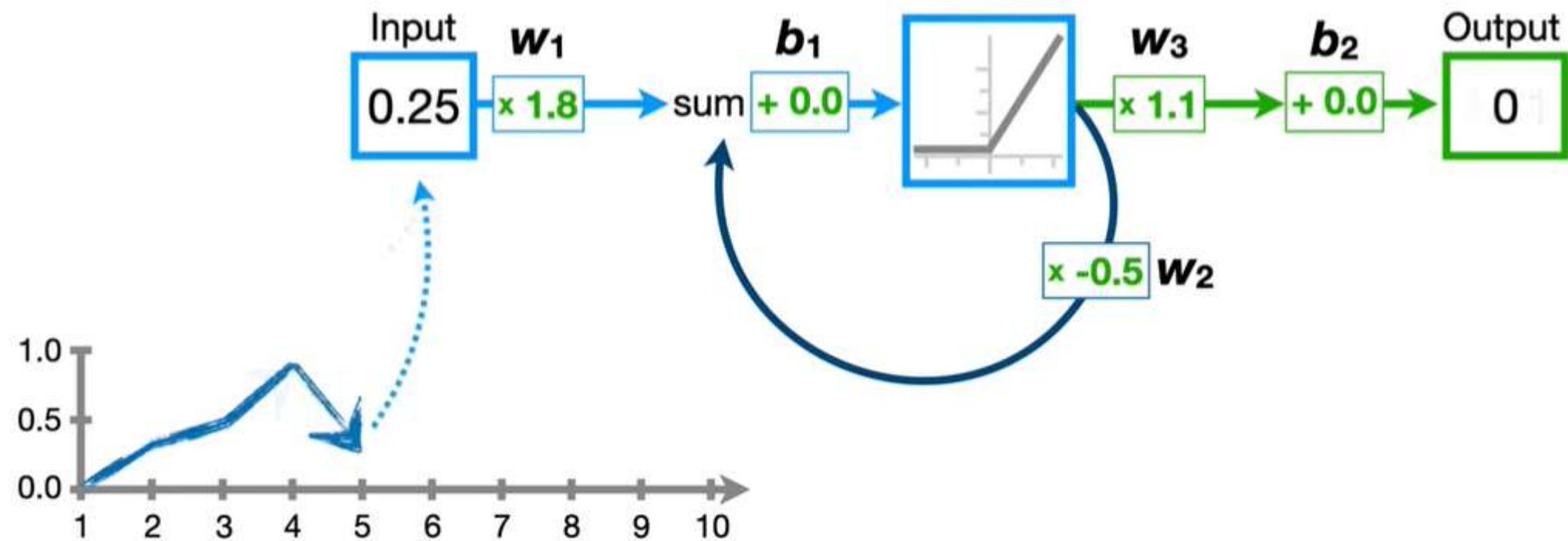


Redes neuronales recurrentes

...the **feedback loop** makes it possible to use *sequential* input values, like stock market prices collected over time, to make predictions.

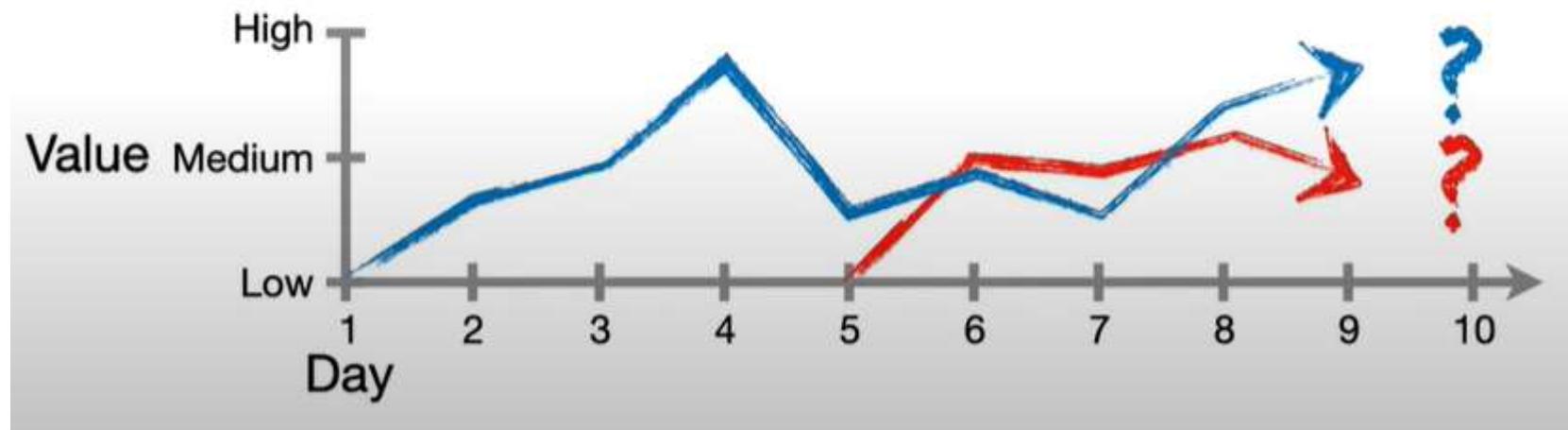


Redes neuronales recurrentes

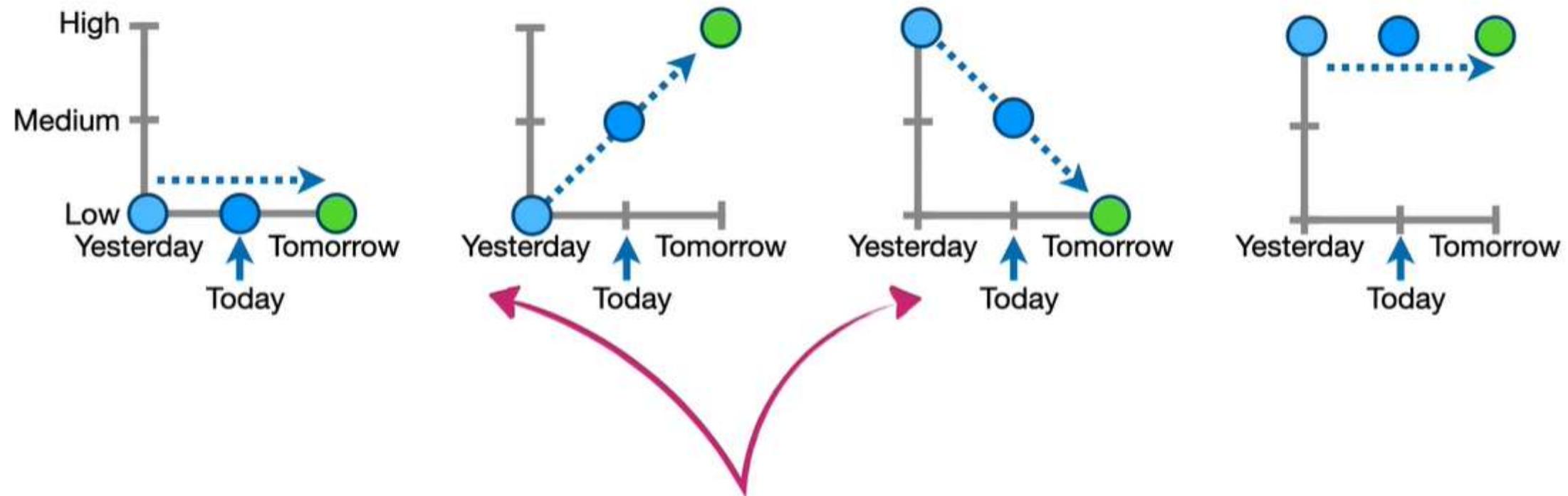


Redes neuronales recurrentes - ejemplo

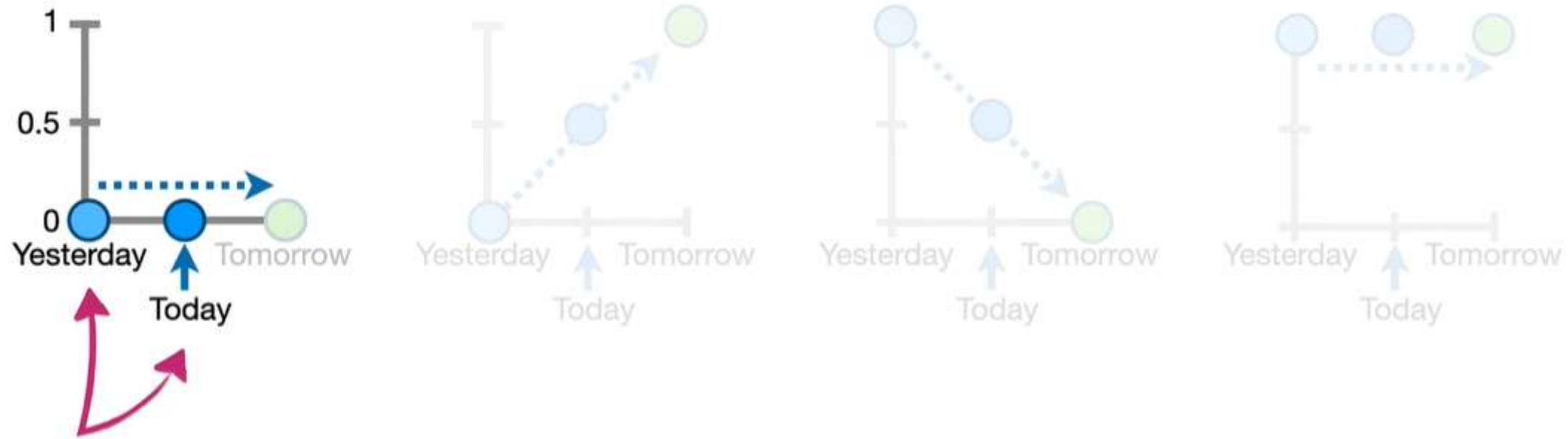
So we need the neural network
to be flexible in terms of how
much sequential data we use
to make a prediction.



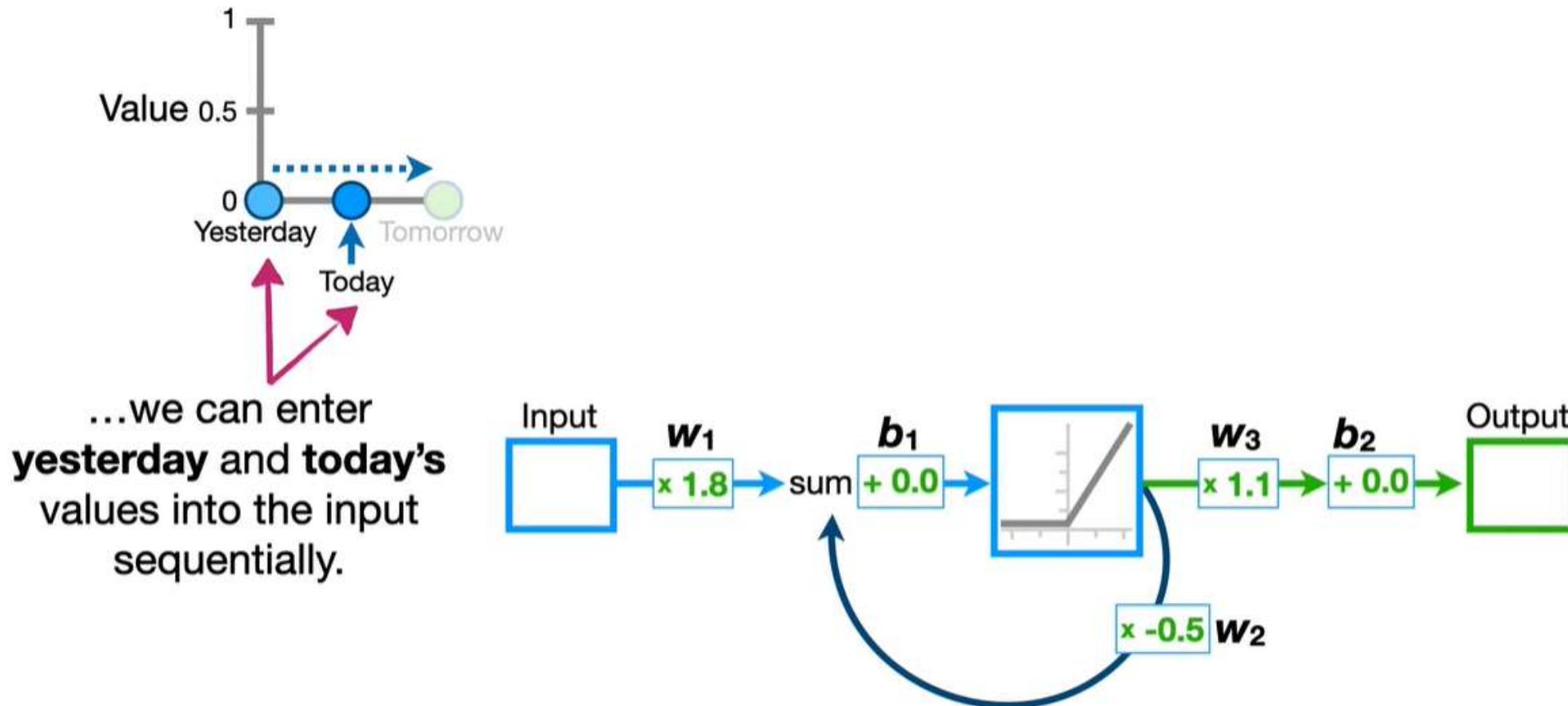
Redes neuronales recurrentes - ejemplo



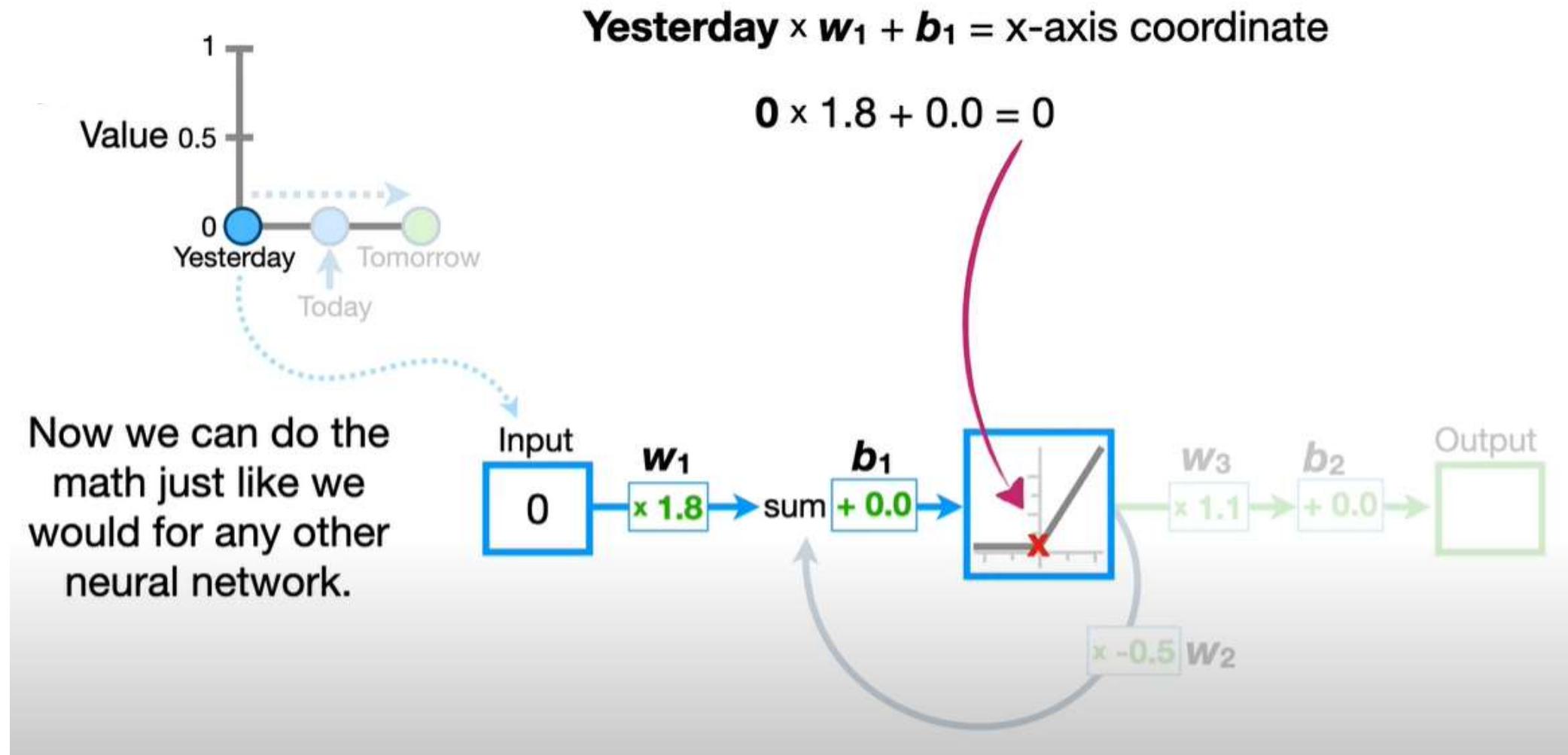
Redes neuronales recurrentes - ejemplo



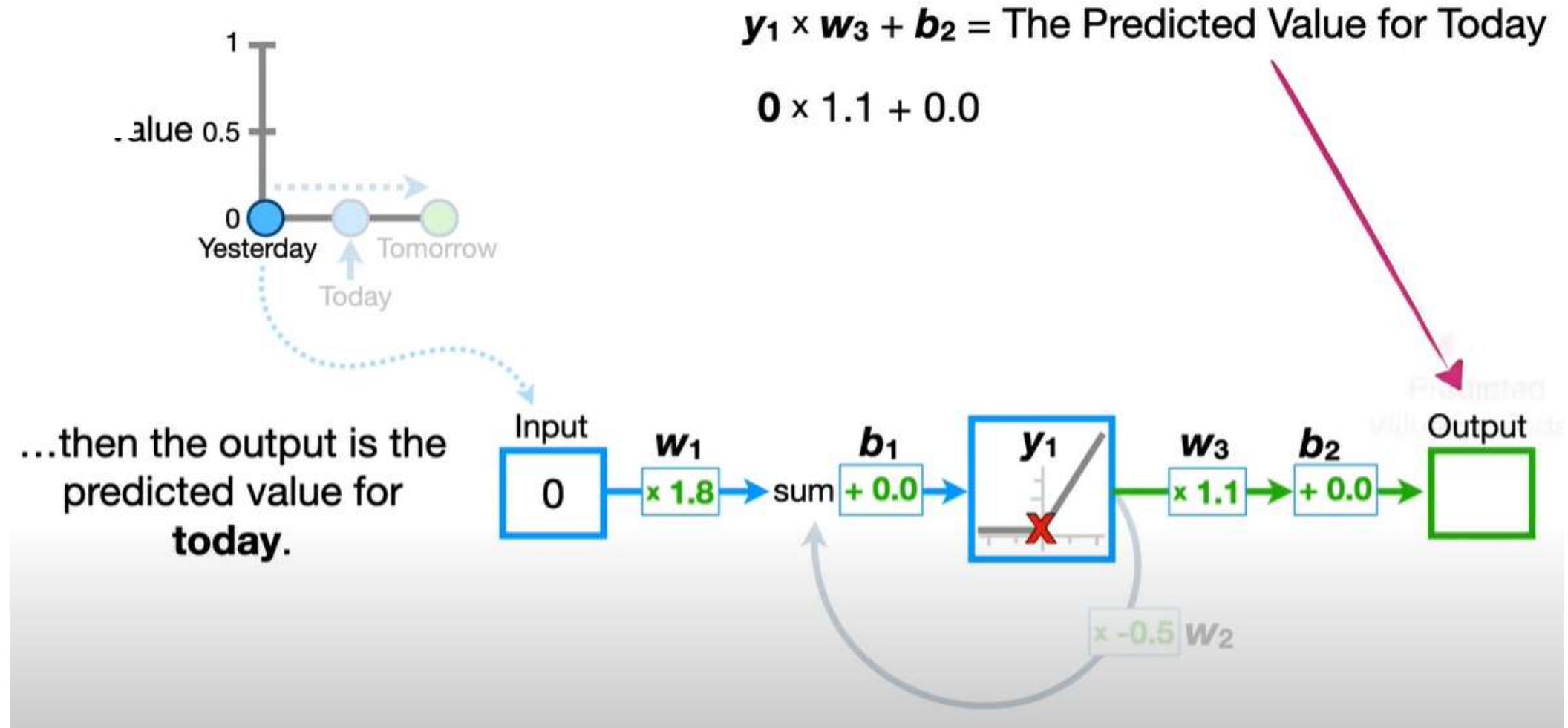
Redes neuronales recurrentes - ejemplo



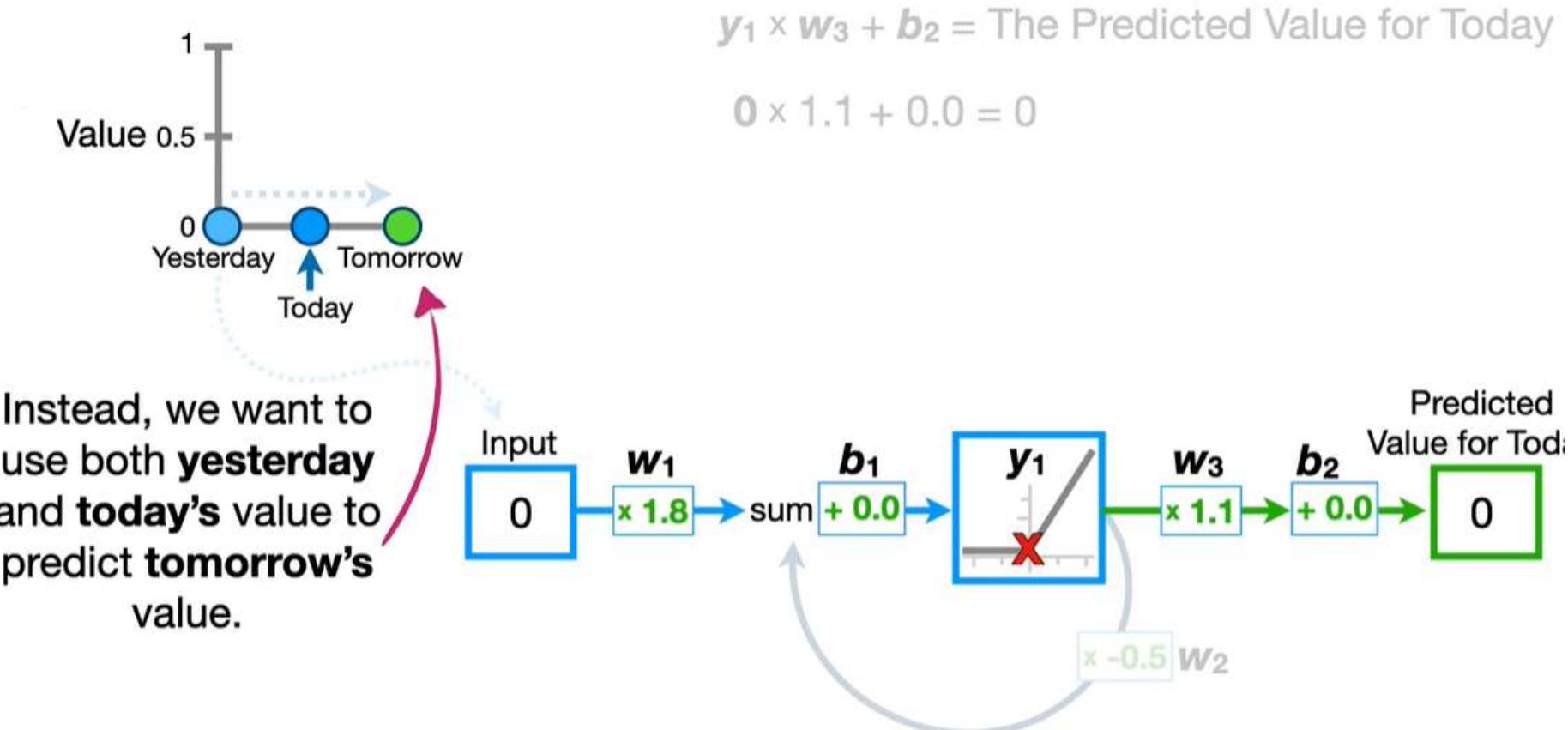
Redes neuronales recurrentes - ejemplo



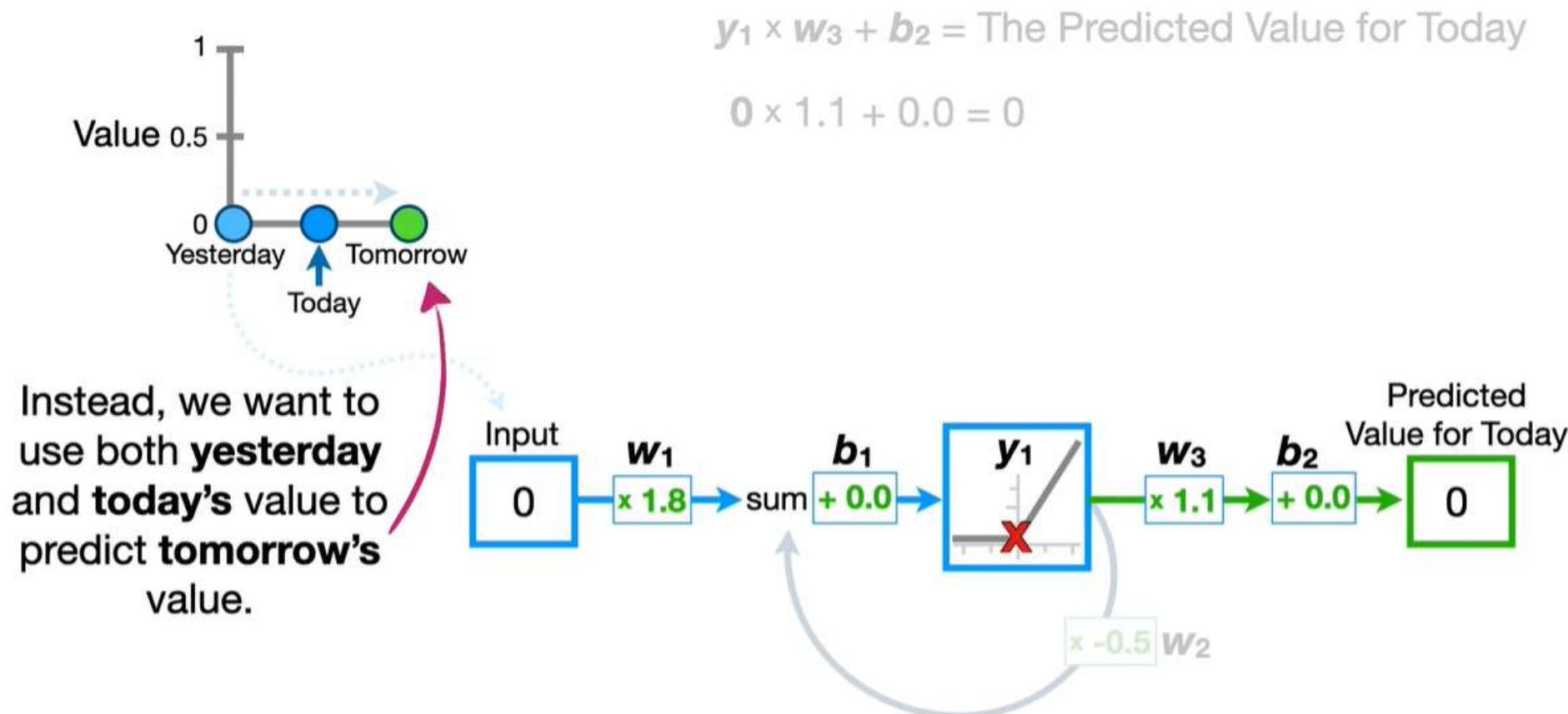
Redes neuronales recurrentes - ejemplo



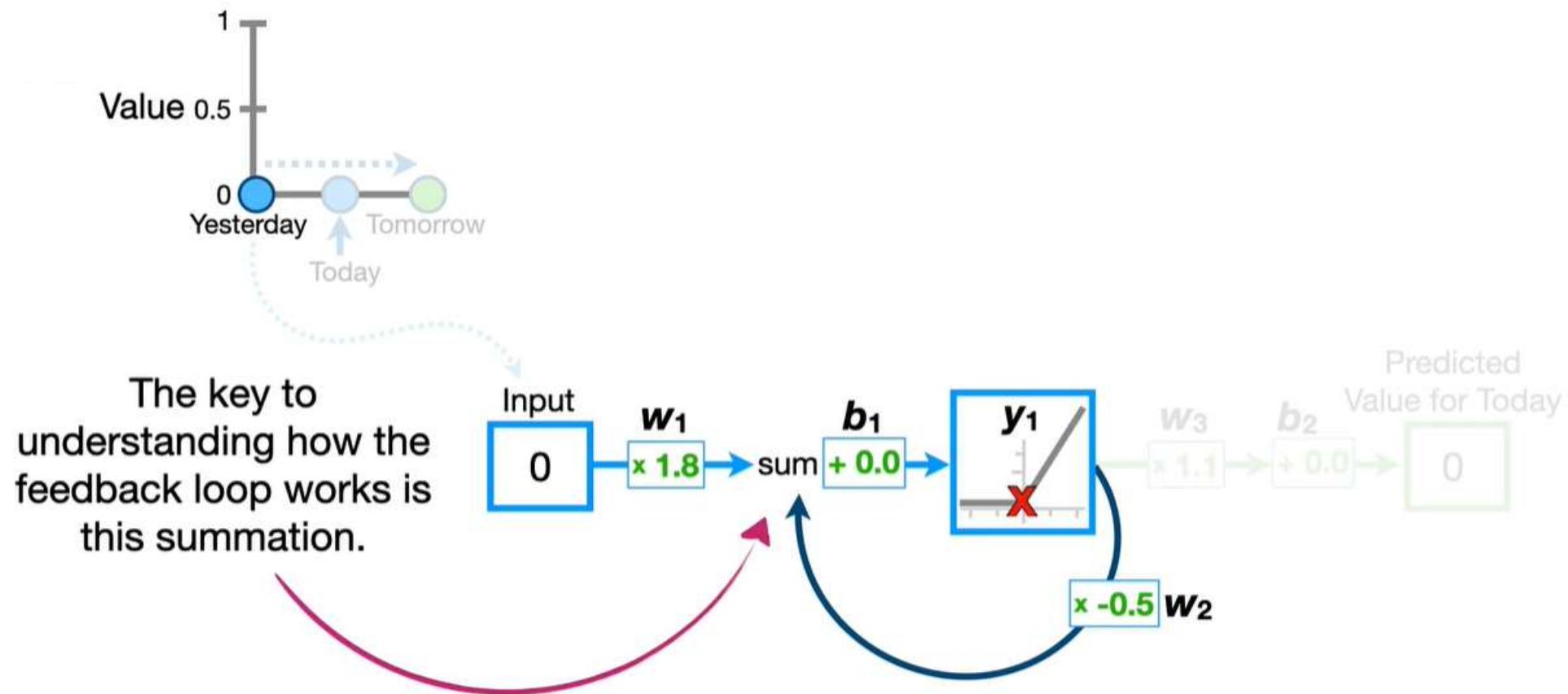
Redes neuronales recurrentes - ejemplo



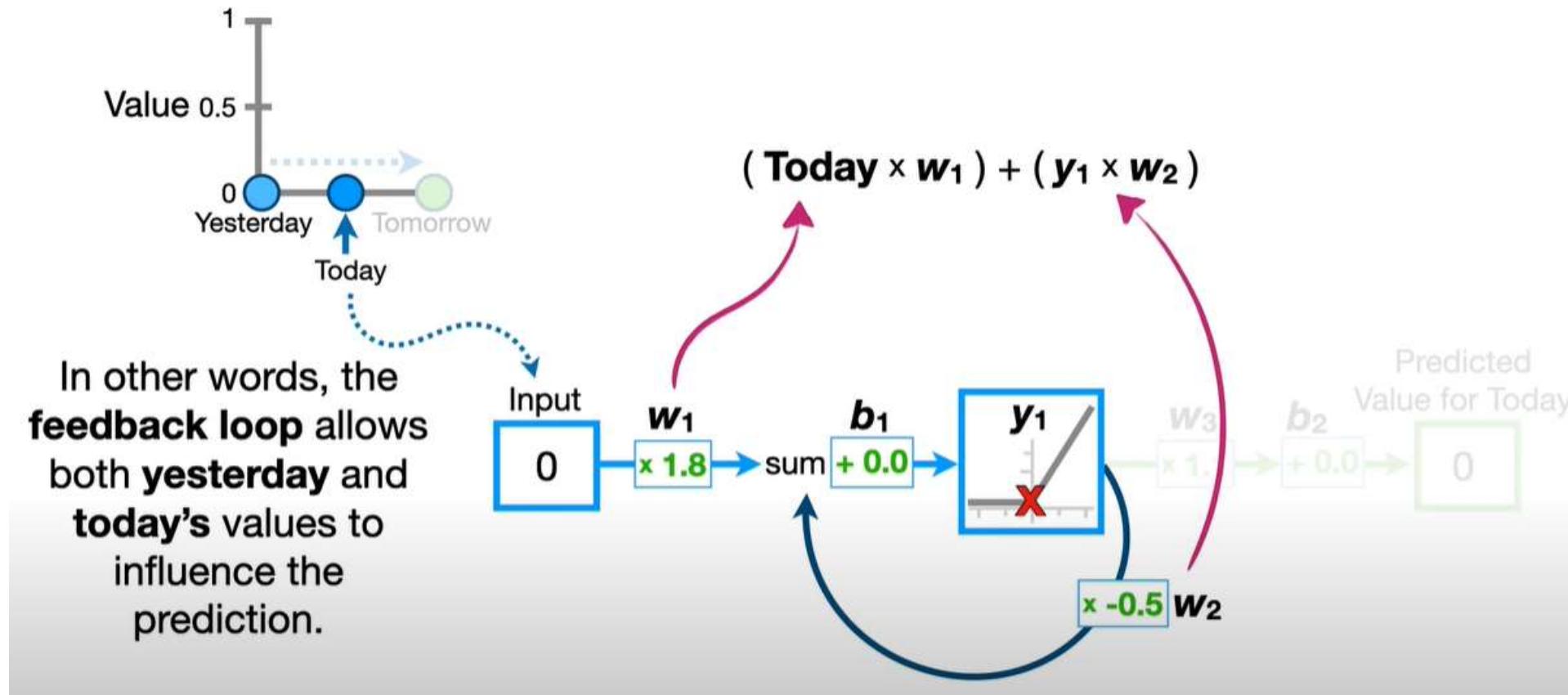
Redes neuronales recurrentes - ejemplo



Redes neuronales recurrentes - ejemplo



Redes neuronales recurrentes - ejemplo



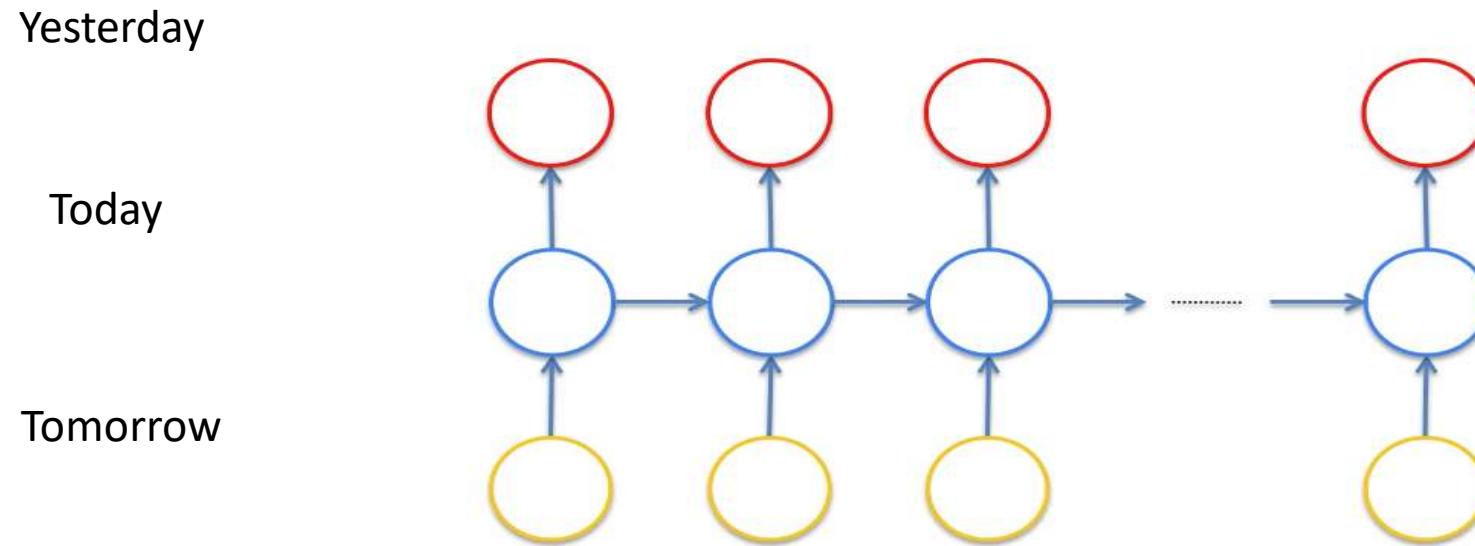
Redes neuronales recurrentes - ejemplo

Recurrence - Recursivity

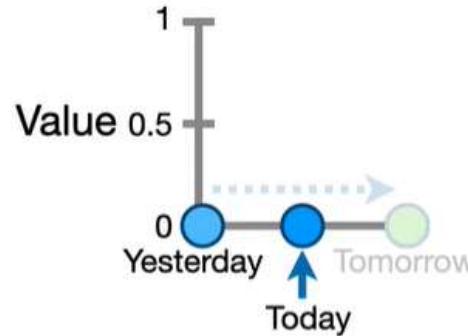
for $T = 3$: $\mathbf{h}_3 = f_{\theta}(\mathbf{h}_2, \mathbf{x}_3)$

$$\begin{aligned} &= f_{\theta}(f_{\theta}(\mathbf{h}_1, \mathbf{x}_2), \mathbf{x}_3) \\ &= f_{\theta}(f_{\theta}(f_{\theta}(\mathbf{h}_0, \mathbf{x}_1), \mathbf{x}_2), \mathbf{x}_3) \end{aligned}$$

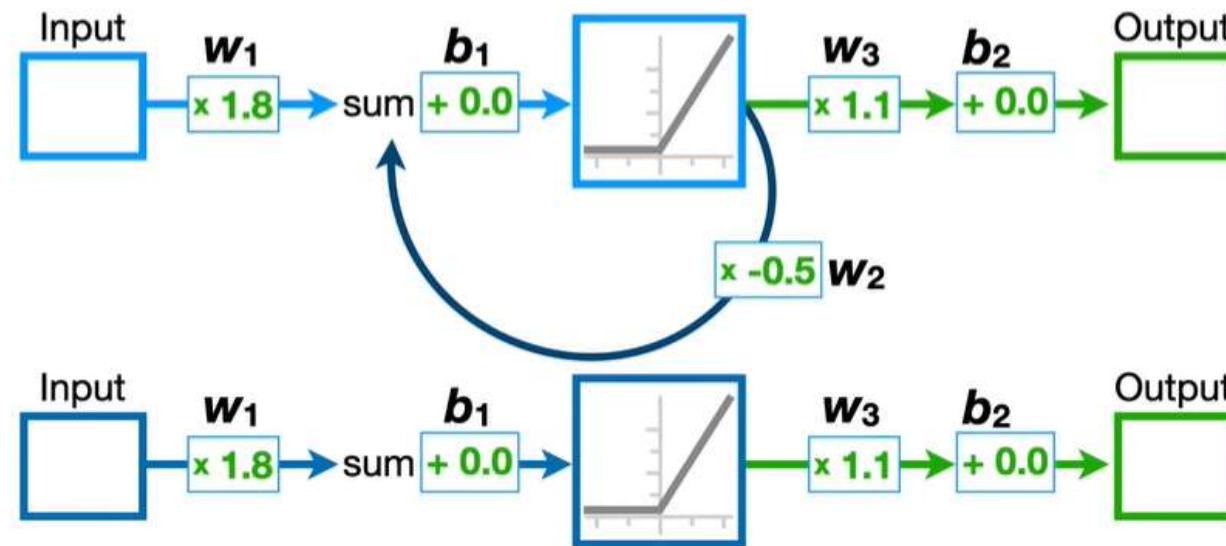
Redes neuronales recurrentes - ejemplo



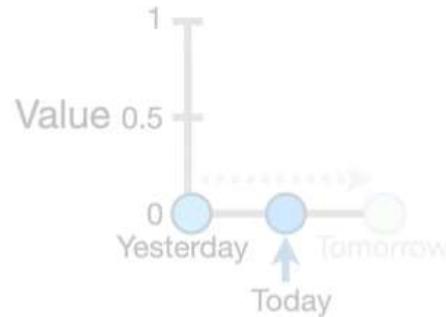
Redes neuronales recurrentes - ejemplo



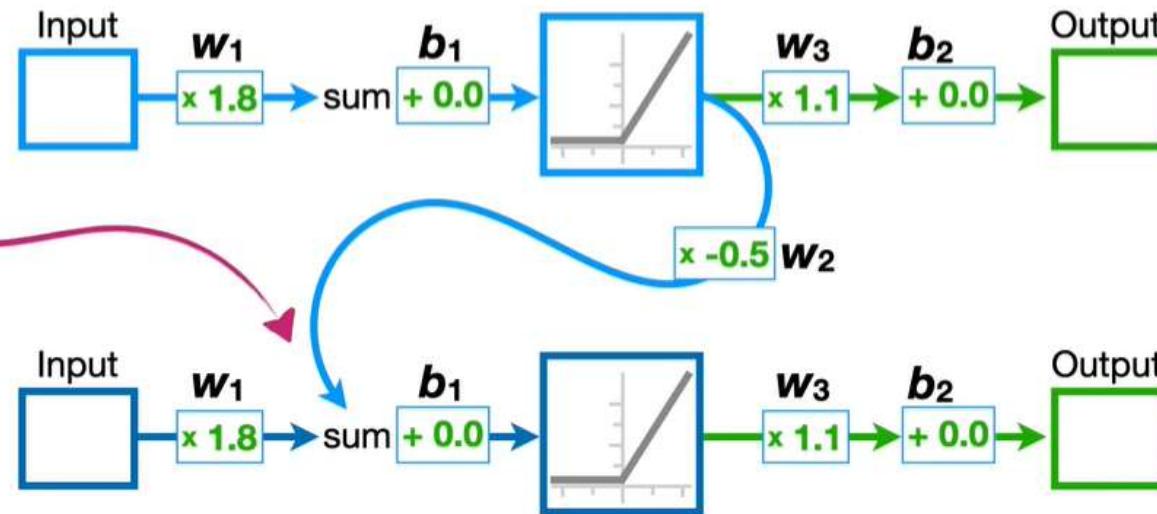
...we can **unroll** the feedback loop by making a copy of the neural network for each input value.



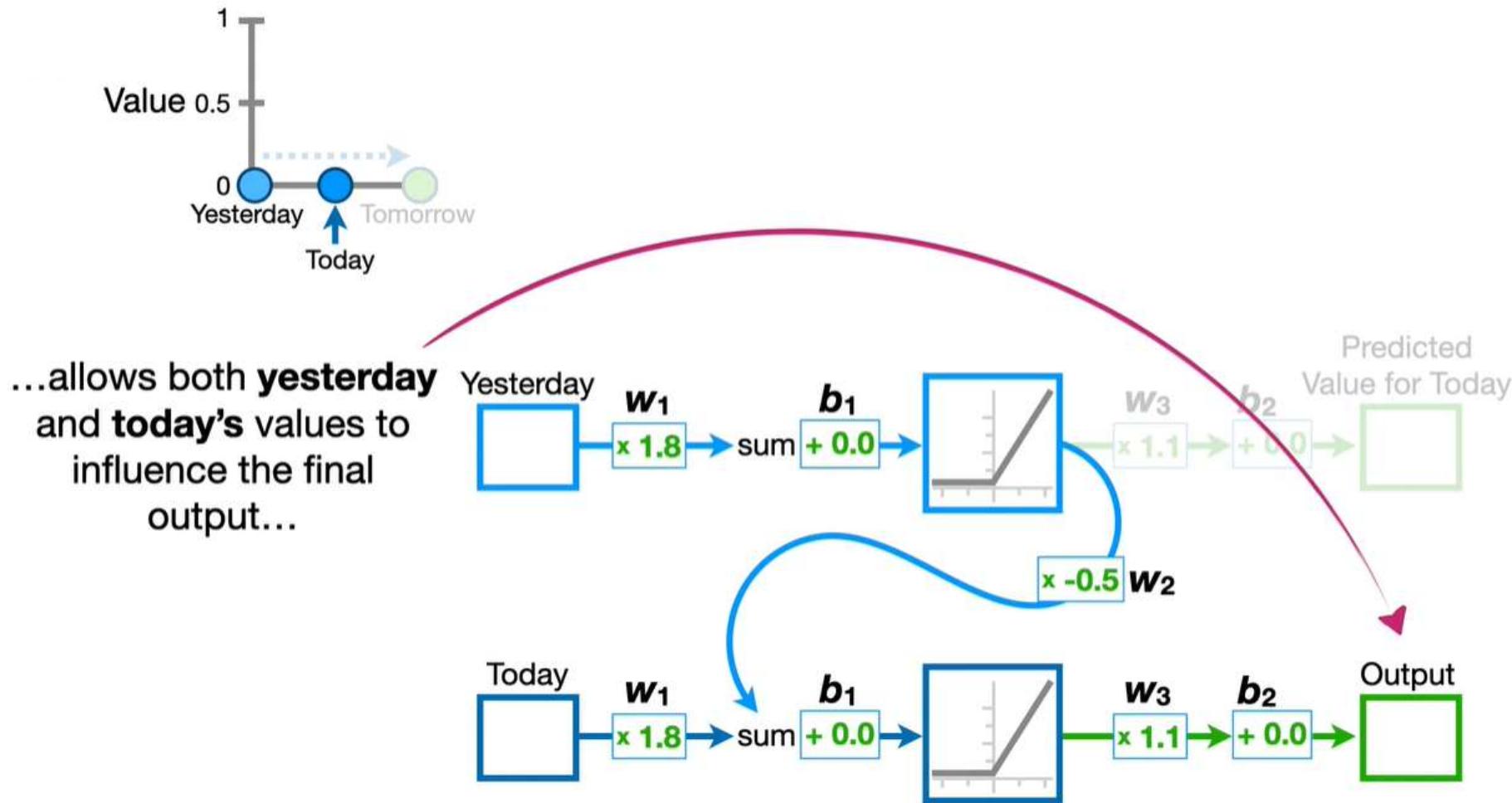
Redes neuronales recurrentes - ejemplo



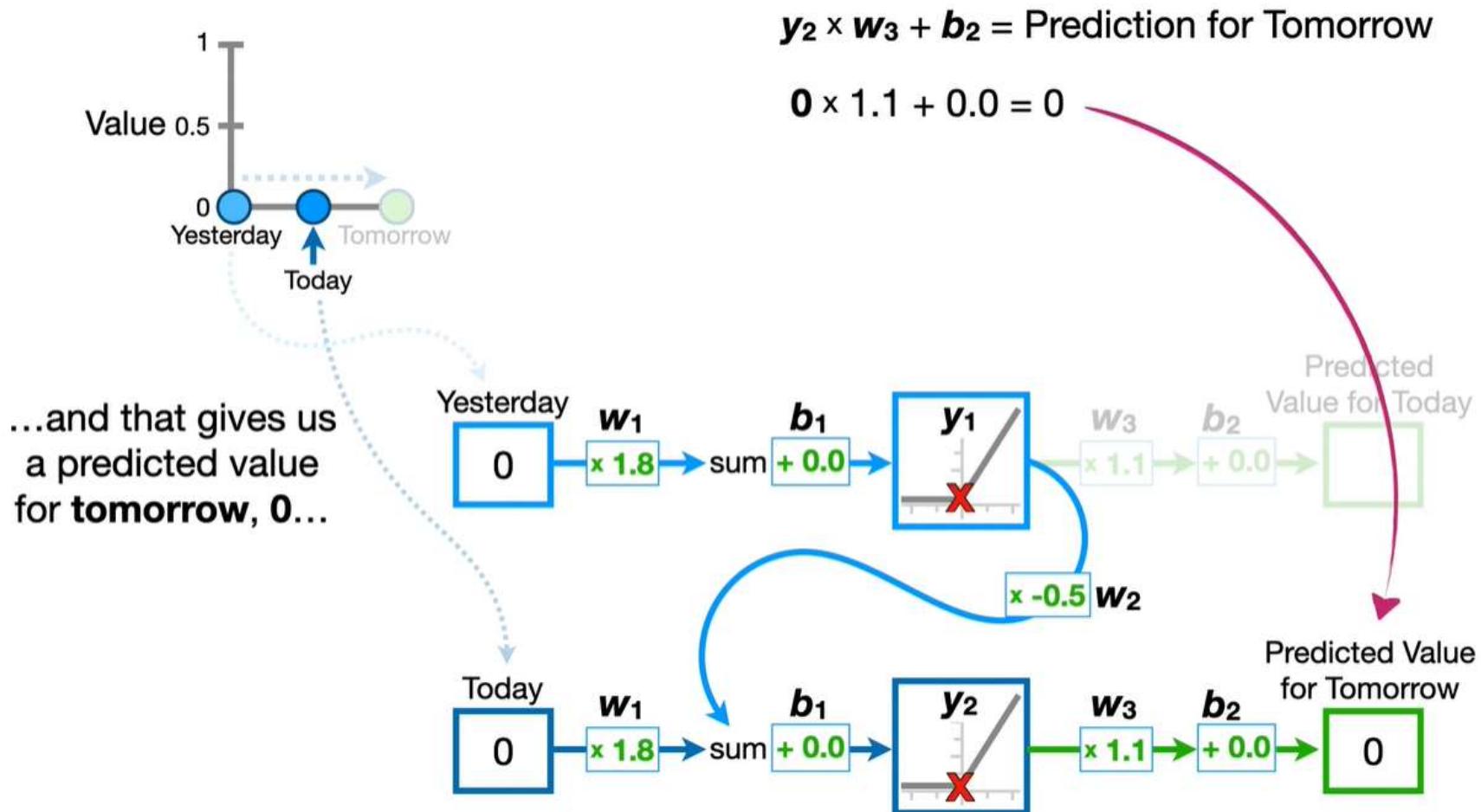
...we can point it to
the sum in the second
copy.



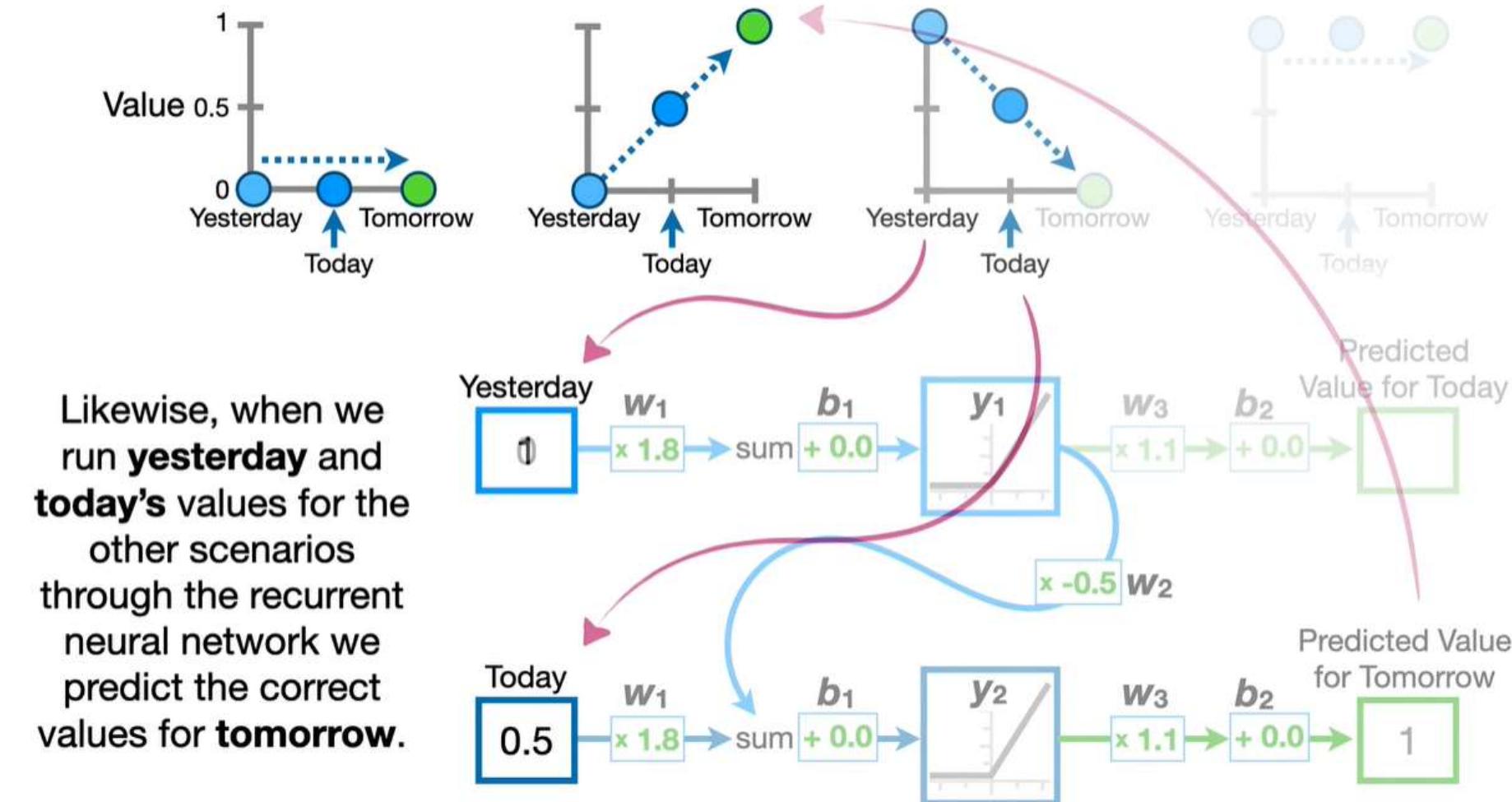
Redes neuronales recurrentes - ejemplo



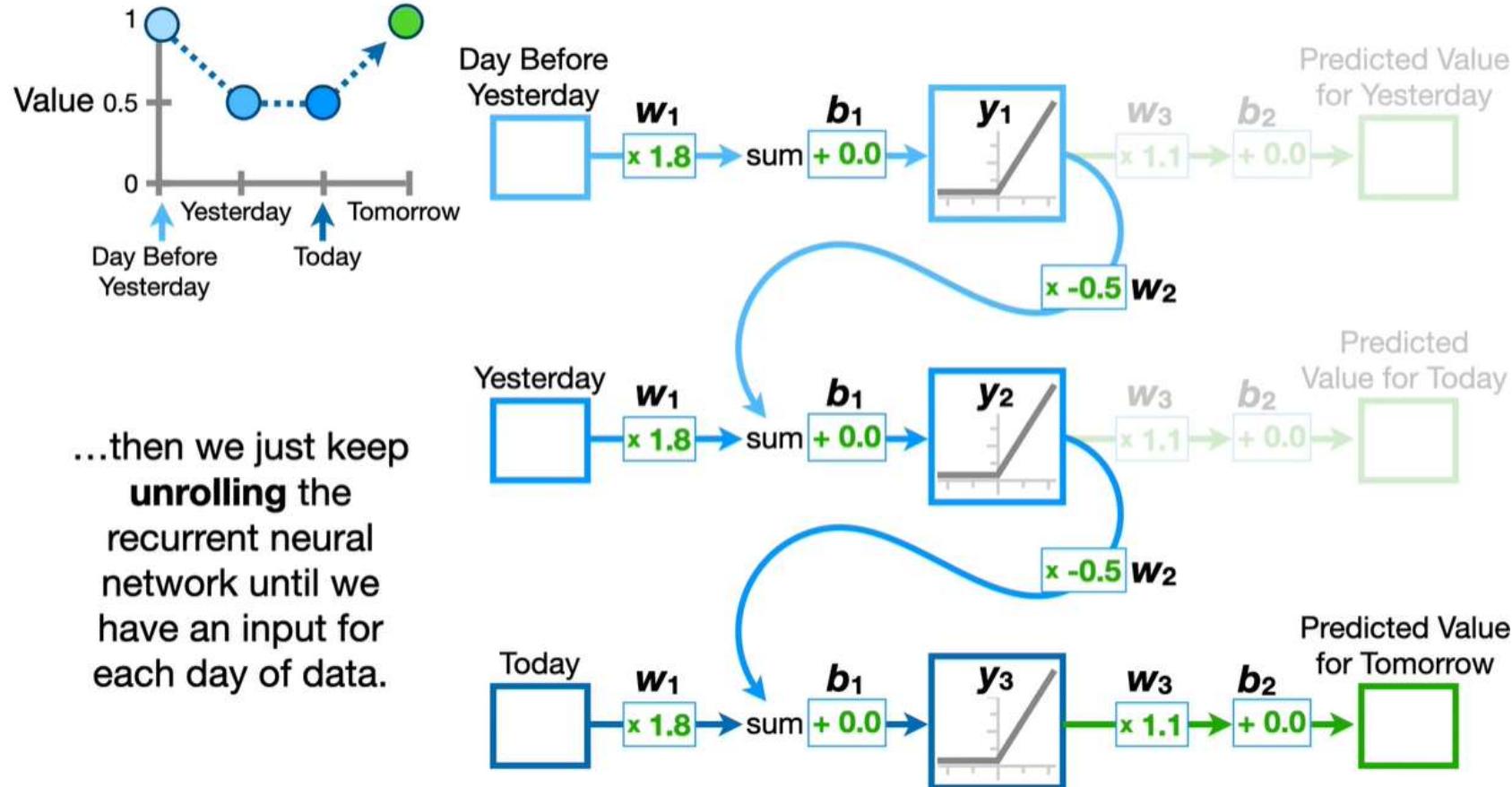
Redes neuronales recurrentes - ejemplo



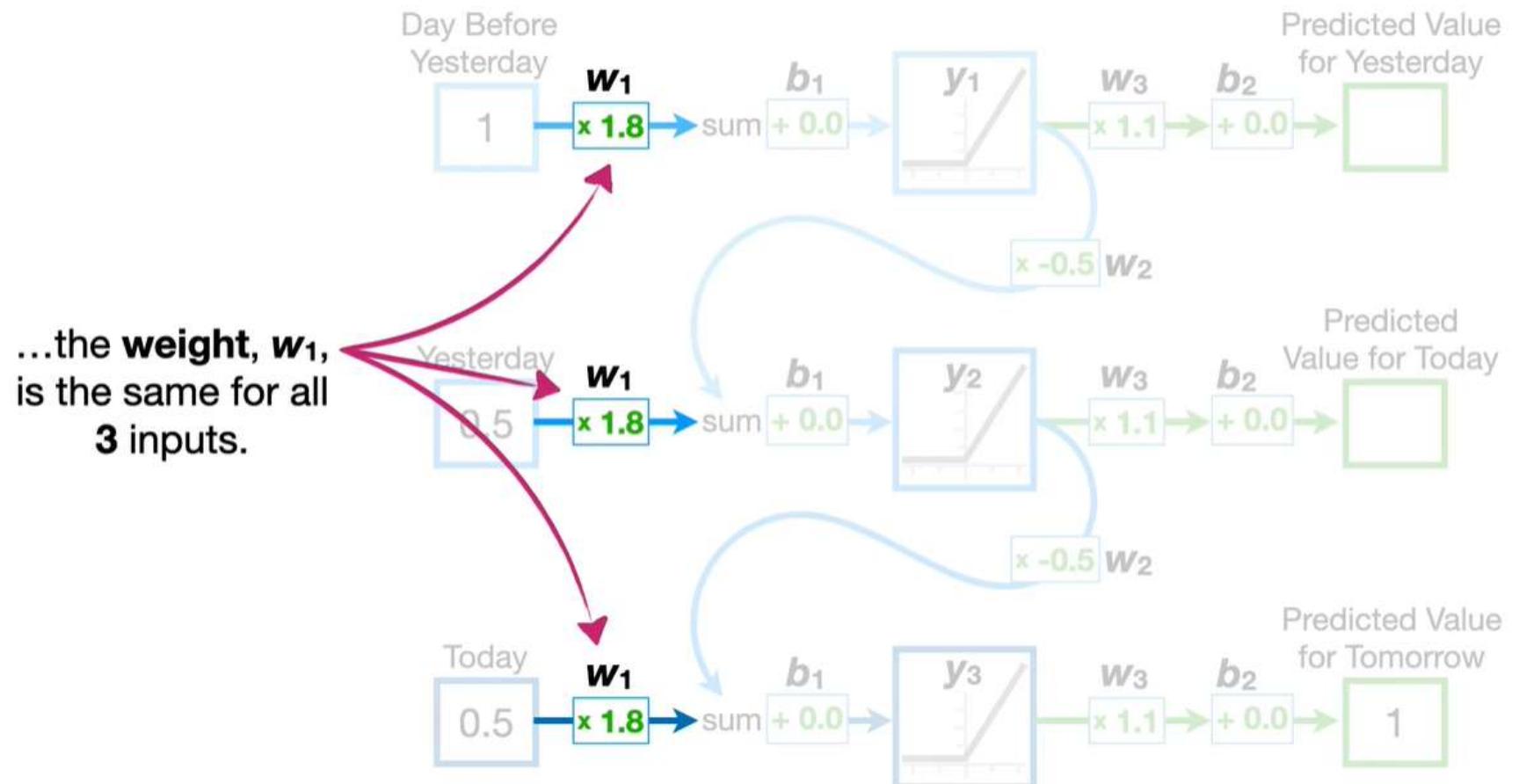
Redes neuronales recurrentes - ejemplo



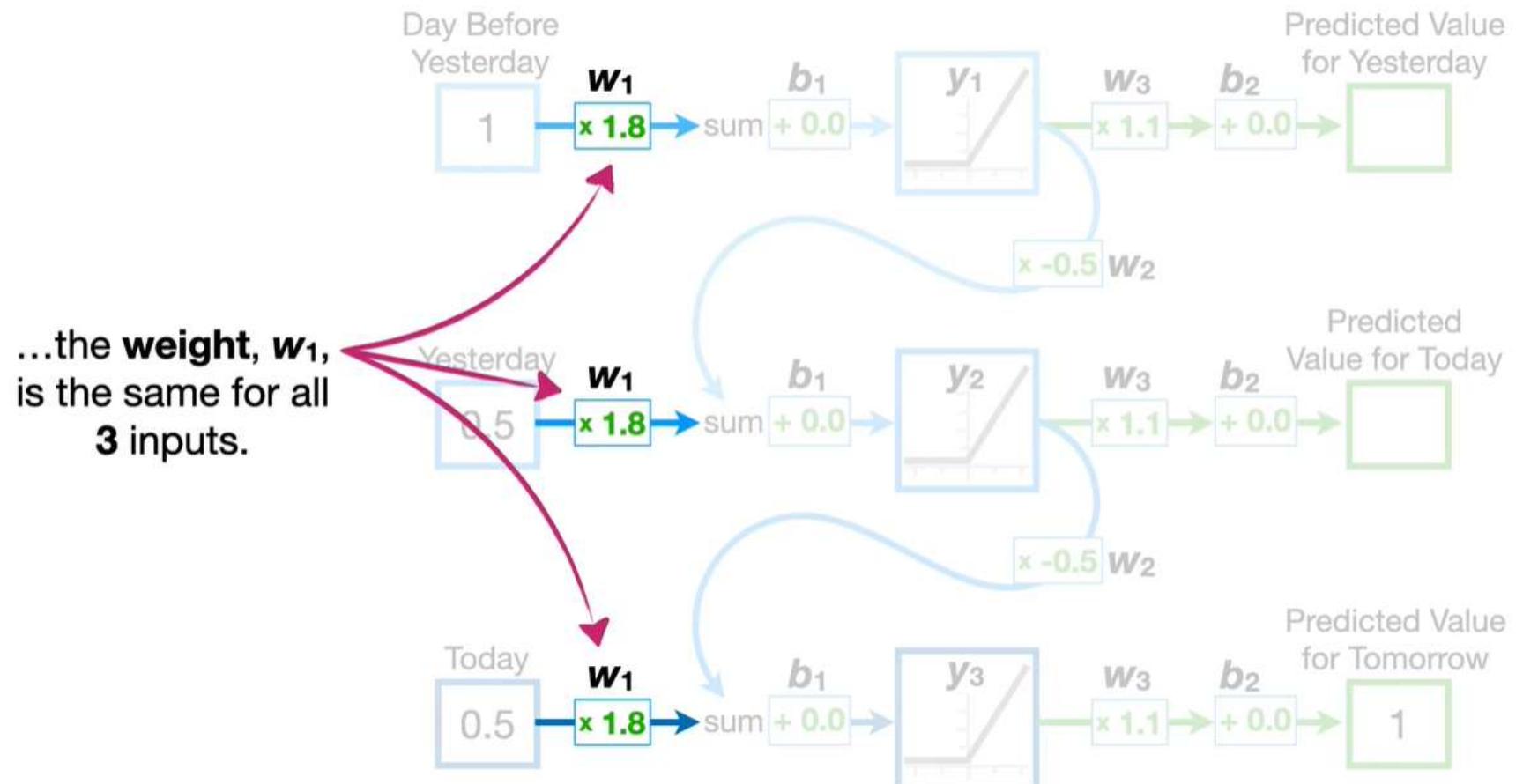
Redes neuronales recurrentes - ejemplo



Redes neuronales recurrentes - ejemplo



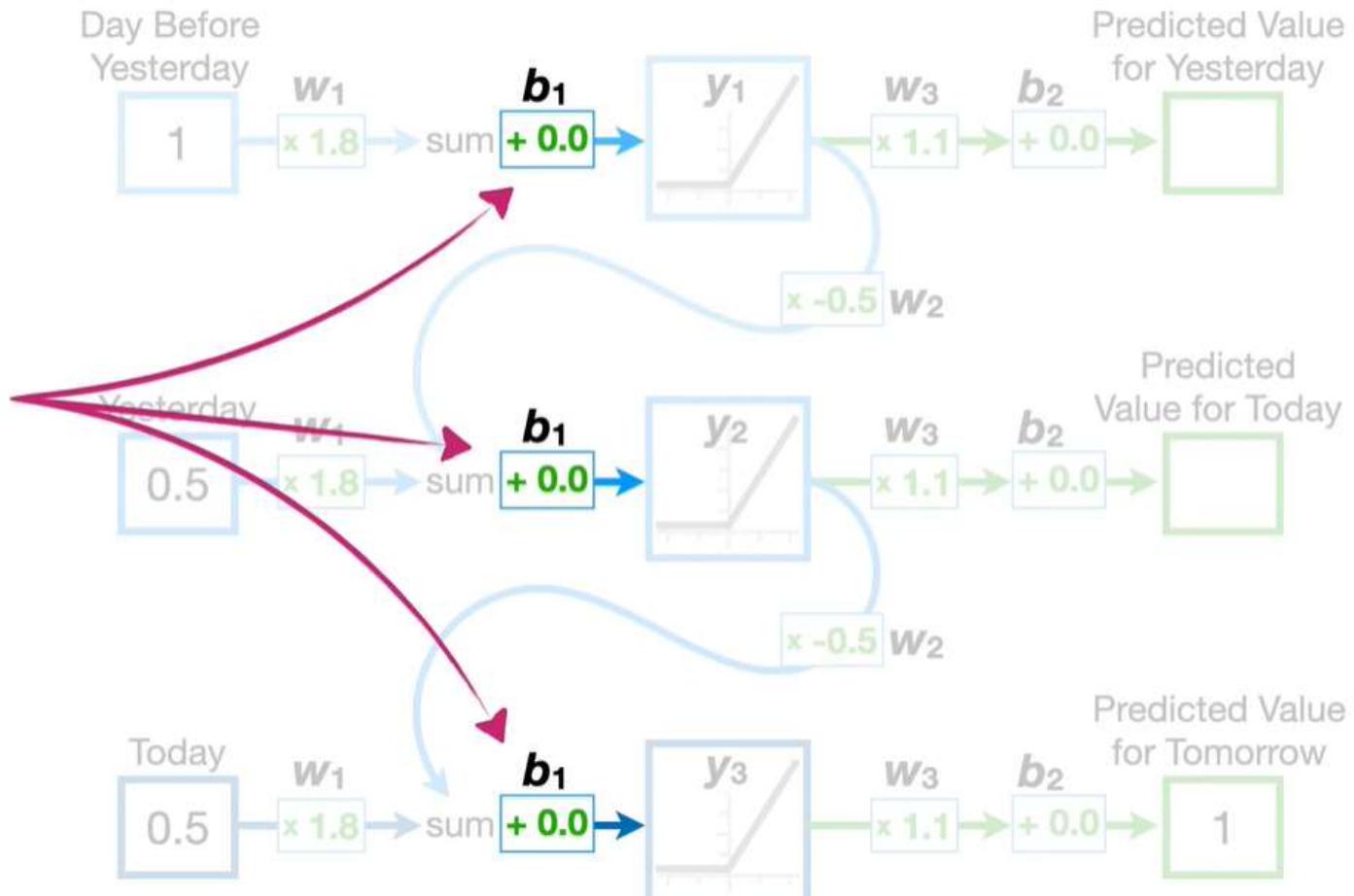
Redes neuronales recurrentes - ejemplo



Redes neuronales recurrentes - ejemplo



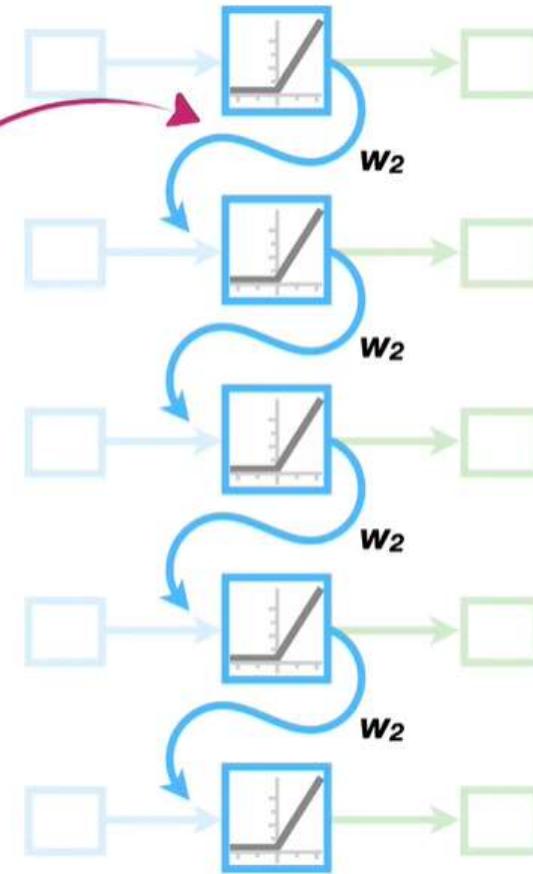
And the **bias**, b_1 ,
is also the same
for all 3 inputs.



Problema de las redes neuronales recurrentes

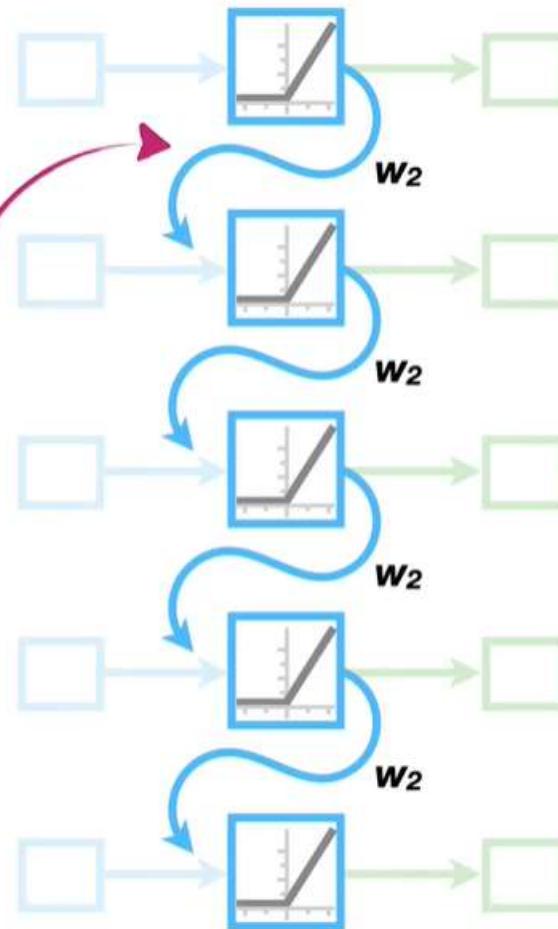


In our example, **The Vanishing/Exploding Gradient problem** has to do with the **weight** along the squiggle that we copy each time we **unroll** the network.

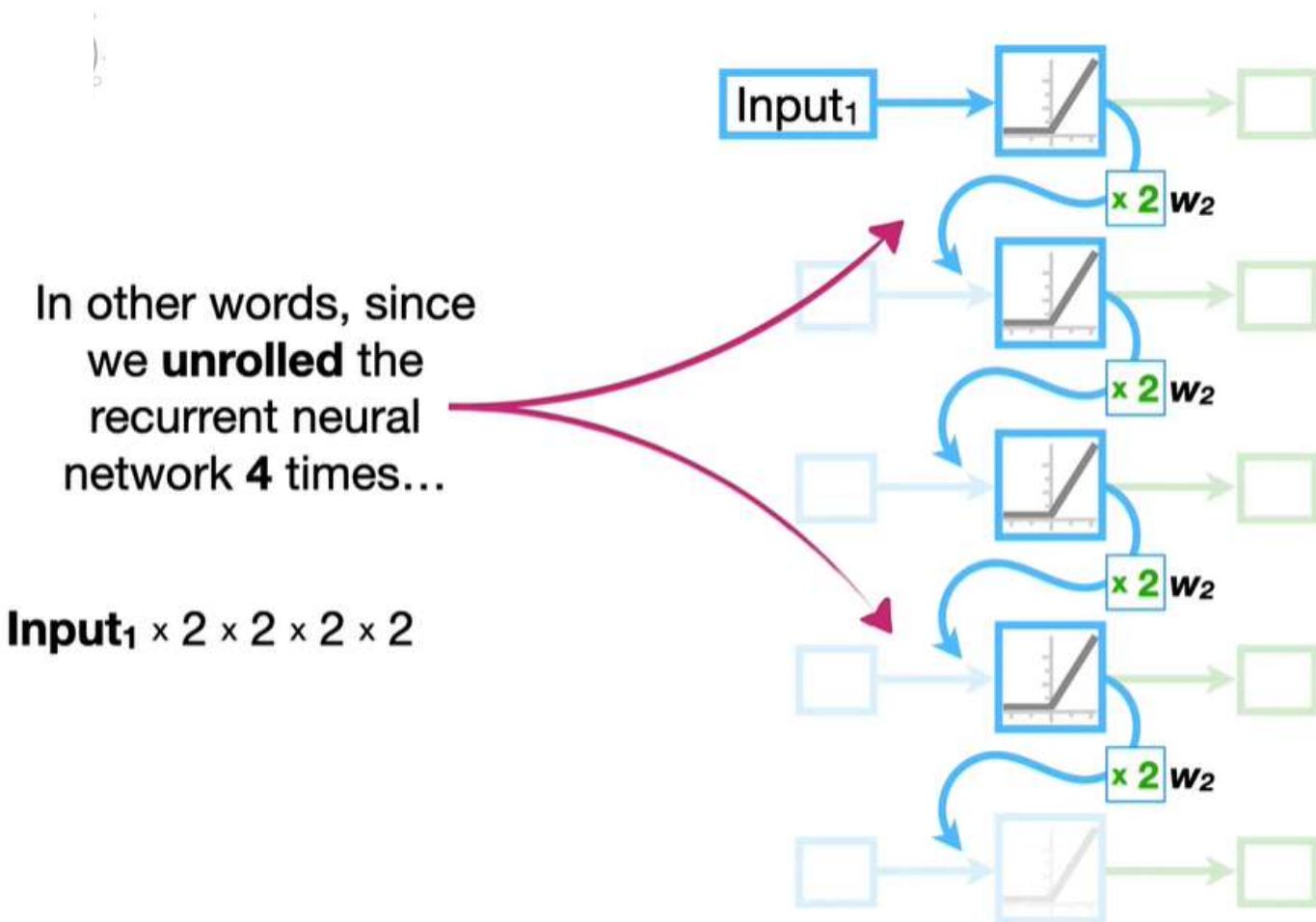


Problema de las redes neuronales recurrentes

In our example, the gradient will **explode**, when we set w_2 to any value larger than 1.



Problema de las redes neuronales recurrentes



Problema de las redes neuronales recurrentes

...we multiply the input value by w_2 , which is 2, raised to the number of times we **unrolled**, which is 4.

$$\begin{aligned} \text{Input}_1 \times 2 \times 2 \times 2 \times 2 \\ = \text{Input}_1 \times 2^4 \\ = \text{Input}_1 \times w_2^{\text{Num. Unroll}} \end{aligned}$$



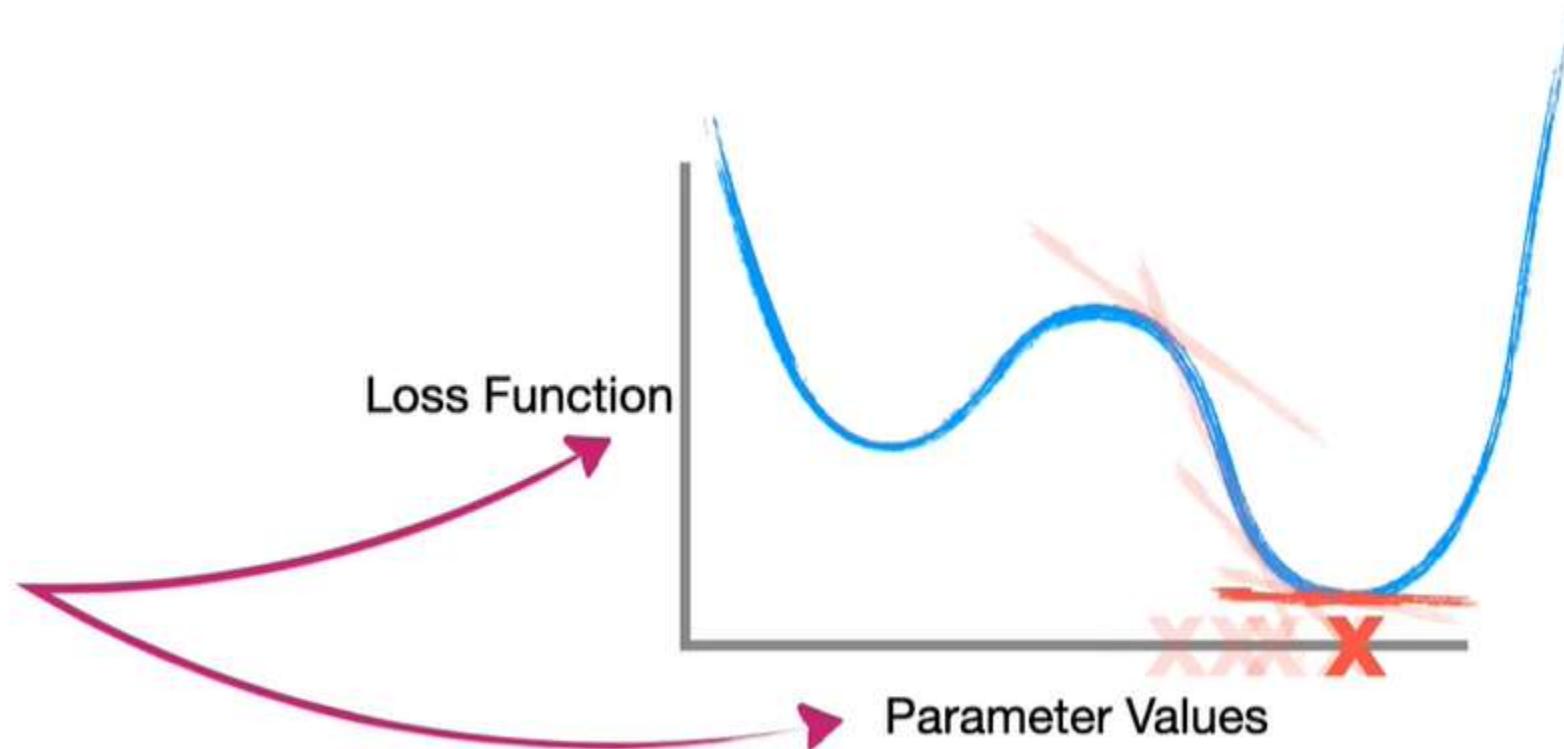
Problema de las redes neuronales recurrentes

If we tried to train this recurrent neural network with backpropagation, this **HUGE NUMBER** would find its way into some of the gradients...

$$\frac{d \text{SSR}}{d w_1} = \frac{d \text{SSR}}{d \text{Predicted}} \times \frac{d \text{Predicted}}{d y_1} \times \frac{d y_1}{d x_1} \times \frac{d x_1}{d w_1} + \dots$$
$$\frac{d x_1}{d w_1} = \frac{d}{d w_1} \text{Input1} \times w_1 \times w_2^{50} \dots$$
$$= \frac{d}{d w_1} \text{Input1} \times w_1 \times \text{A HUGE NUMBER} \dots$$
$$= \text{Input}_1 \times 2^{50} = \text{Input}_1 \times \text{A HUGE NUMBER}$$
$$= \text{Input}_1 \times w_2^{\text{Num. Unroll}}$$

Problema de las redes neuronales recurrentes

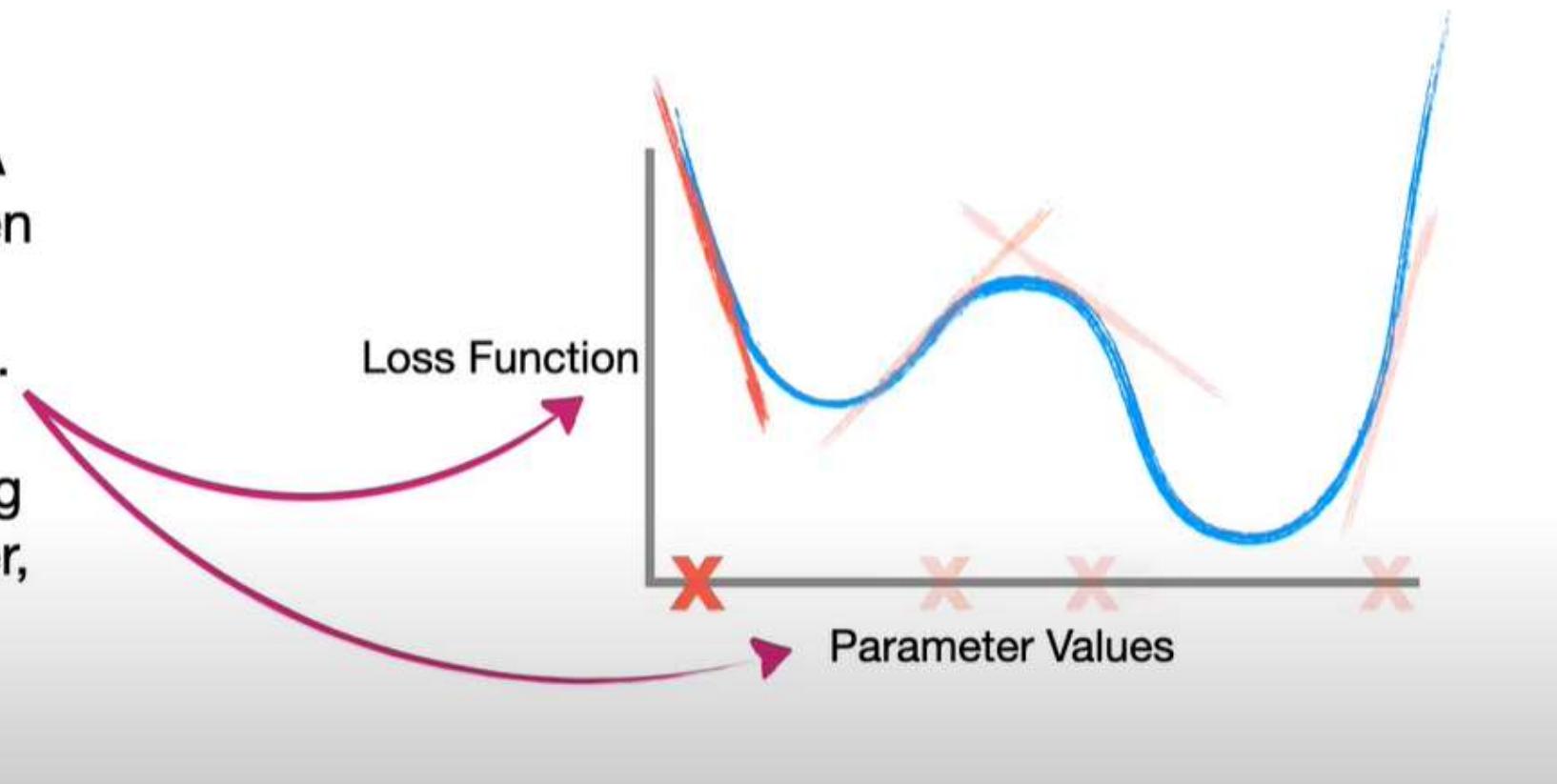
In other words, in order to find the parameter values that give us the lowest value for the **Loss Function**, we usually want to take relatively small steps.



Problema de las redes neuronales recurrentes

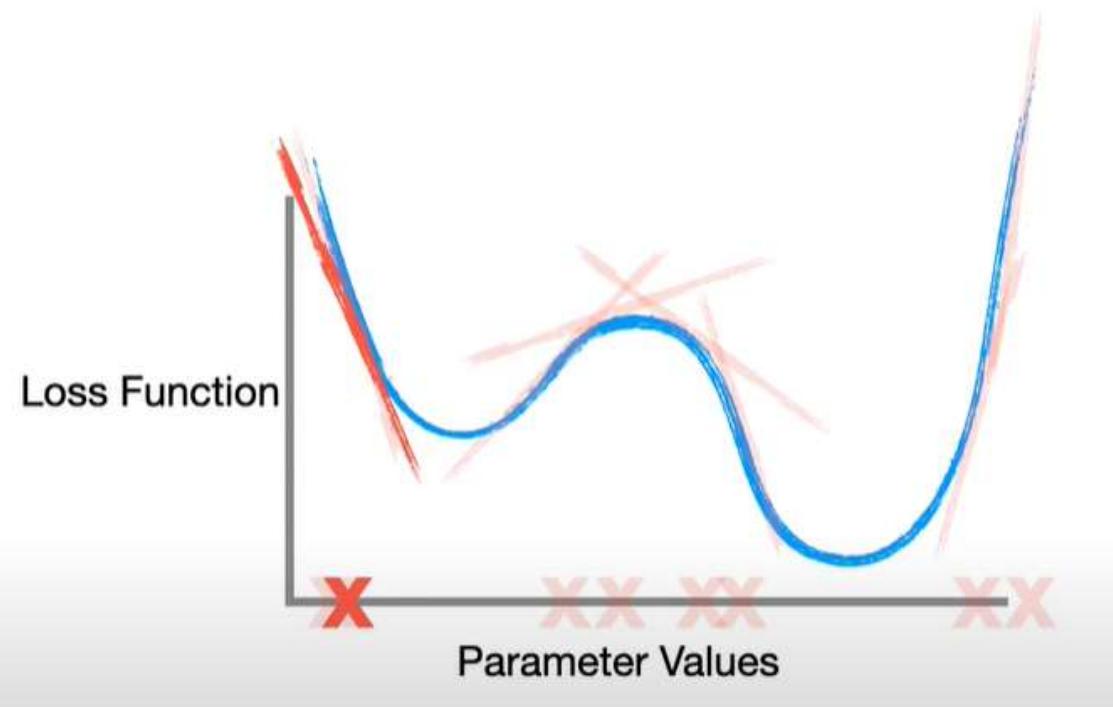
However, when the **Gradient** contains a **HUGE NUMBER**, then we'll end up taking relatively large steps.

And instead of finding the optimal parameter, we'll just bounce around a lot.



Problema de las redes neuronales recurrentes

One way to prevent
The Exploding Gradient Problem
would be to limit w_2 to values < 1.



Problema de las redes neuronales recurrentes

Now, just like before, we multiply the first input by w_2 raised to the number of times we **unroll** the network.

$\text{Input}_1 \times w_2^{\text{Num. Unroll}}$



Problema de las redes neuronales recurrentes



...and 0.5^{50} is a number
super close to 0.

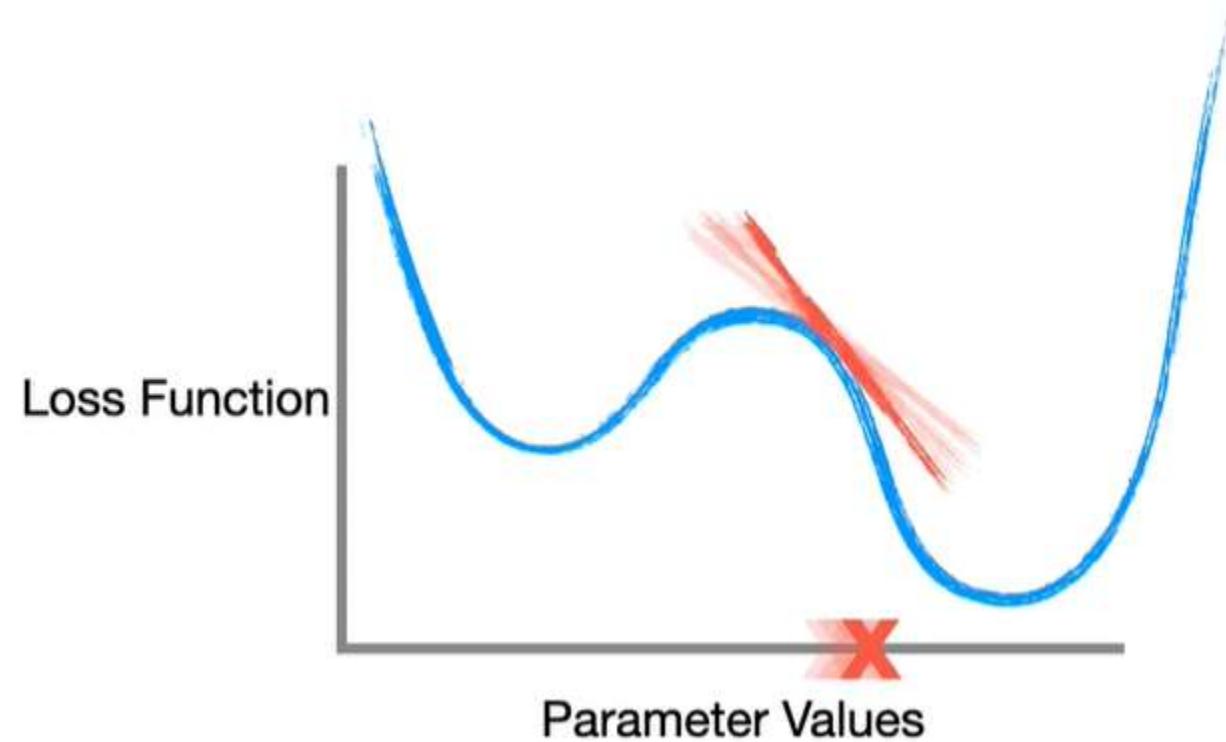
$$\text{Input}_1 \times 0.5^{50} = \text{Input}_1 \times \text{a number super close to 0}$$

$$\text{Input}_1 \times w_2^{\text{Num. Unroll}}$$

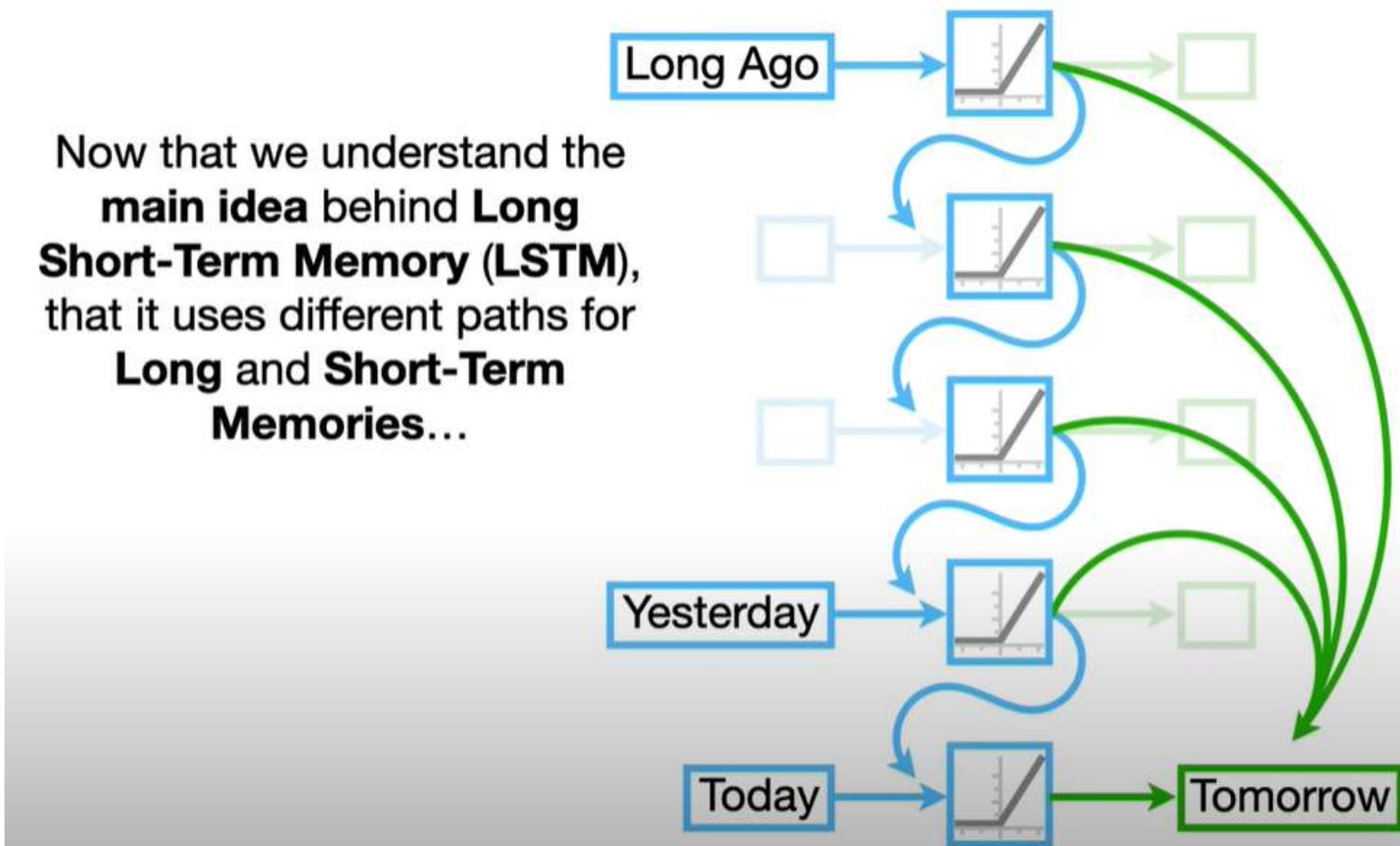


Problema de las redes neuronales recurrentes

And as a result, we end up hitting the maximum number of steps we are allowed to take before we find the optimal value.



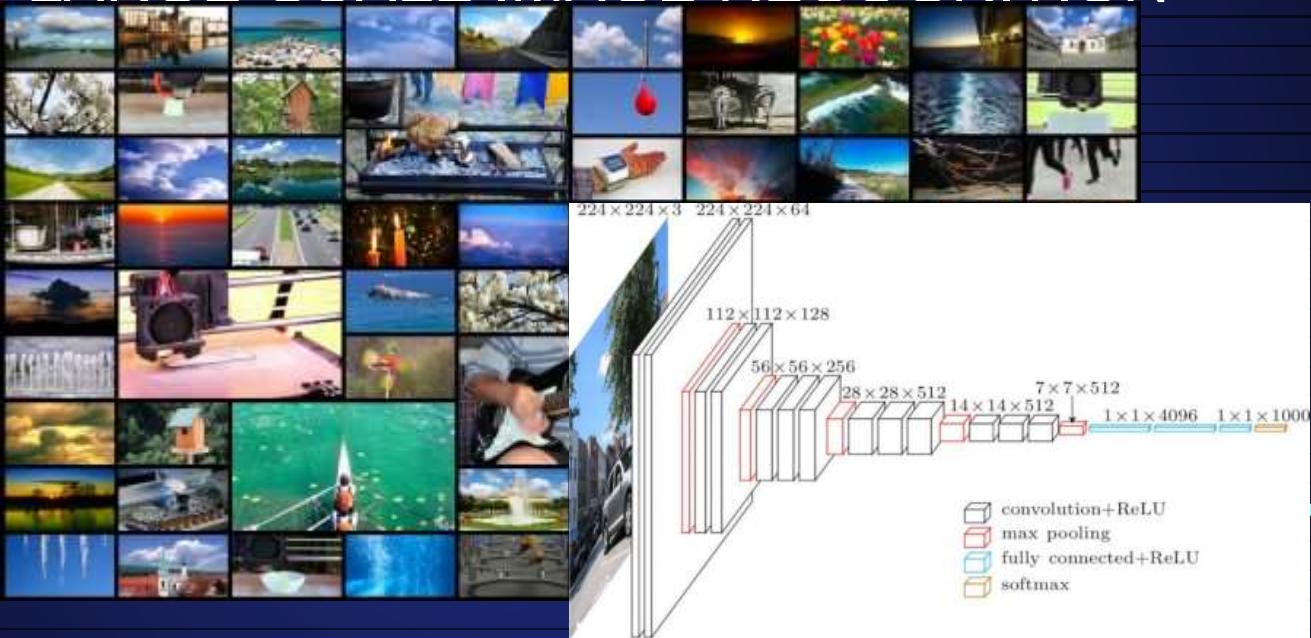
LSTM



VGG - 16

Chipana Perez, Gabriela Angel
Vizcarra Vargas, Piero Emiliano
Hilares Angelo, Maryori Lizeth

VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION



Karen Simonyan y Andrew Zisserman

Artículo de presentación

Very deep convolutional networks
for large-scale image recognition

Autores:

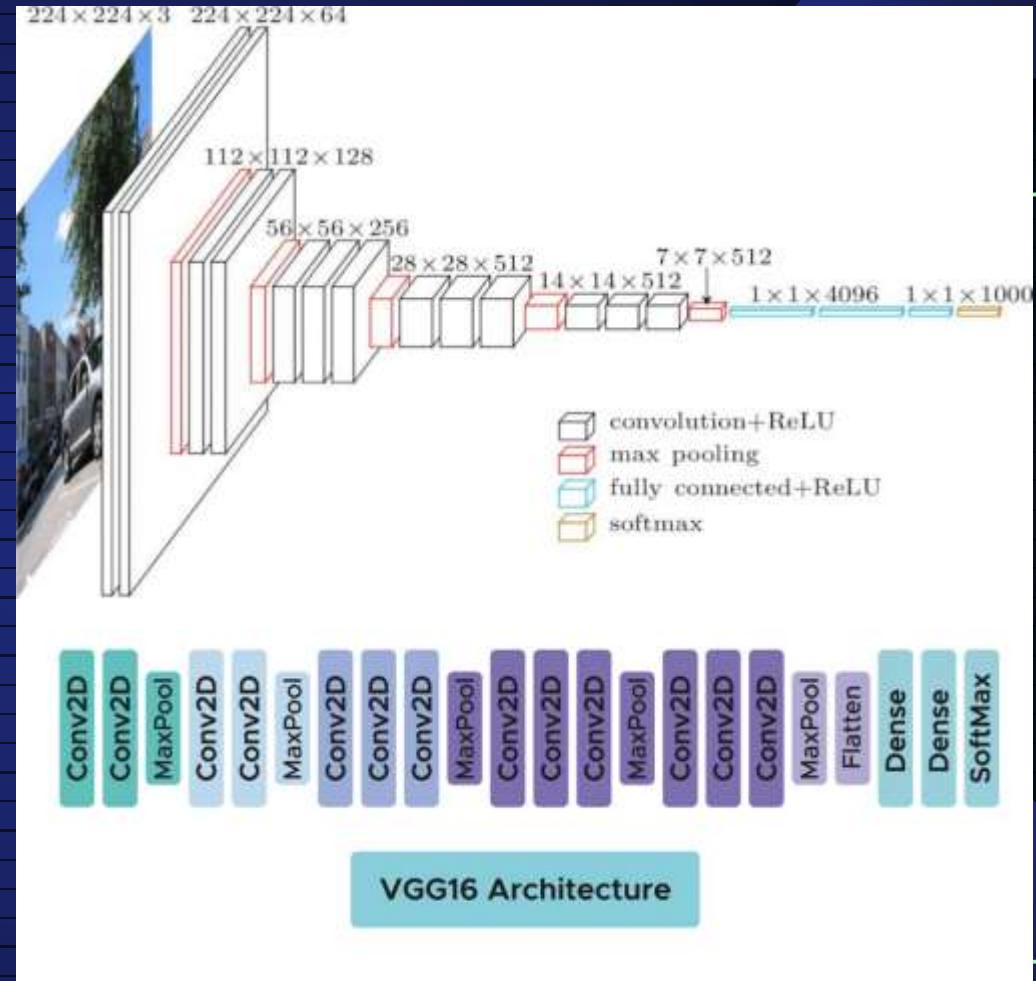
Karen Simonyan
Andrew Zisserman

- CNN alto rendimiento en el reconocimiento de imágenes a gran escala.
- Menciona la dificultad de entrenar redes más profundas y la necesidad de explorar arquitecturas que puedan manejar esta complejidad
- Objetivo: Presentar una arquitectura de CNN VGG-16 (y también VGG-19), que tiene una profundidad significativamente mayor que otras arquitecturas.
El objetivo es demostrar que una mayor profundidad de la red puede mejorar el rendimiento en la clasificación de imágenes en el conjunto de datos ImageNet.



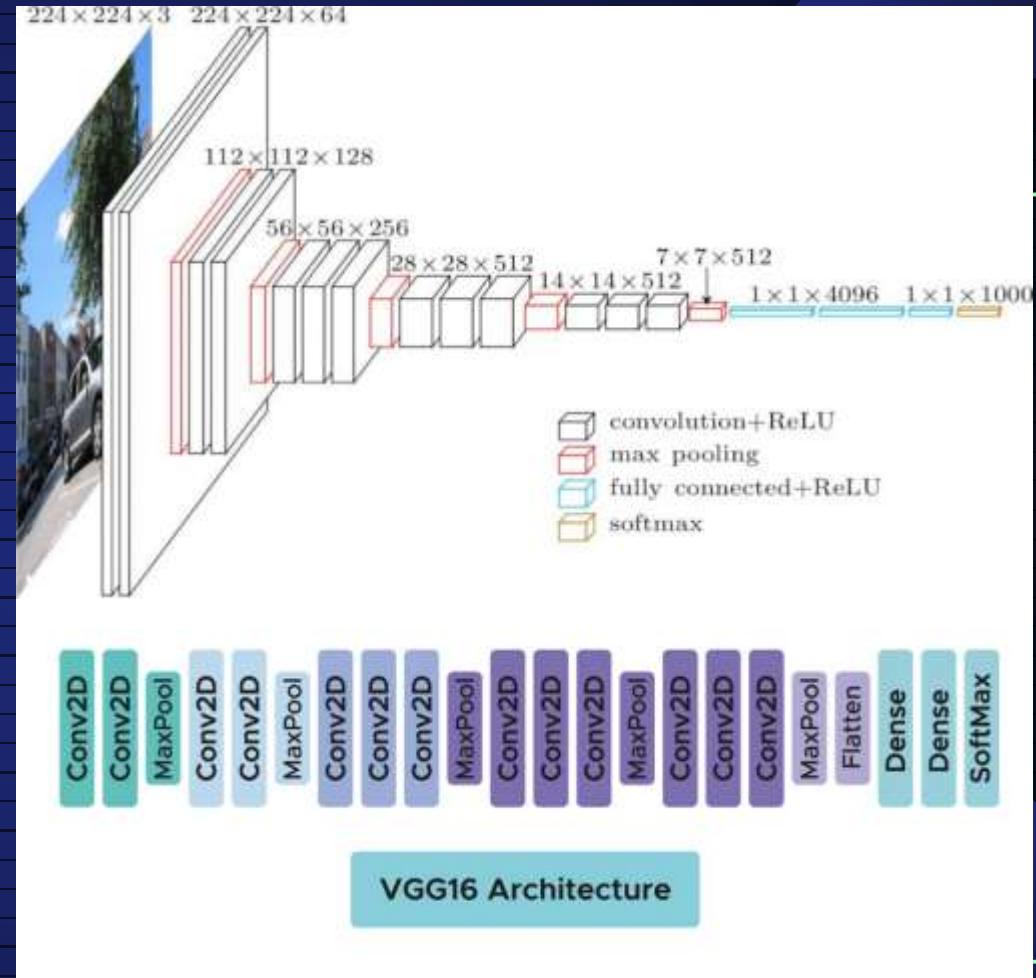
Arquitectura/estructura

- **Capas de entrada:** imágenes de tamaño 224×224
- **Capas convolucionales:** filtros de tamaño 3×3 con un paso de 1 píxel .
Aplican la función de activación ReLU después de cada convolución.
Objetivo: extraer características relevantes de las imágenes en diferentes niveles de abstracción.
- **Capas de agrupación (pooling):** Se aplica cada 2 capas convolucionales, (Max Pooling) con un tamaño de ventana de 2×2 y un paso de 2 píxeles.
Objetivo: Reducir la dimensionalidad y ayuda a obtener una representación más compacta de las características extraídas.



Arquitectura/estructura

- **Capas totalmente conectadas:** Cada capa con 4096 unidades.
Objetivo: transformar las características extraídas en una salida clasificada. Después de cada capa totalmente conectada, se aplica la función de activación ReLU, excepto en la última capa.
- **Capa de salida:** La capa de salida consiste en una capa totalmente conectada con 1000 unidades, correspondientes a las 1000 clases diferentes en el conjunto de datos ImageNet. La función de activación utilizada en esta capa es la función Softmax, que asigna probabilidades a cada clase.



Datos

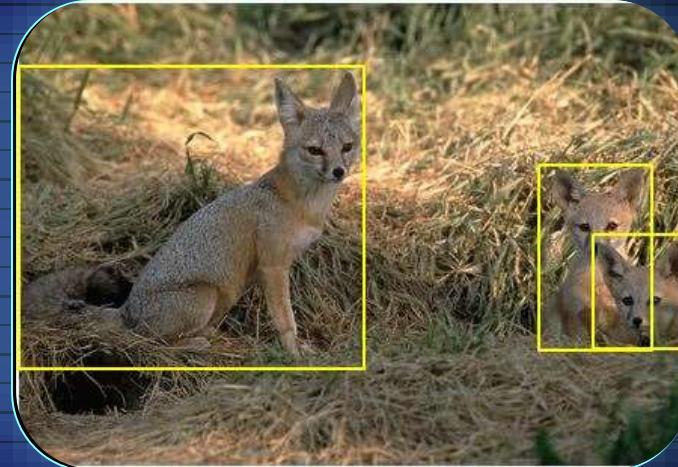
ILSVRC

Contiene los datos de imagen y la realidad del terreno para los conjuntos de entrenamiento y validación, y los datos de imagen para el conjunto de prueba.

Las anotaciones se ordenan por sus synsets (por ejemplo, "gato persa", "bicicleta de montaña" o "perro caliente") como su wnid.

Estas identificaciones se ven como n00141669.

El nombre de cada imagen tiene una correspondencia directa con el nombre del archivo de anotación.



ImageNet Large-Scale Visual
Recognition Challenge

Table 7: Comparison with the state of the art in ILSVRC classification. Our method is denoted as “VGG”. Only the results obtained without outside training data are reported.

Method	top-1 val. error (%)	top-5 val. error (%)	top-5 test error (%)
VGG (2 nets, multi-crop & dense eval.)	23.7	6.8	6.8
VGG (1 net, multi-crop & dense eval.)	24.4	7.1	7.0
VGG (ILSVRC submission, 7 nets, dense eval.)	24.7	7.5	7.3
GoogLeNet (Szegedy et al., 2014) (1 net)	-		7.9
GoogLeNet (Szegedy et al., 2014) (7 nets)	-		6.7
MSRA (He et al., 2014) (11 nets)	-	-	8.1
MSRA (He et al., 2014) (1 net)	27.9	9.1	9.1
Clarifai (Russakovsky et al., 2014) (multiple nets)	-	-	11.7
Clarifai (Russakovsky et al., 2014) (1 net)	-	-	12.5
Zeiler & Fergus (Zeiler & Fergus, 2013) (6 nets)	36.0	14.7	14.8
Zeiler & Fergus (Zeiler & Fergus, 2013) (1 net)	37.5	16.0	16.1
OverFeat (Sermanet et al., 2014) (7 nets)	34.0	13.2	13.6
OverFeat (Sermanet et al., 2014) (1 net)	35.7	14.2	-
Krizhevsky et al. (Krizhevsky et al., 2012) (5 nets)	38.1	16.4	16.4
Krizhevsky et al. (Krizhevsky et al., 2012) (1 net)	40.7	18.2	-

Áreas de aplicación

Clasificación de imágenes

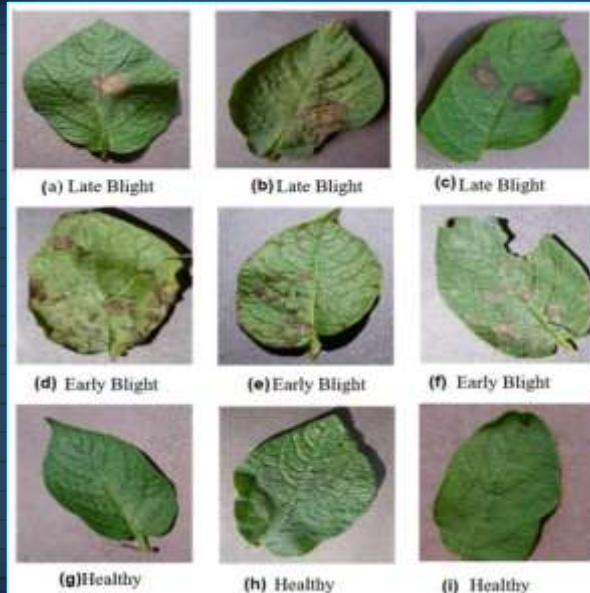
VGG-16 ha sido ampliamente utilizada para tareas de clasificación de imágenes en diferentes dominios, como reconocimiento de objetos, detección de enfermedades en imágenes médicas, clasificación de escenas, etc.



Predicción de edad ósea con red basada en VGG-16 y transfer learning

• • • • •

Áreas de aplicación



Detección de objetos

Junto con técnicas de detección como R-CNN, Faster R-CNN y YOLO, la VGG-16 se ha utilizado para la detección de objetos en imágenes, lo que implica identificar la presencia y ubicación de objetos específicos en una imagen.

Application of convolutional neural
networks for detection of the late blight
Phytophthora infestans in potato
Solanum tuberosum



Áreas de aplicación

Recuperación de información visual

La VGG-16 se ha utilizado en aplicaciones de recuperación de información visual, donde el objetivo es buscar imágenes similares en grandes bases de datos de imágenes. La extracción de características de la VGG-16 permite representar las imágenes de manera compacta y compararlas eficientemente.

Segmentación semántica

La VGG-16 se ha utilizado en aplicaciones de segmentación semántica, donde el objetivo es asignar una etiqueta semántica a cada píxel de una imagen. Esta tarea es útil en aplicaciones como la conducción autónoma, el etiquetado automático de imágenes y la realidad aumentada.



Demo

Conclusiones

1. VGG-16 es una arquitectura de red convolucional popular y ampliamente utilizada en el campo de la visión por computadora. Fue desarrollada por el grupo de investigación Visual Geometry Group en la Universidad de Oxford.
2. La arquitectura VGG-16 se caracteriza por su simplicidad y eficacia. Está compuesta por 16 capas, incluyendo 13 capas convolucionales y 3 capas completamente conectadas.
3. VGG-16 ha demostrado un rendimiento impresionante en tareas de clasificación de imágenes, logrando resultados cercanos al estado del arte en conjuntos de datos desafiantes como ImageNet.
4. La estructura de VGG-16, con capas convolucionales en cascada, permite aprender características visuales jerárquicas de una imagen, desde características de bajo nivel (como bordes y texturas) hasta características de alto nivel (como formas y objetos).



Conclusiones

5. La capacidad de transferencia de aprendizaje de VGG-16 es una de sus características más destacadas. Al pre-entrenar la red en grandes conjuntos de datos, se pueden transferir los conocimientos aprendidos a tareas específicas con conjuntos de datos más pequeños, ahorrando tiempo y recursos de entrenamiento.
6. VGG-16 se ha utilizado en una variedad de aplicaciones, como clasificación de objetos, detección de objetos, segmentación semántica, recuperación de información visual, entre otras. Su capacidad para aprender características visuales complejas la hace valiosa en diversos problemas de visión por computadora.
7. Aunque VGG-16 ha sido una arquitectura influyente, también tiene algunas limitaciones. Es una red profunda y relativamente pesada, lo que puede dificultar su implementación en dispositivos con recursos limitados. Además, puede requerir conjuntos de datos grandes para un entrenamiento efectivo.



Conclusiones

En general, la arquitectura VGG-16 ha dejado una huella significativa en el campo de la visión por computadora. Su simplicidad, eficacia y capacidad de transferencia de aprendizaje la han convertido en una opción popular para muchas aplicaciones de procesamiento de imágenes y ha sentado las bases para el desarrollo de arquitecturas más avanzadas.

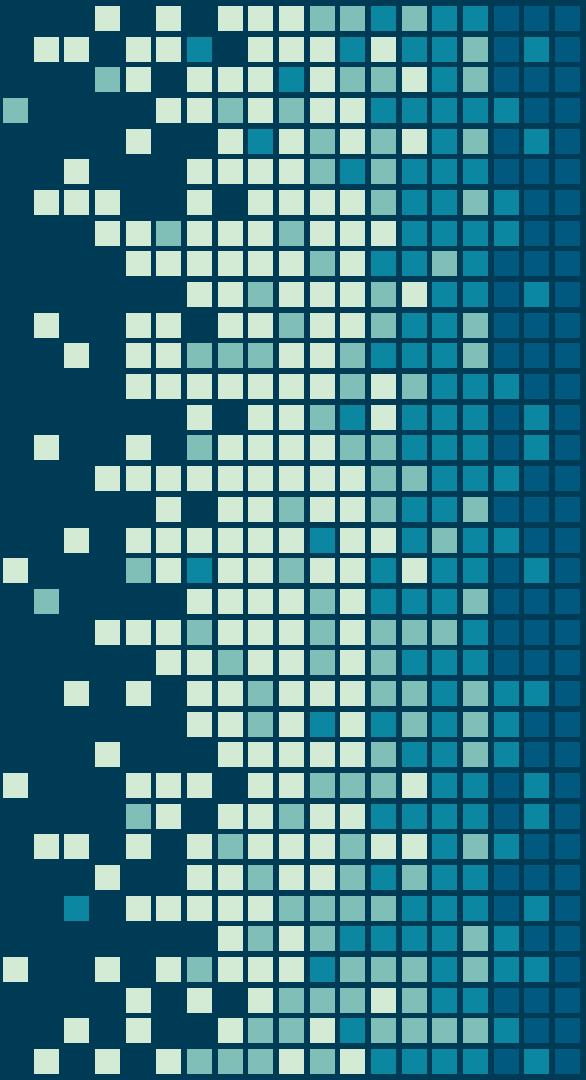


Redes Neuronales Convolucionales:LeNet-5

Integrantes:

- **Quispe Huanca, Joselyn Lizeth**
- **Macedo Huaman Vanessa Mayra**

1. INTRODUCCIÓN



LeNet se refiere a LeNet-5 y es una red neuronal convolucional simple.

LeNet-5 fue una de las primeras redes neuronales convolucionales y promovió el desarrollo del deep learning.

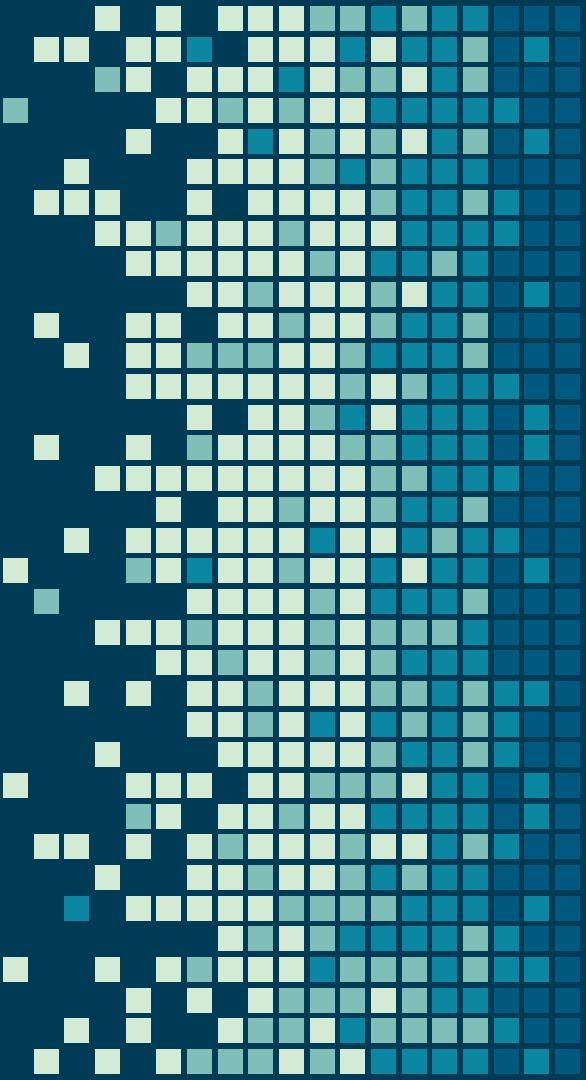
Fue propuesto en 1989 por Yann LeCun



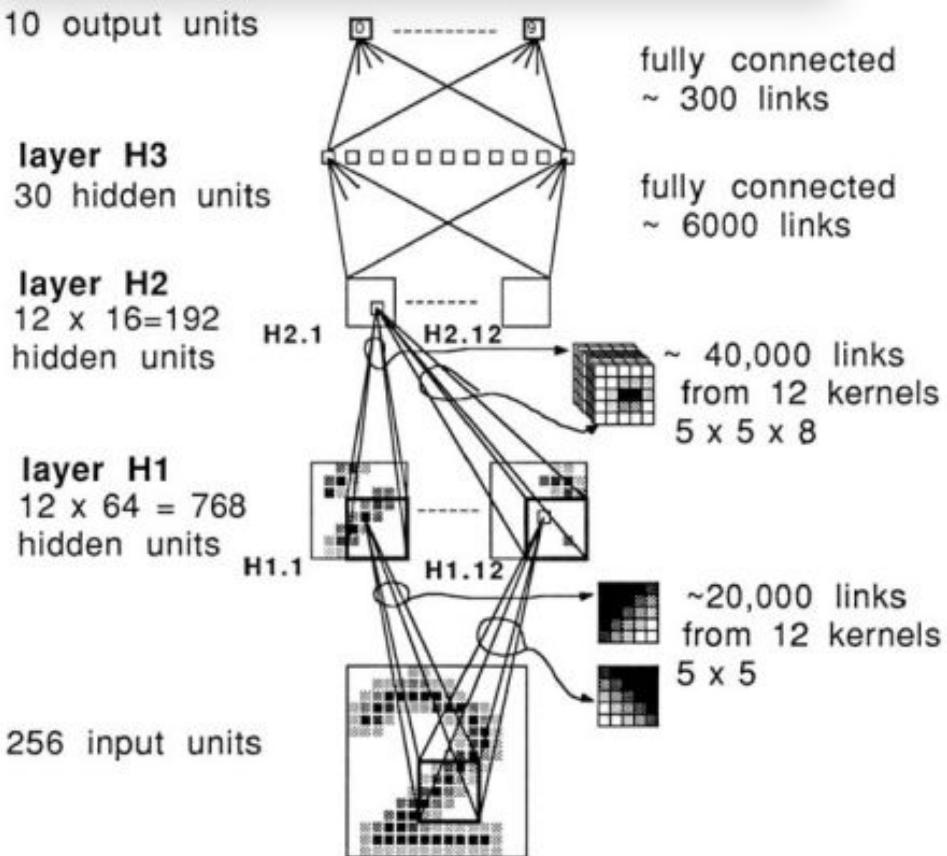
Yann LeCun

2. PAPER

Backpropagation Applied to
Handwritten Zip Code Recognition

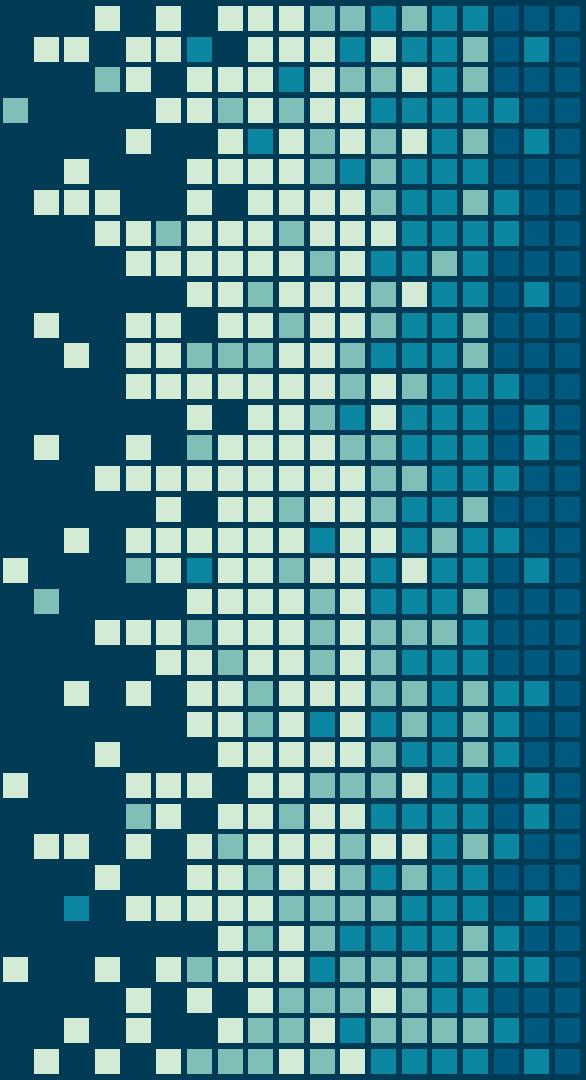


1611915485726803224414186
 6359720299299722510044701
 3084111591010615406103631
 1064111030475262001179966
 8912056708557131427955460
 2018730187112993089970984
 0109707597331972015519055
 1075318255182814358090943
 1787541655460354603546055
 18255108503047520439401

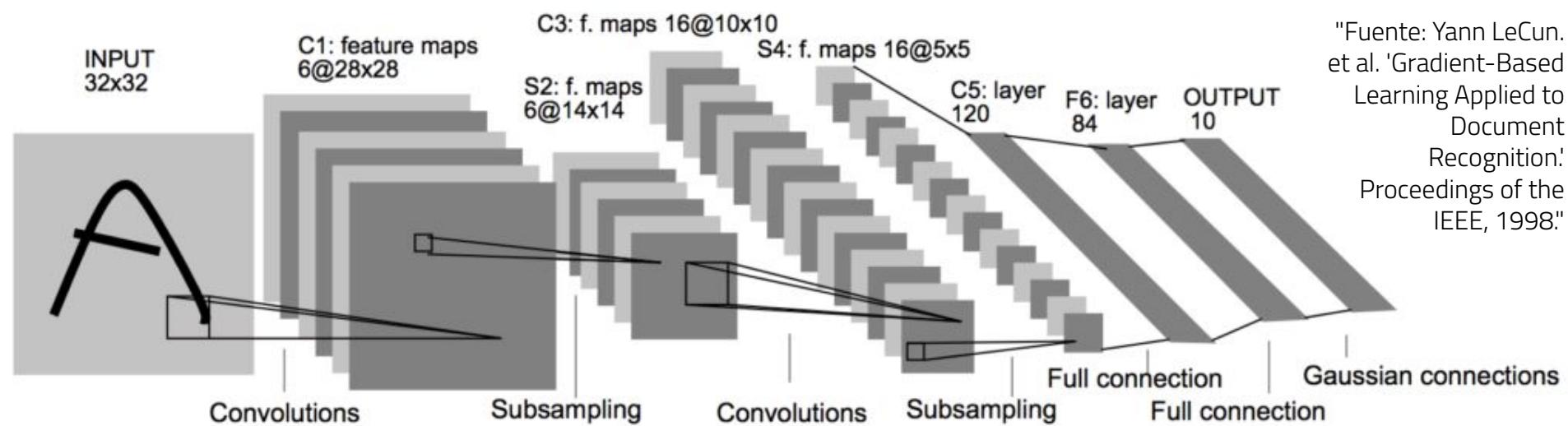


3. ARQUITECTURA

LeNet-5

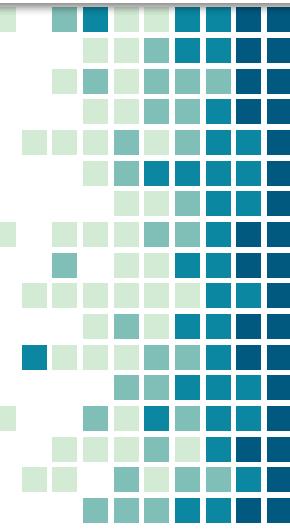


"Fuente: Yann LeCun,
et al. 'Gradient-Based
Learning Applied to
Document
Recognition:
Proceedings of the
IEEE, 1998."

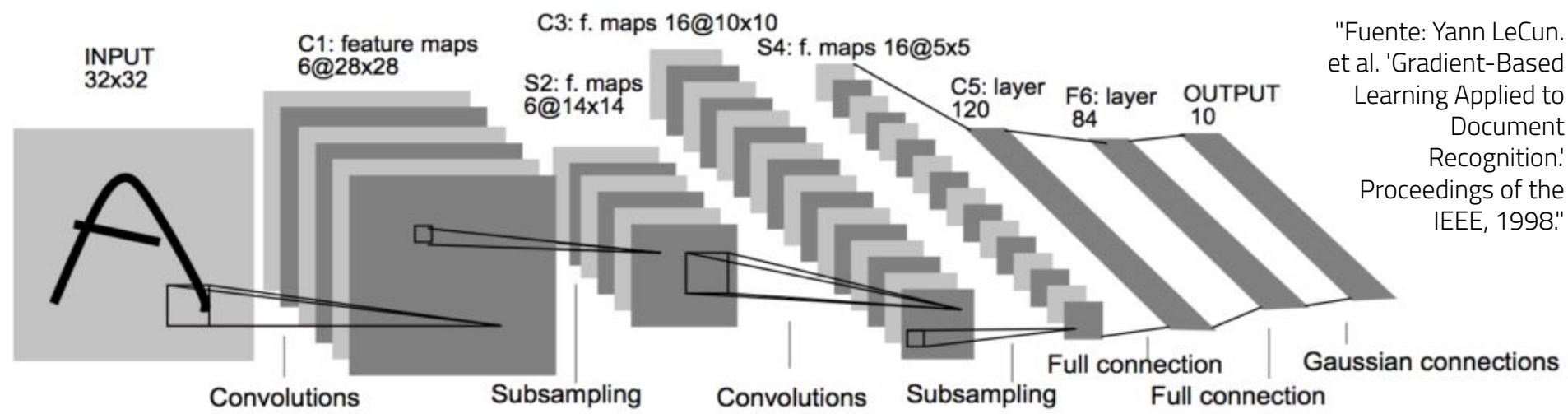


Contiene:

- Capa de entrada
- Capa convolucional (C1)
- Capa de pooling (S2)
- Capa convolucional (C3)
- Capa de pooling (S4)
- Capa completamente conectada (C5)
- Capa completamente conectada (F6)
- Capa de salida



"Fuente: Yann LeCun, et al. 'Gradient-Based Learning Applied to Document Recognition: Proceedings of the IEEE, 1998."



Capa de entrada

- Imágenes en escala de grises de 32x32 píxeles.

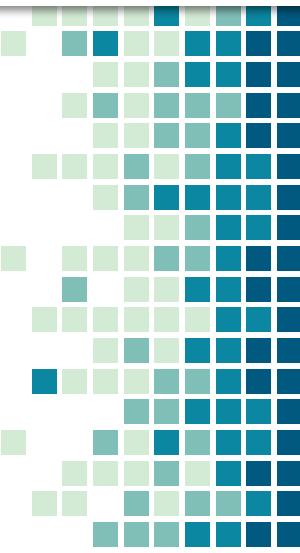
Capa convolucional (C1)

- 6 filtros de tamaño 5x5
- Tangente hiperbólica (tanh)

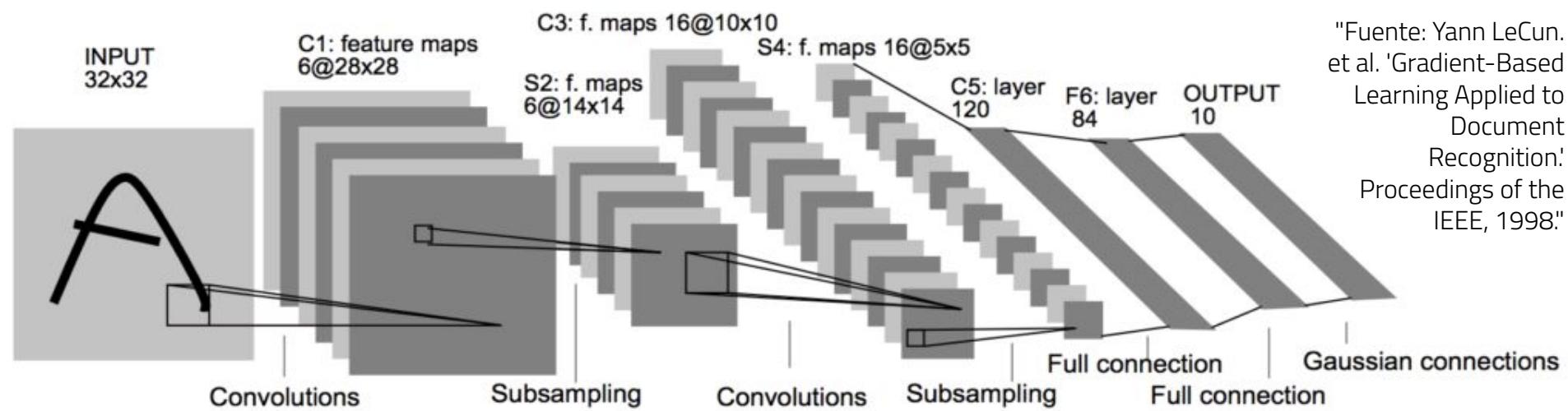
- Stride: 1
- 6 mapas de características de 28x28

Capa de pooling (S2)

- Average pooling.
- Tamaño de ventana 2x2
- Stride: 2
- 6 mapas de características de 14x14



"Fuente: Yann LeCun,
et al. 'Gradient-Based
Learning Applied to
Document
Recognition:
Proceedings of the
IEEE, 1998."



Capa convolucional (C3)

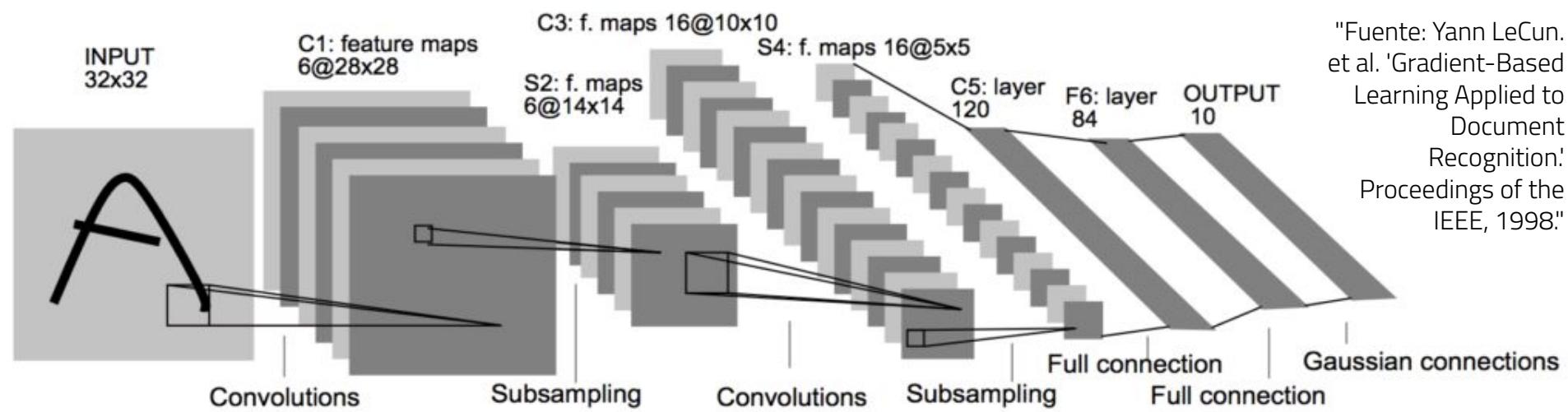
- 16 filtros de tamaño 5x5
- Tangente hiperbólica (tanh)
- Stride: 1
- 16 mapas de características de 10x10

Capa de pooling (S4)

- Average pooling.
- Tamaño de ventana 2x2
- Stride: 2
- 16 mapas de características de 5x5.



"Fuente: Yann LeCun,
et al. 'Gradient-Based
Learning Applied to
Document
Recognition:
Proceedings of the
IEEE, 1998."



Capa completamente conectada (C5)

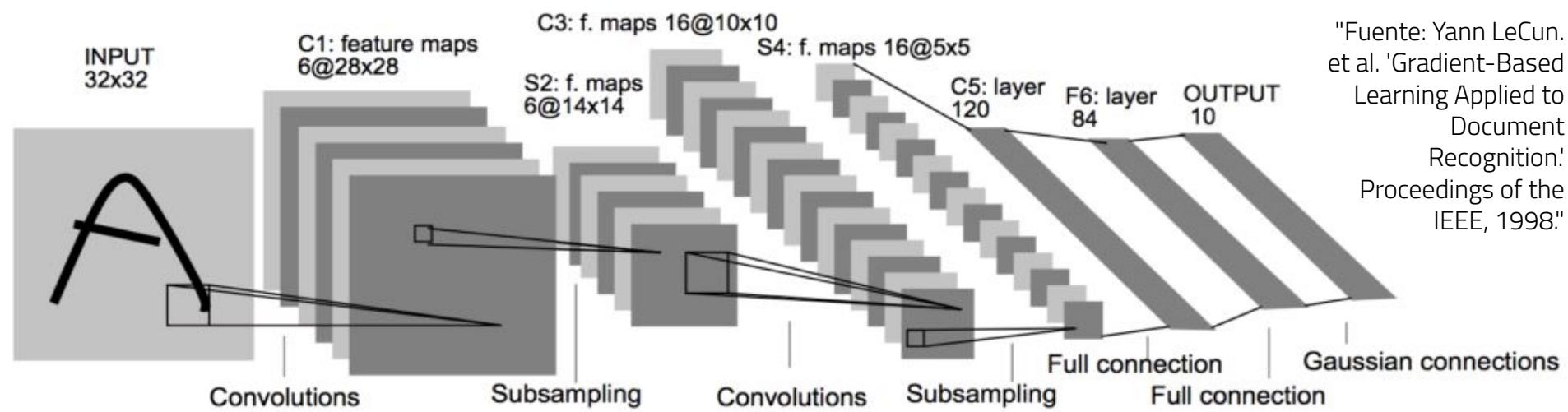
- 120 neuronas totalmente conectadas.
- Tangente hiperbólica (tanh).
- Un vector de 120 dimensiones.

Capa completamente conectada (F6)

- 84 neuronas totalmente conectadas.
- Tangente hiperbólica (tanh).
- Un vector de 84 dimensiones.

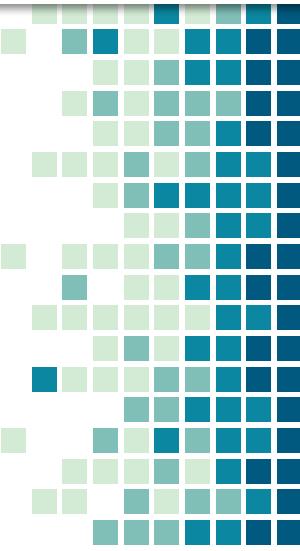


"Fuente: Yann LeCun,
et al. 'Gradient-Based
Learning Applied to
Document
Recognition:
Proceedings of the
IEEE, 1998."

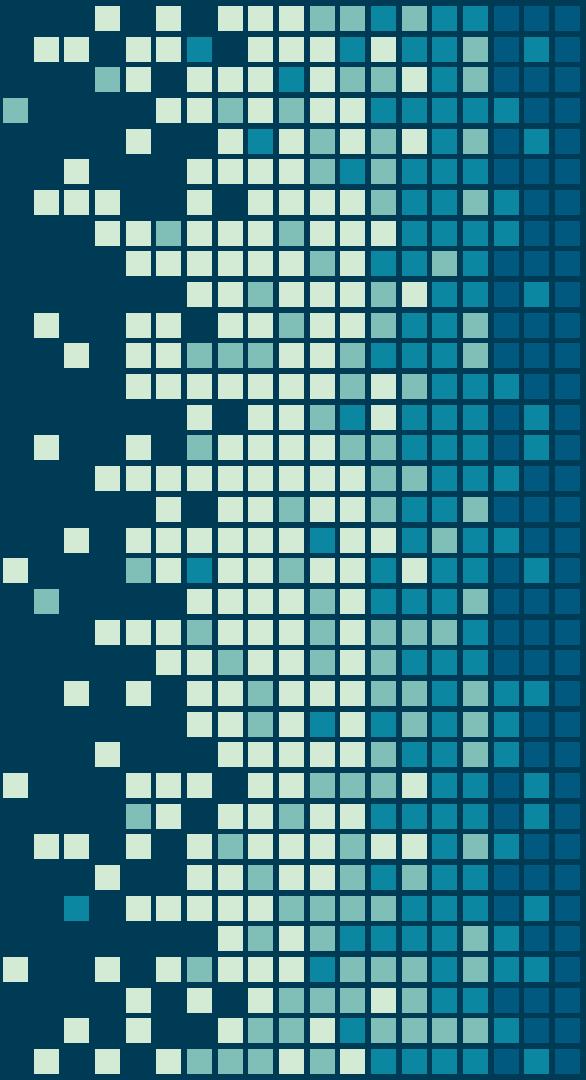


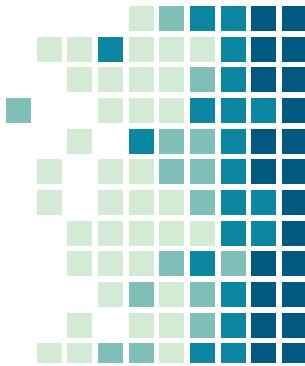
Capa de salida

- 10 neuronas (una para cada clase en el conjunto de datos utilizado en el paper).
- Función de activación: Softmax.
- Un vector de probabilidades de 10 dimensiones, representando la probabilidad de cada clase.
- **Gaussian Connections:** Se refiere a la capa de normalización o suavizado, o a la función de activación gaussiana en la capa de salida softmax



4. DATASET MNIST



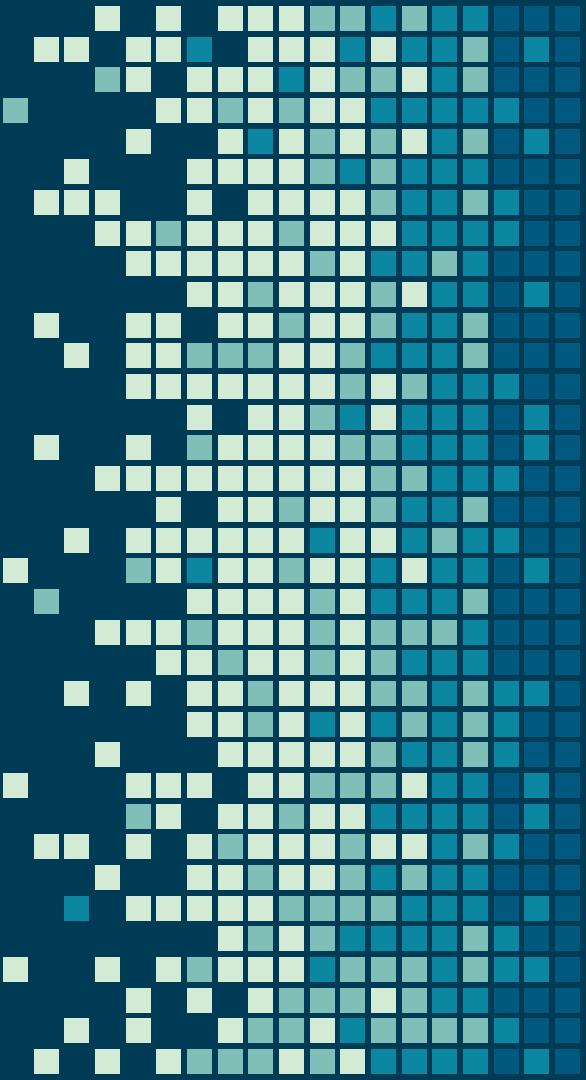


MNIST (Modified National Institute of Standards and Technology database)

- Consiste en imágenes de dígitos escritos a mano, en escala de grises de 28x28 píxeles, donde cada imagen representa un dígito del 0 al 9.
- Conjunto de entrenamiento: 60,000.
- Conjunto de prueba: 10,000.

3	6	8	1	7	9	6	6	9	1
6	7	5	7	8	6	3	4	8	5
2	1	7	9	7	1	2	8	4	6
4	8	1	9	0	1	8	8	9	4
7	6	1	8	6	4	1	5	6	0
7	5	9	2	6	5	8	1	9	7
2	2	2	2	3	4	4	8	0	
0	2	3	8	0	7	3	8	5	7
0	1	4	6	4	6	0	2	4	3
7	1	2	8	7	6	9	8	6	1

5. APLICACIÓN



APLICACIONES

Reconocimiento de dígitos y caracteres

Reconocimiento de matrículas de vehículos, reconocimiento de códigos de barras, etc.

Reconocimiento de objetos

Se utiliza para tareas de clasificación de objetos, detección de objetos y segmentación

Reconocimiento facial

Detectar y reconocer caras en imágenes o videos

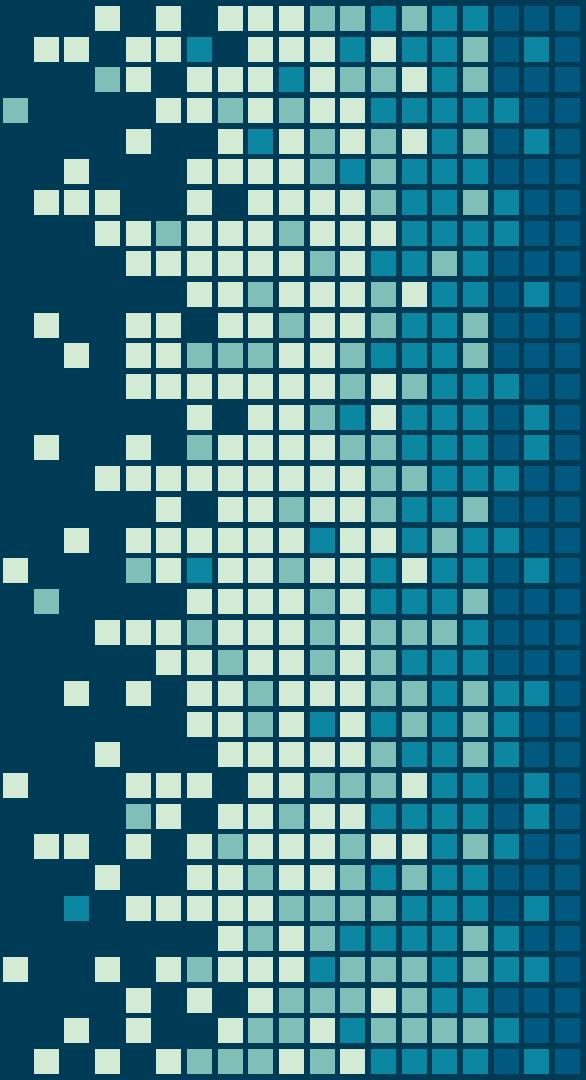
Visión por computadora en general

Detección de anomalías, reconocimiento de gestos, seguimiento de objetos, reconstrucción 3D a partir de imágenes, entre otros.

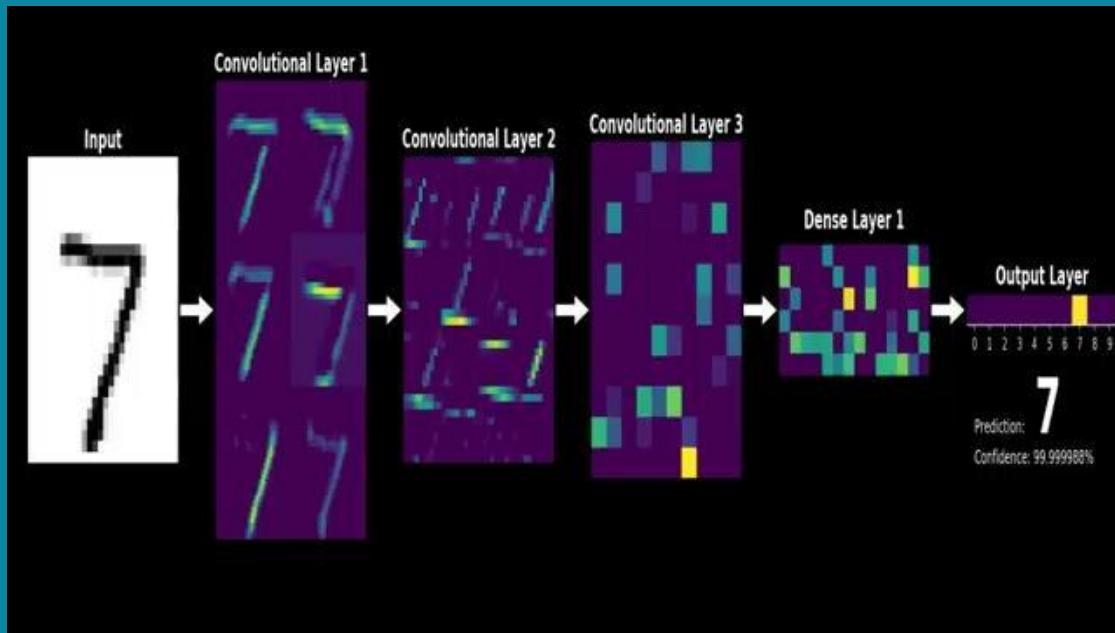
Análisis de imágenes médicas

Detectar cáncer de mama, la retinopatía diabética, la detección de tumores cerebrales, entre otros

6. Demo



“ DEMO





CONCLUSIONES

- LeNet fue una de las primeras arquitecturas exitosas de redes neuronales convolucionales, diseñada específicamente para el reconocimiento de caracteres escritos a mano.
- La estructura jerárquica de LeNet, permitió extraer características relevantes de las imágenes de entrada y lograr un alto rendimiento en la clasificación de dígitos..

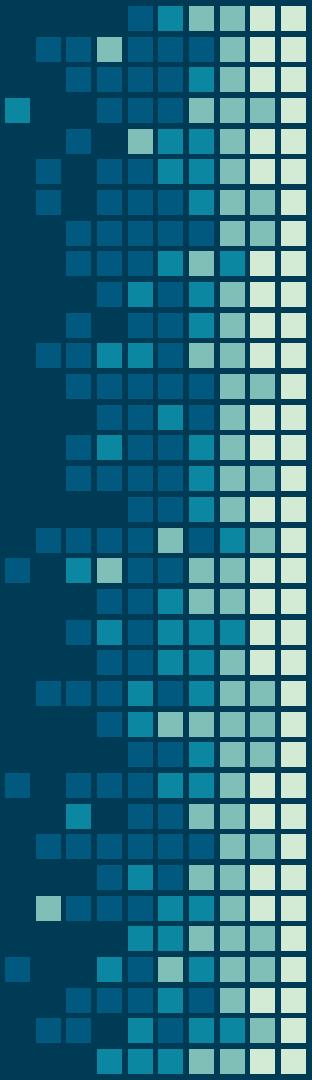


CONCLUSIONES

- El entrenamiento y evaluación de LeNet se realizó utilizando el dataset MNIST, demostrando su eficacia en la clasificación de imágenes de dígitos escritos a mano.
- LeNet sentó las bases para el desarrollo y avance de otras arquitecturas más sofisticadas de redes neuronales convolucionales, y ha sido ampliamente utilizada en aplicaciones de reconocimiento de patrones y visión por computadora.

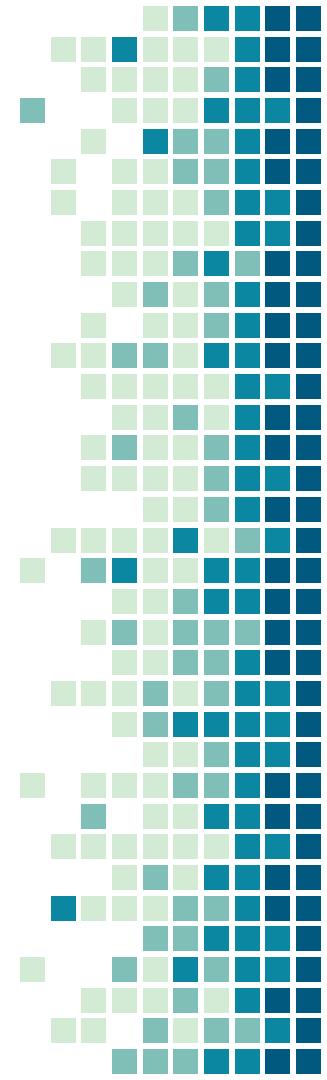
THANKS!

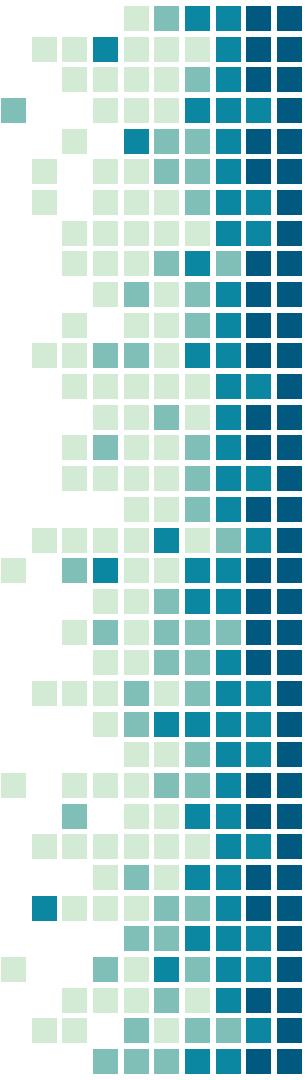
Any questions?



BIBLIOGRAFÍA

"Fuente: Yann LeCun. et al. 'Gradient-Based Learning Applied to Document Recognition.' Proceedings of the IEEE, 1998."





THIS IS A SLIDE TITLE

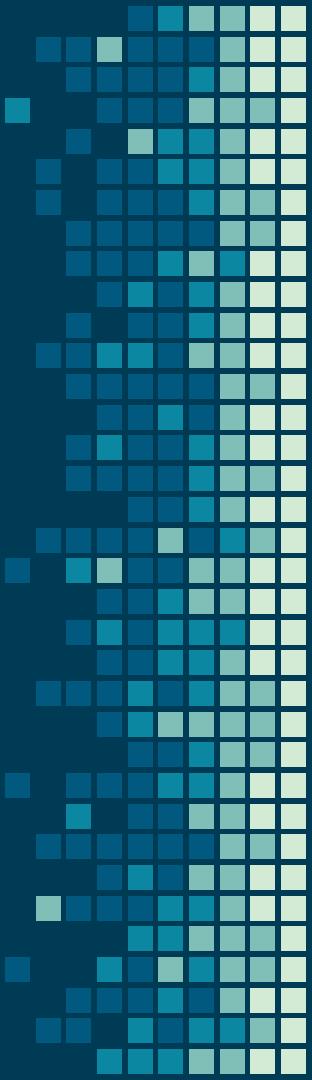
- Here you have a list of items
- And some text
- But remember not to overload your slides with content

Your audience will listen to you or read the content,
but won't do both.



BIG PT

Bring the attention of your audience over
a key concept using icons or illustrations



IN TWO OR THREE COLUMNS

Yellow

Is the color of gold, butter and ripe lemons. In the spectrum of visible light, yellow is found between green and orange.

Blue

Is the colour of the clear sky and the deep sea. It is located between violet and green on the optical spectrum.

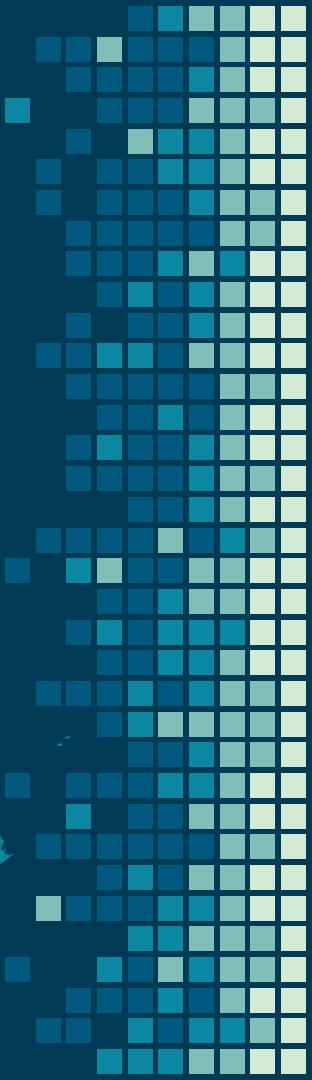
Red

Is the color of blood, and because of this it has historically been associated with sacrifice, danger and courage.

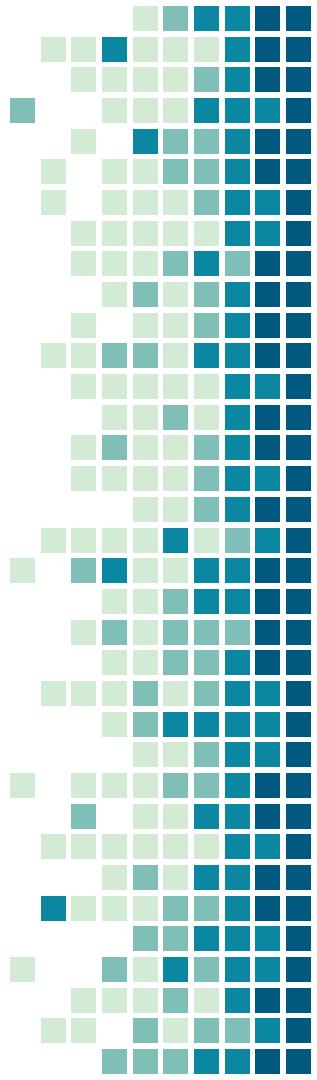
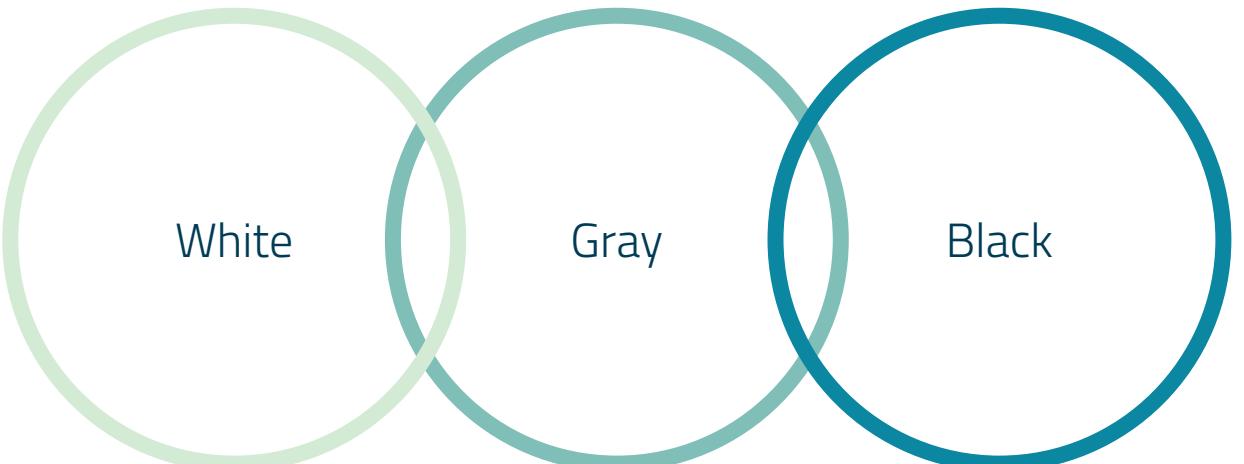
A photograph of a woman with long dark hair, seen from behind, sitting at a wooden desk. She is holding a silver spoon and stirring a dark liquid in a grey mug. A small Tazo tea bag hangs from the bottom of the mug. Her right hand rests on the trackpad of an open white laptop. The laptop screen displays a website with a grid of images and the word "Design".

Want big impact?
Use big image.

MAPS

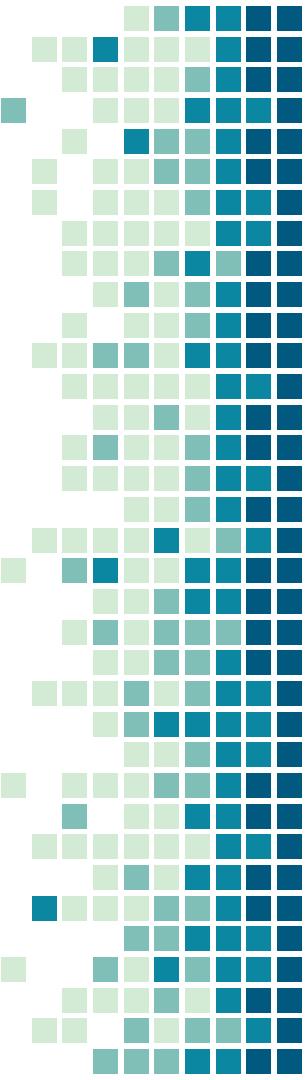


USE CHARTS TO EXPLAIN YOUR IDEAS



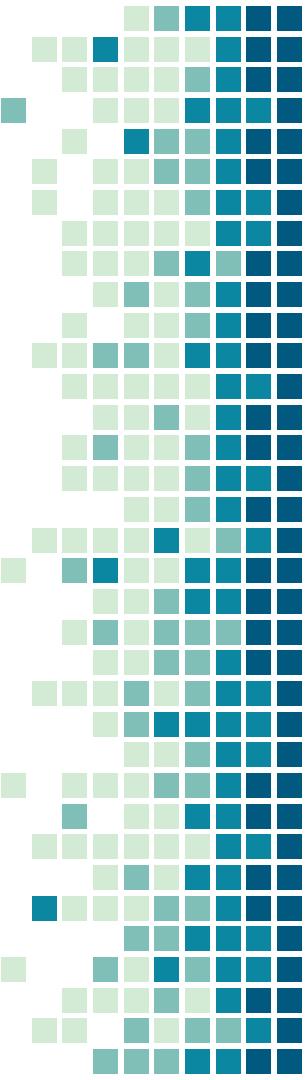
AND TABLES TO COMPARE DATA

	A	B	C
Yellow	10	20	7
Blue	30	15	10
Orange	5	24	16



89,526,124

Whoa! That's a big number, aren't you proud?



89,526,124\$

That's a lot of money

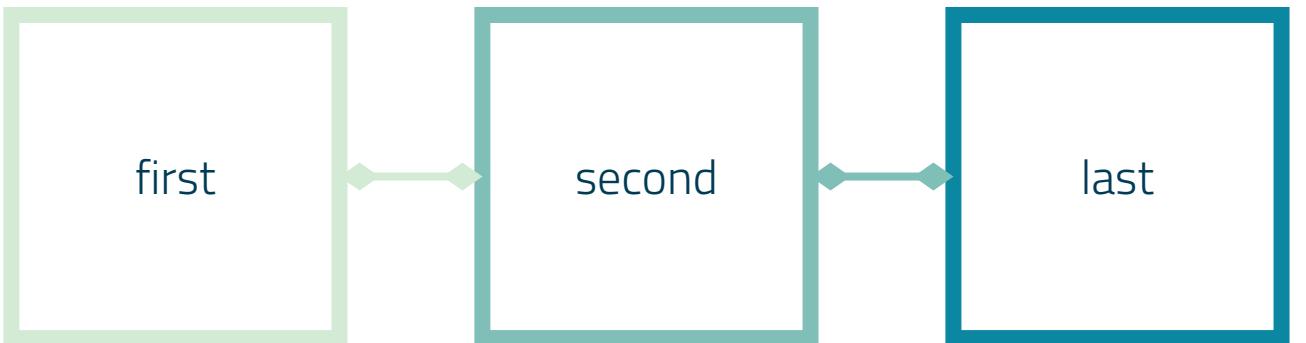
185,244 users

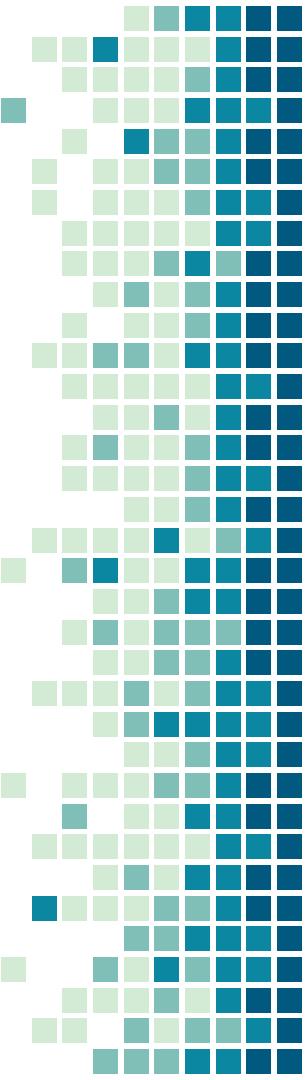
And a lot of users

100%

Total success!

OUR PROCESS IS EASY





LET'S REVIEW SOME CONCEPTS

Yellow

Is the color of gold, butter and ripe lemons. In the spectrum of visible light, yellow is found between green and orange.

Yellow

Is the color of gold, butter and ripe lemons. In the spectrum of visible light, yellow is found between green and orange.

Blue

Is the colour of the clear sky and the deep sea. It is located between violet and green on the optical spectrum.

Blue

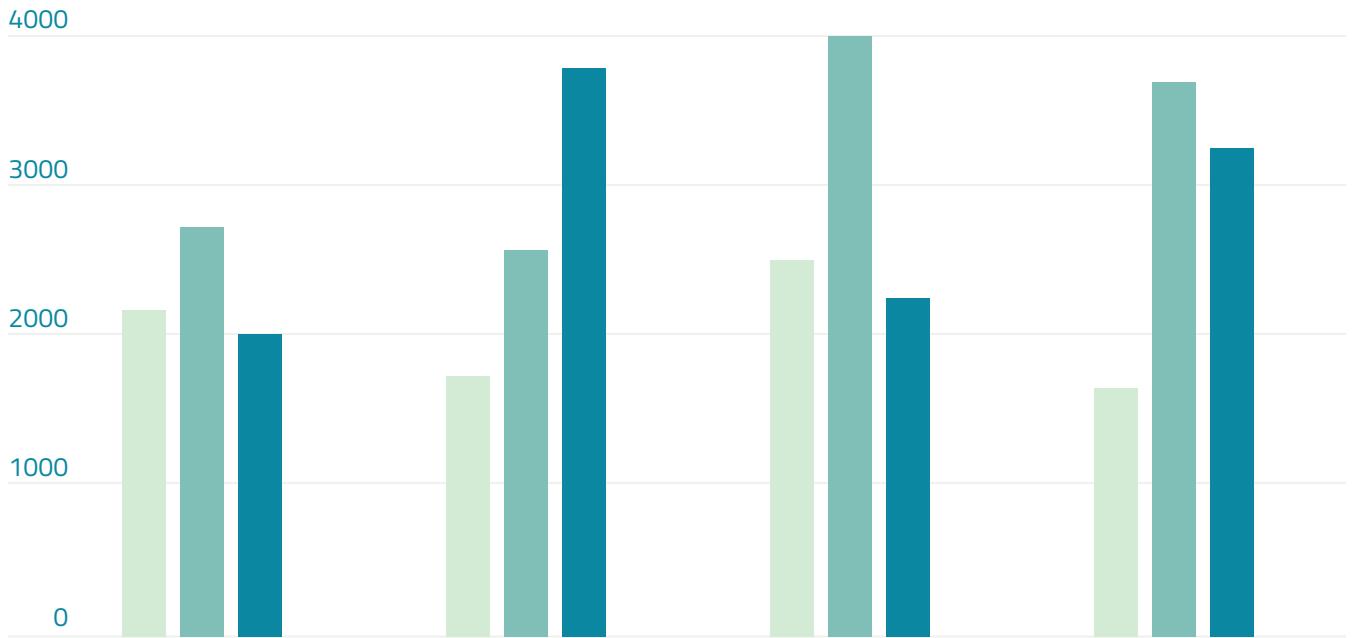
Is the colour of the clear sky and the deep sea. It is located between violet and green on the optical spectrum.

Red

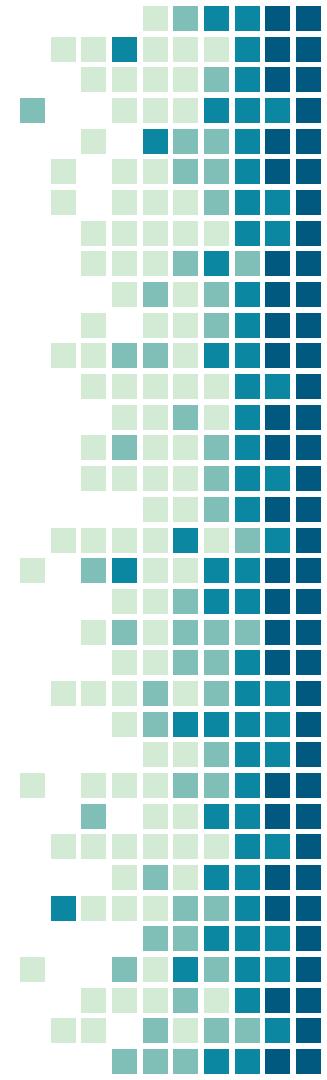
Is the color of blood, and because of this it has historically been associated with sacrifice, danger and courage.

Red

Is the color of blood, and because of this it has historically been associated with sacrifice, danger and courage.

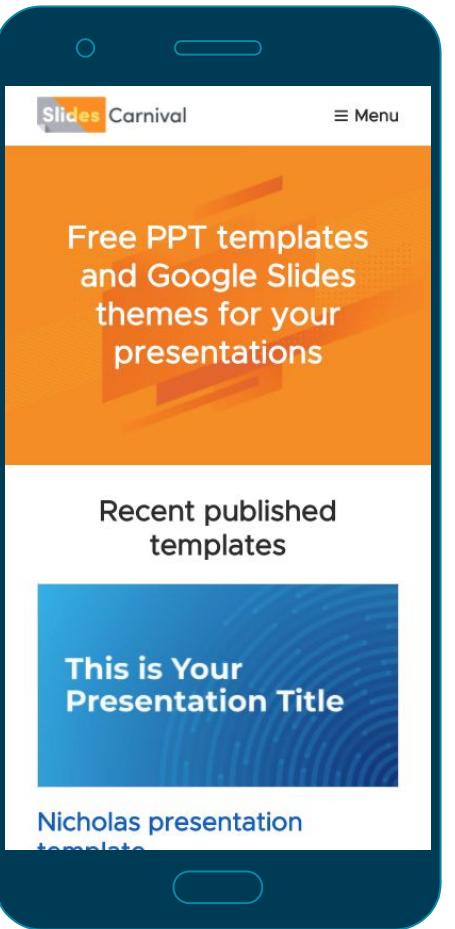


You can insert graphs from Excel or Google Sheets



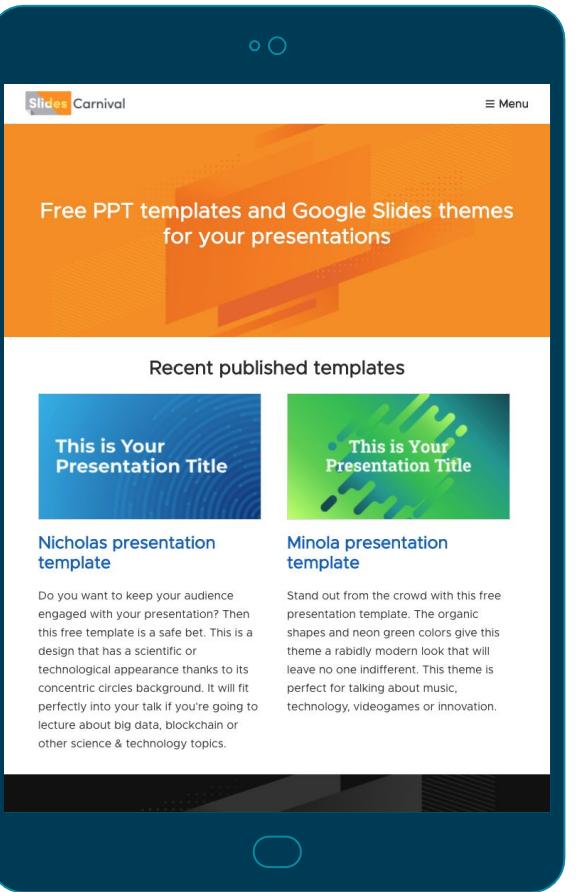
MOBILE PROJECT

Show and explain your web, app or software projects using these gadget templates.



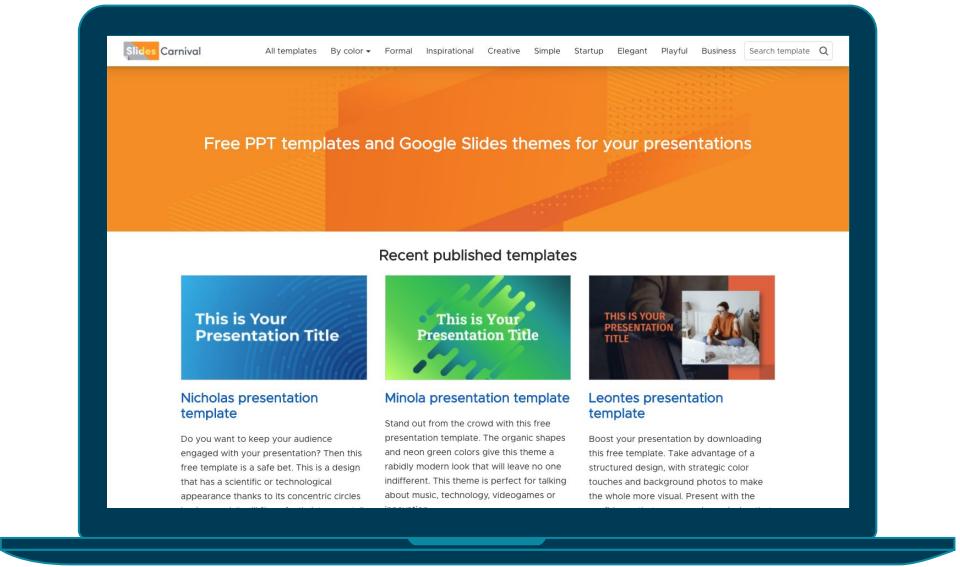
TABLET PROJECT

Show and explain your web, app or software projects using these gadget templates.



DESKTOP PROJECT

Show and explain your web, app or software projects using these gadget templates.



CREDITS

Special thanks to all the people who made and released these awesome resources for free:

- Presentation template by [SlidesCarnival](#)
- Photographs by [Unsplash](#)

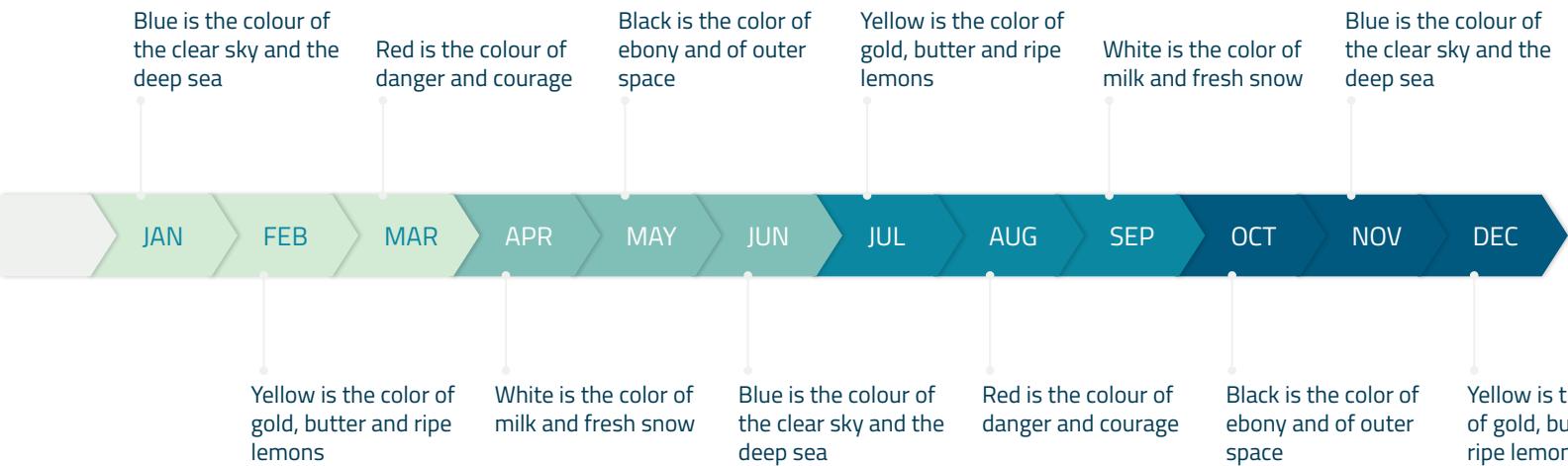
2.

EXTRA RESOURCES

For Business Plans, Marketing Plans,
Project Proposals, Lessons, etc



TIMELINE



ROADMAP

Blue is the colour of the clear sky and the deep sea

1

Red is the colour of danger and courage

3

Black is the color of ebony and of outer space

5

Yellow is the color of gold, butter and ripe lemons

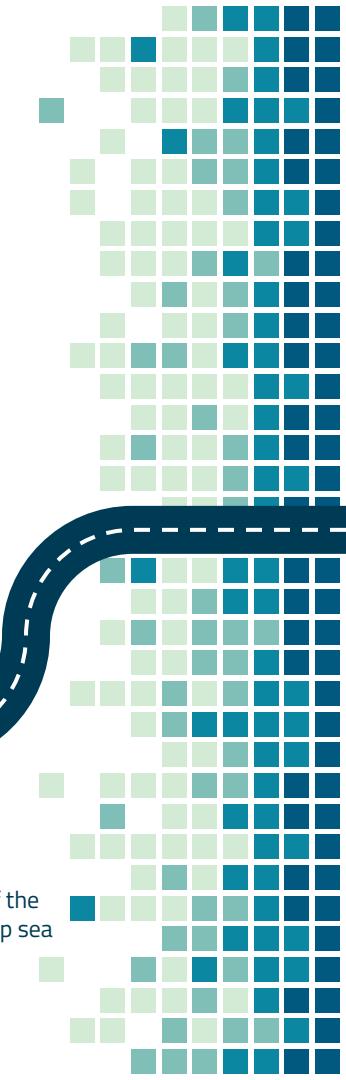
2

White is the color of milk and fresh snow

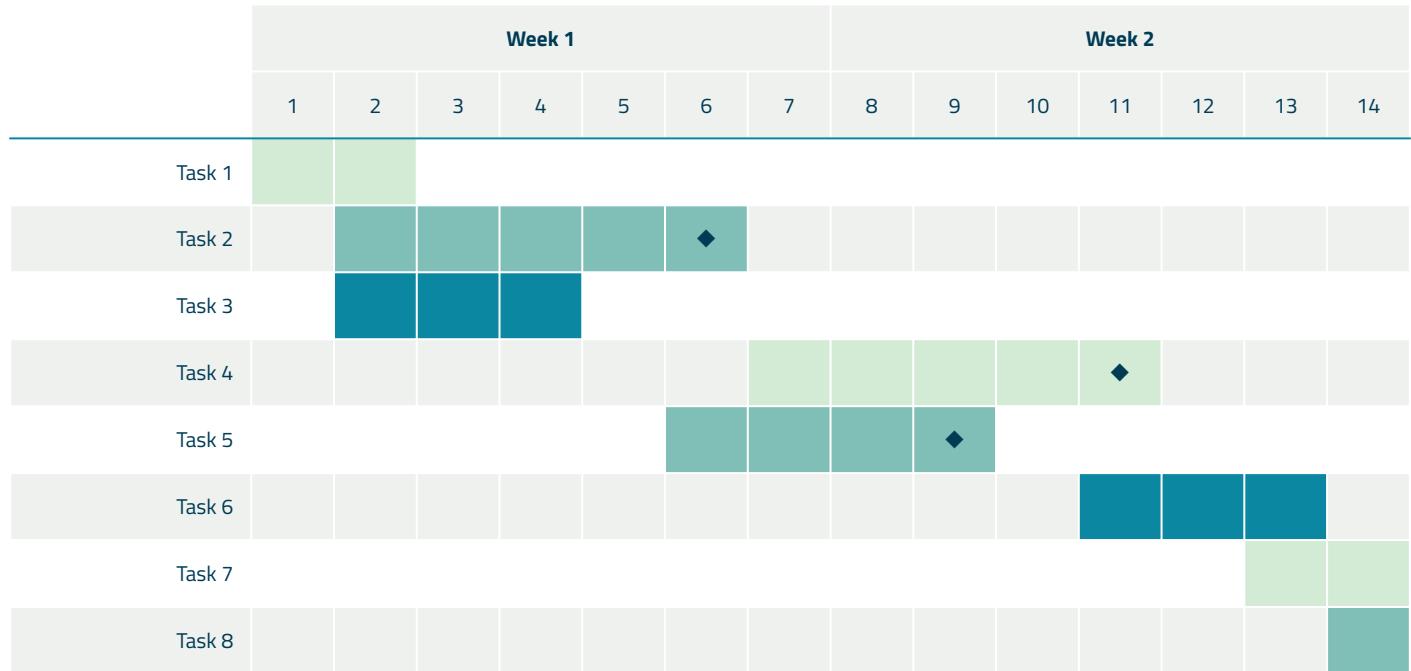
4

Blue is the colour of the clear sky and the deep sea

6



GANTT CHART



SWOT ANALYSIS

STRENGTHS

Blue is the colour of the clear sky and the deep sea

S

Black is the color of ebony and of outer space

OPPORTUNITIES

WEAKNESSES

Yellow is the color of gold, butter and ripe lemons

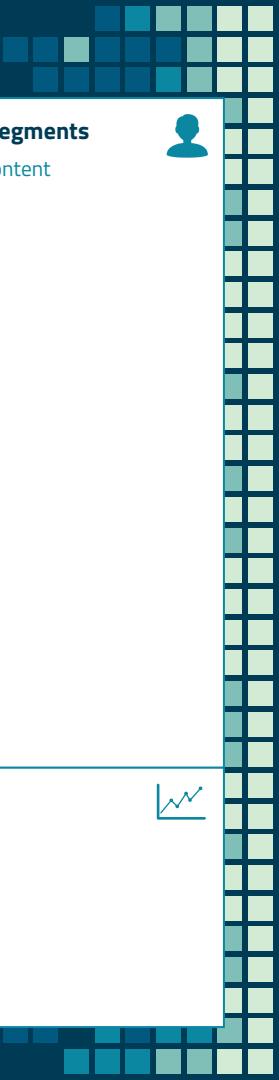
W

White is the color of milk and fresh snow

T

THREATS

BUSINESS MODEL CANVAS



Key Partners Insert your content		Key Activities Insert your content		Value Propositions Insert your content		Customer Relationships Insert your content		Customer Segments Insert your content	
Key Resources Insert your content						Channels Insert your content			
Cost Structure Insert your content				Revenue Streams Insert your content					

FUNNEL



TEAM PRESENTATION



Imani Jackson

JOB TITLE

Blue is the colour of the clear
sky and the deep sea



Marcos Galán

JOB TITLE

Blue is the colour of the clear
sky and the deep sea



Ixchel Valdía

JOB TITLE

Blue is the colour of the clear
sky and the deep sea



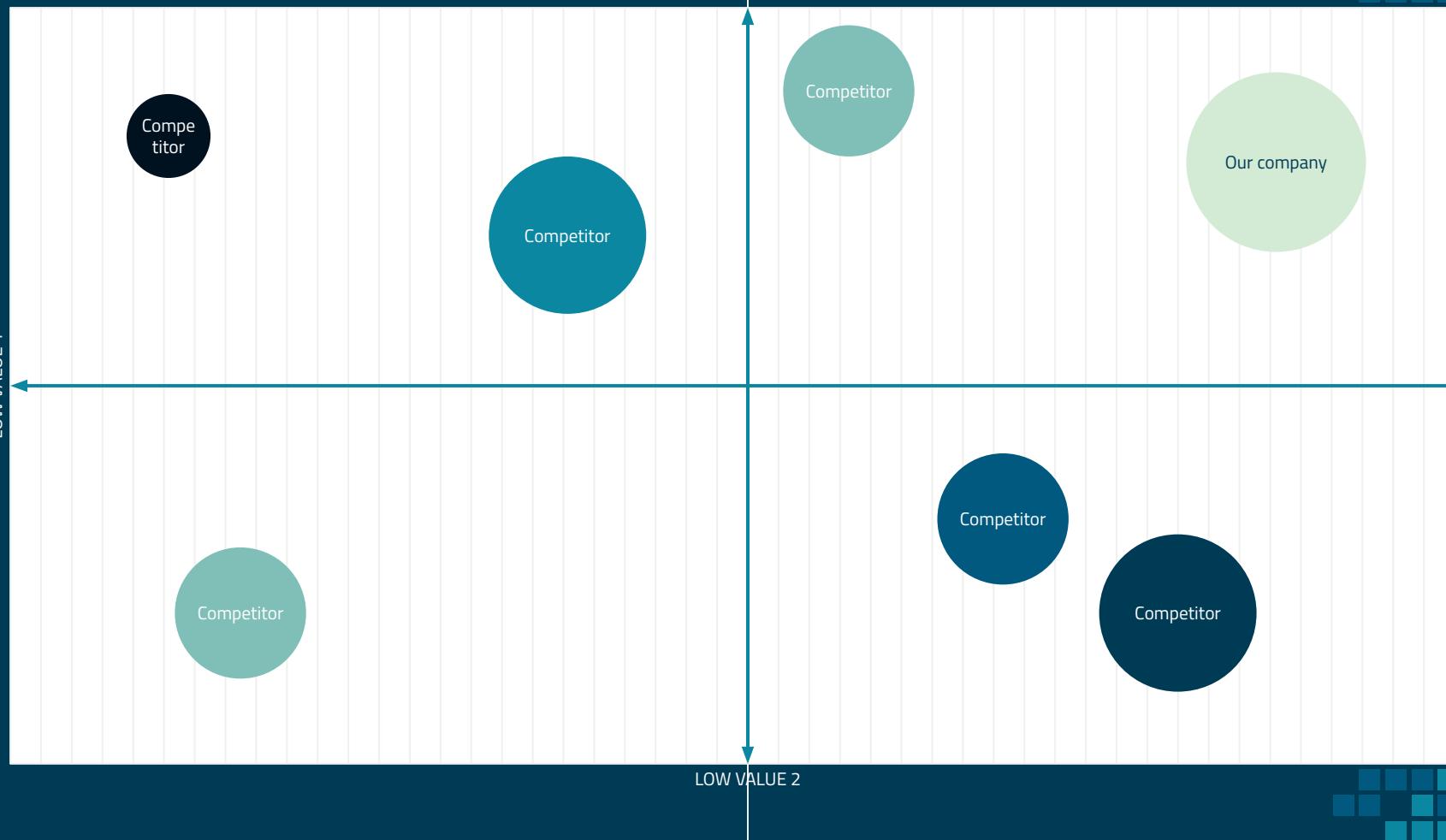
Nils Årud

JOB TITLE

Blue is the colour of the clear
sky and the deep sea



COMPETITOR MATRIX



WEEKLY PLANNER

	SUNDAY	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY
9:00 - 9:45	Task						
10:00 - 10:45	Task						
11:00 - 11:45	Task						
12:00 - 13:15	✓ Free time						
13:30 - 14:15	Task						
14:30 - 15:15	Task						
15:30 - 16:15	Task						



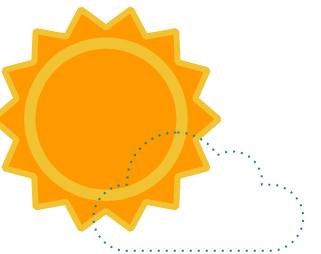
SlidesCarnival icons are editable shapes.

This means that you can:

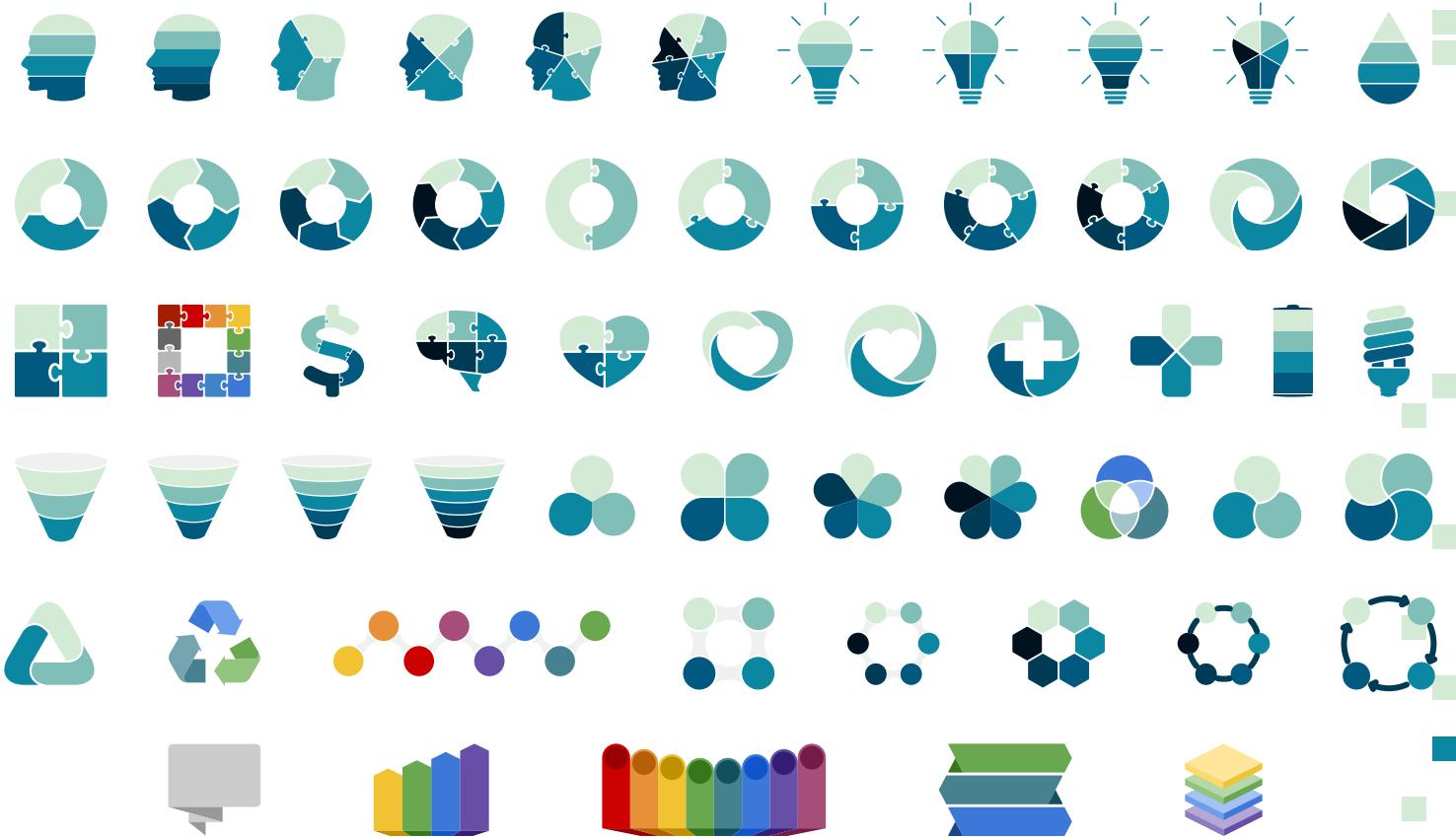
- Resize them without losing quality.
- Change fill color and opacity.
- Change line color, width and style.

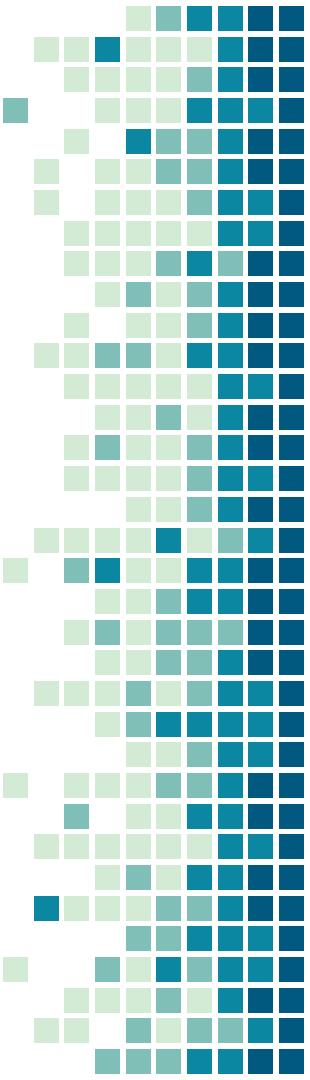
Isn't that nice? :)

Examples:



DIAGRAMS AND INFOGRAPHICS





You can also use any emoji as an icon!

And of course it resizes without losing quality.

How? Follow Google instructions

<https://twitter.com/googledocs/status/730087240156643328>



and many more...



Free templates for all your presentation needs



For PowerPoint and
Google Slides



100% free for personal
or commercial use

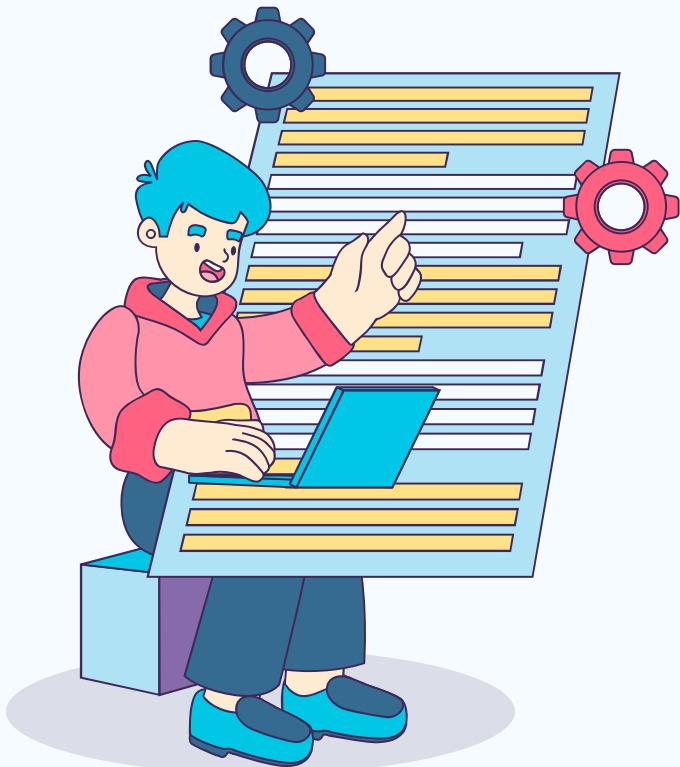


Ready to use,
professional and
customizable



Blow your audience
away with attractive
visuals

...
...



ResNet-50

Deep Residual Learning
for Image Recognition



Table of contents

01

Introducción

02

Arquitectura

03

Dataset

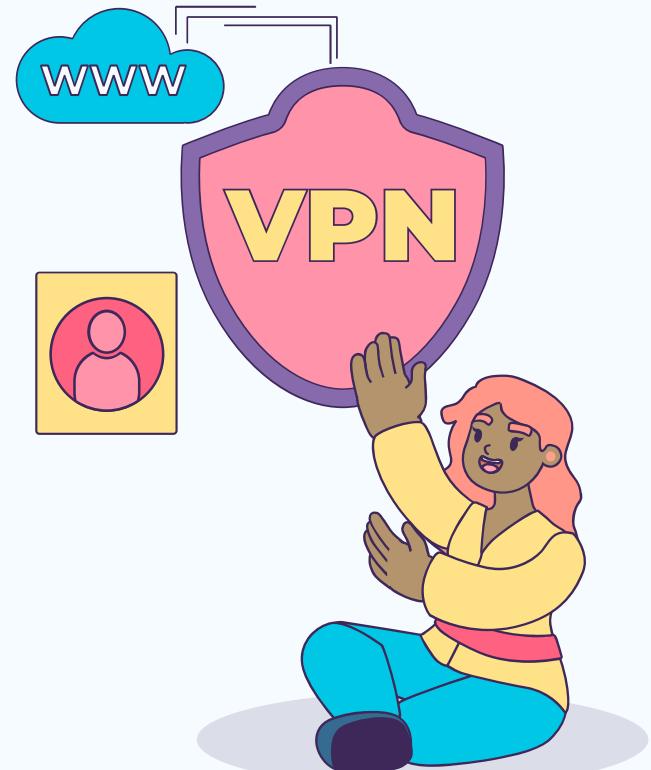
04

Áreas de aplicación

000

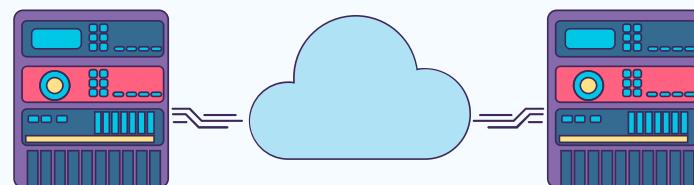
01

Introducción



Introducción

- Resnet50 se refiere a Residual Network que utiliza 50 capas convolucionales.
- Introducido en 2015 por He Kaiming, Zhang Xiangyu, Ren Shaoqing, and Sun Jian en el paper “Deep Residual Learning for Image Recognition”.
- Demostrar que estas redes residuales son más fáciles de optimizar y pueden ganar precisión con una profundidad considerablemente mayor.



Deep Residual Learning for Image Recognition

Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun

Microsoft Research

{kahe, v-xiangz, v-shren, jiansun}@microsoft.com

Abstract

Deeper neural networks are more difficult to train. We present a residual learning framework to ease the training of networks that are substantially deeper than those used previously. We explicitly reformulate the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. We provide comprehensive empirical evidence showing that these residual networks are easier to optimize, and can gain accuracy from

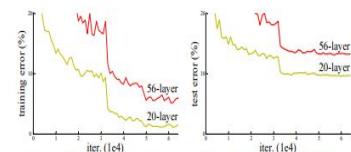


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer “plain” networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

¿Aprender mejores redes es tan fácil como apilar más capas?

- Agregar más capas introduce algunos problemas durante el entrenamiento como:



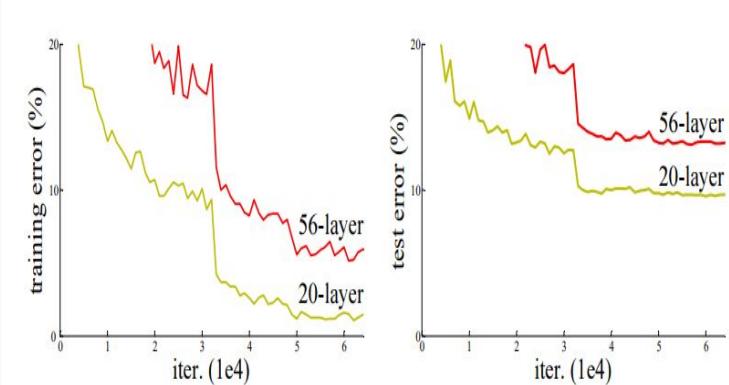
Problema de los gradientes desvanecidos/explorosivos

- Dificulta la convergencia desde el principio.
- Solución:
 - Inicialización normalizada
 - Capas de normalización intermedias



Problema de degradación

- Las redes más profundas son capaces de empezar a converger.
- Añadir más capas al modelo conduce un mayor error de entrenamiento.
- No es causada por el overfitting.

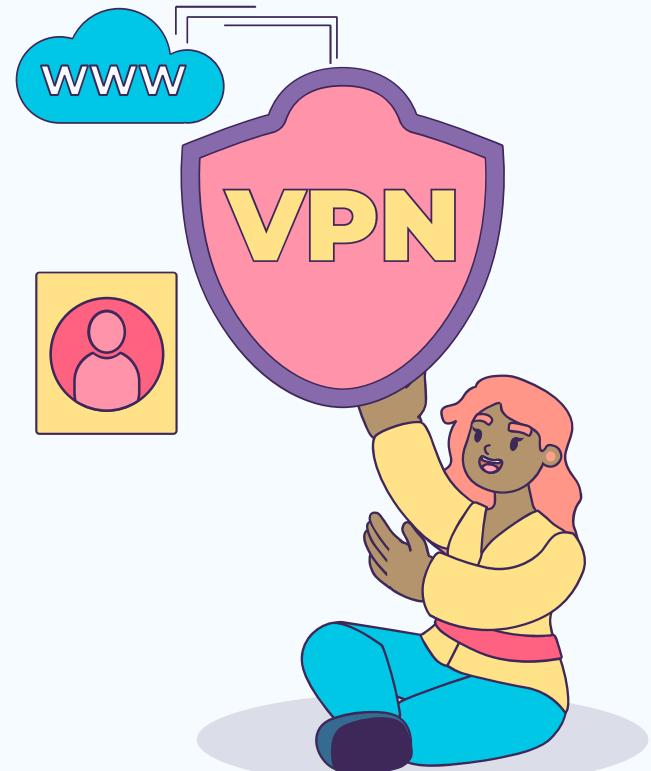


Dataset: CIFAR - 10

000

02

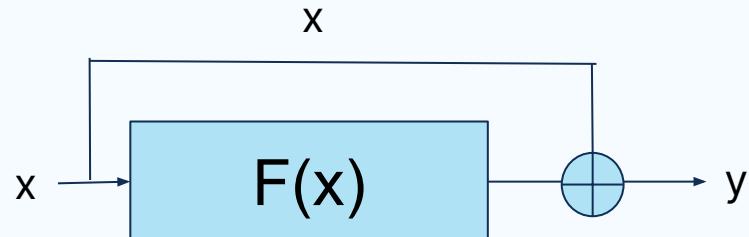
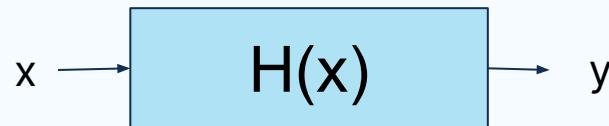
Arquitectura



Residual Learning

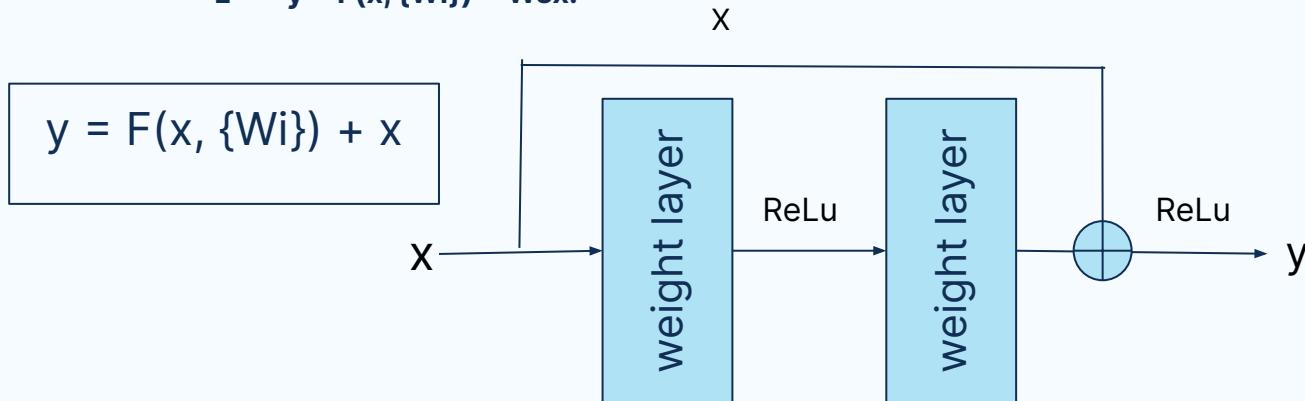
- $H(x)$: mapeo subyacente que debe ser ajustada por unas cuantas capas apiladas
- x : Entradas a la primera de estas capas
- En lugar de esperar que las capas apiladas aproximen $H(x)$, dejamos explícitamente que estas capas aproximen una función residual:

$$F(x) := H(x) - x$$



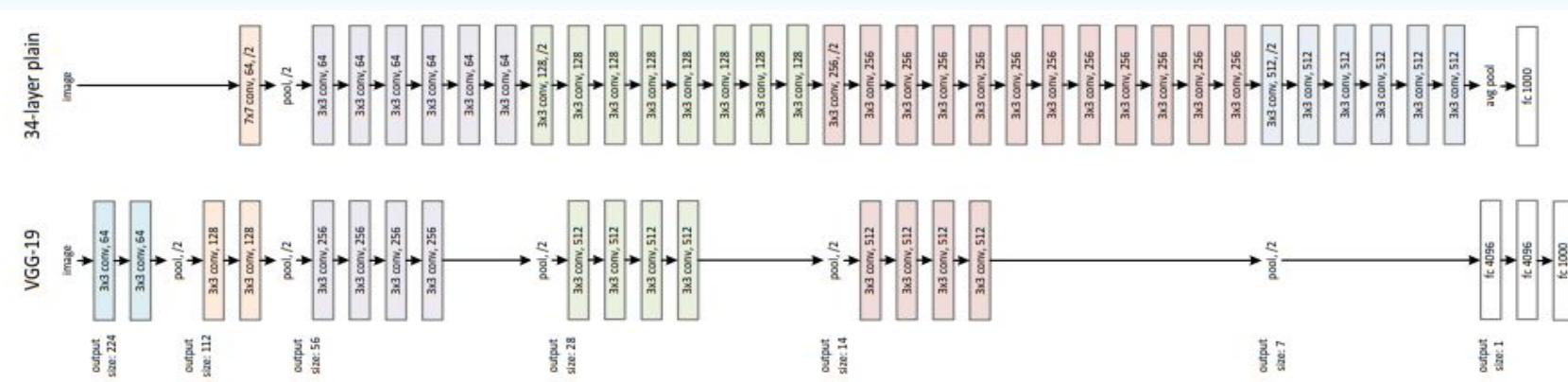
Residual Block

- **$F(x, \{W_i\})$:** Representa el mapeo residual que debe aprenderse.
- La operación $F + x$ se realiza mediante una conexión abreviada y una suma de elementos.
- Dimensiones diferentes (entrada/salida):
 - Realizar una proyección lineal W_s por las conexiones de acceso directo para igualar las dimensiones:
 - $y = F(x, \{W_i\}) + W_s x.$



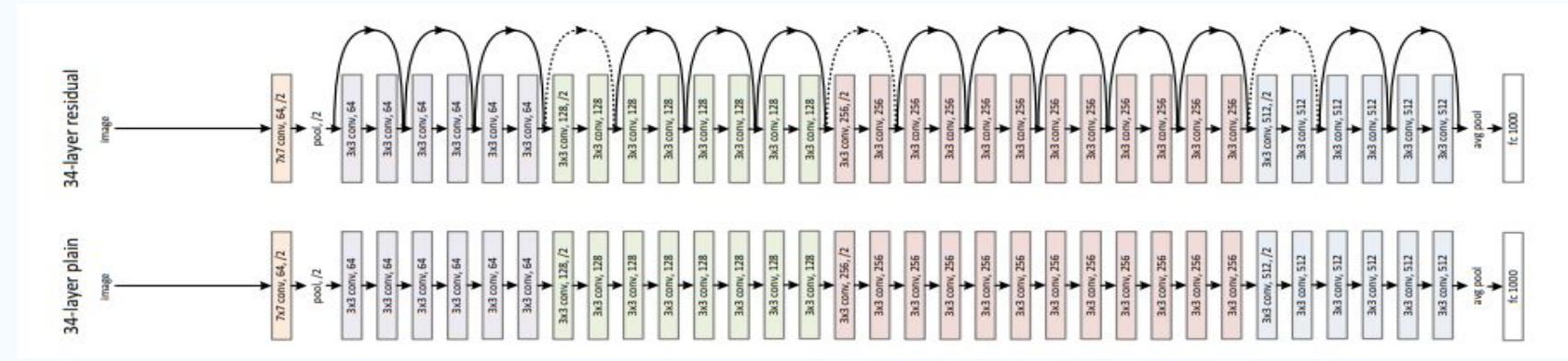
Arquitectura

- **PLAIN NETWORK:**
- Las capas convolucionales tienen en su mayoría filtros de 3×3
 - Para el mismo tamaño del mapa de características de salida, las capas tienen el mismo número de filtros.
 - Si el tamaño del mapa de características se reduce a la mitad, el número de filtros se duplica para preservar la complejidad temporal por capa.
- El modelo básico de 34 capas tiene 3.600 millones de FLOPs, en cambio, VGG-19 tiene 19.600 millones de FLOPs.



Arquitectura

- **RESIDUAL NETWORK:**
- Inserta conexiones de atajo que convierten la red en su versión residual equivalente.
- Cuando hay diferencia en las dimensiones, existen dos opciones:
 - El atajo sigue realizando el mapeo de identidad, rellena con entradas cero adicionales para aumentar la dimensionalidad.
 - Se utiliza la ecuación del atajo de proyección, igualando las dimensiones (mediante convoluciones 1×1).

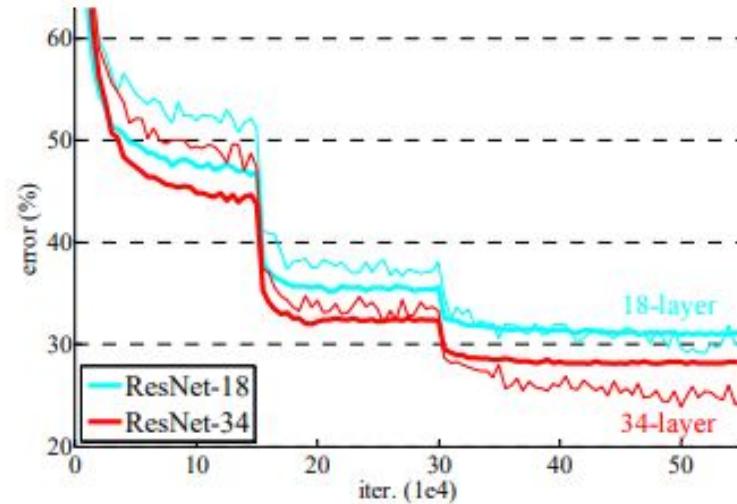
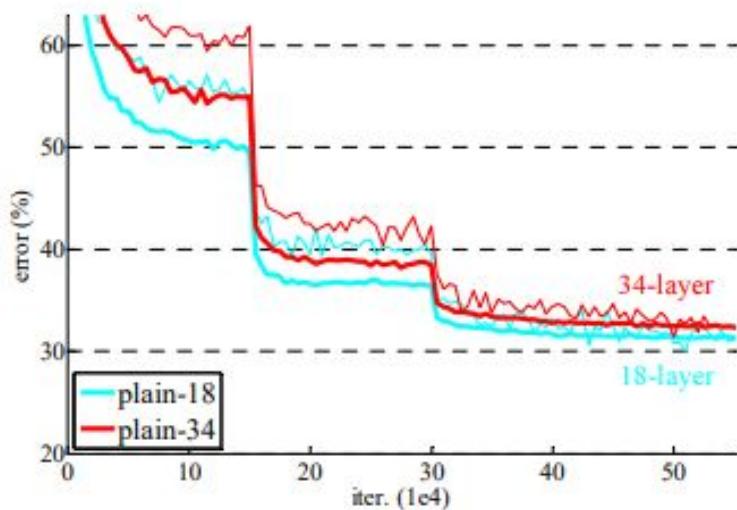


Implementación

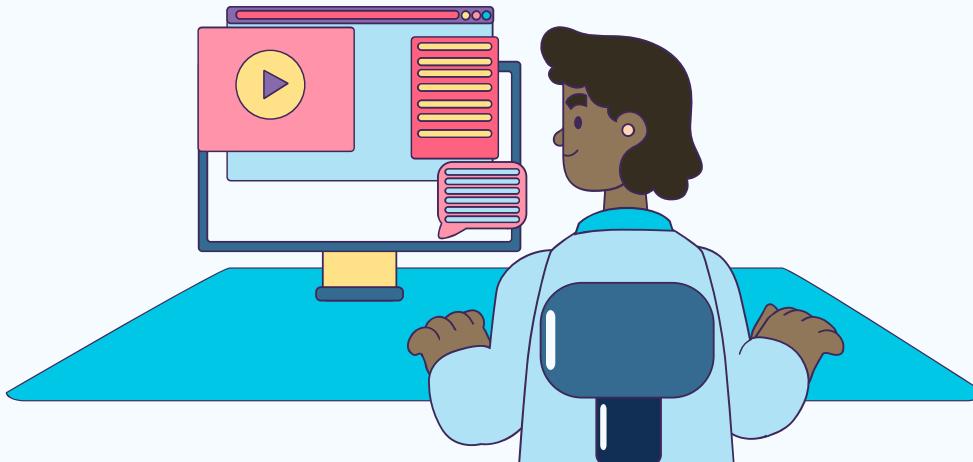
La arquitectura ResNet incluye los siguientes elementos:

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112			7×7, 64, stride 2		
conv2_x	56×56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Resultados



03 DATASET





14,197,122 images, 21841 synsets indexed

[Home](#) [Download](#) [Challenges](#) [About](#)Not logged in. [Login](#) | [Signup](#)

ImageNet Large Scale Visual Recognition Challenge (ILSVRC)

Competition

The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) evaluates algorithms for object detection and image classification at large scale. One high level motivation is to allow researchers to compare progress in detection across a wider variety of objects -- taking advantage of the quite expensive labeling effort. Another motivation is to measure the progress of computer vision for large scale image indexing for retrieval and annotation.

For details about each challenge please refer to the corresponding page.

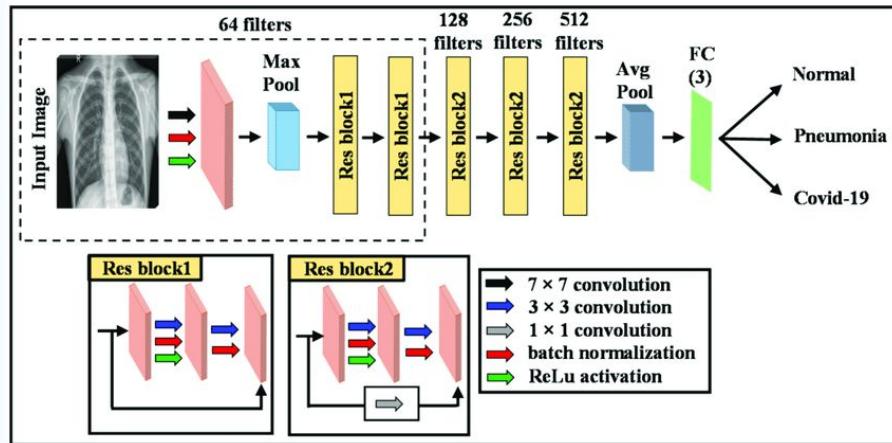
- [ILSVRC 2017](#)
- [ILSVRC 2016](#)
- [ILSVRC 2015](#)
- [ILSVRC 2014](#)
- [ILSVRC 2013](#)
- [ILSVRC 2012](#)
- [ILSVRC 2011](#)
- [ILSVRC 2010](#)

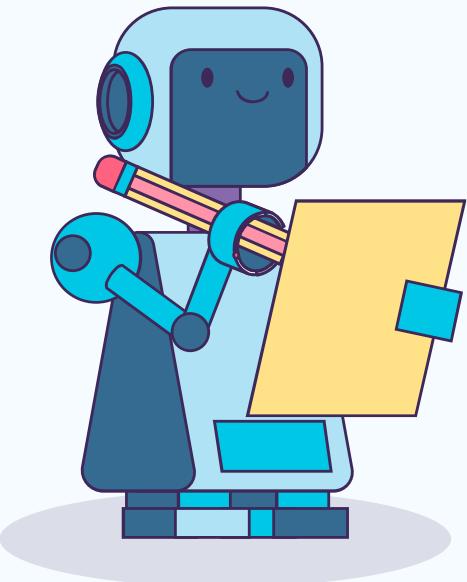
04

Áreas de aplicación



- Clasificación de imágenes
- Detección de objetos
- Segmentación semántica
- Reconocimiento facial
- Procesamiento de imágenes médicas
- Análisis de imágenes en tiempo real





Thanks!

Inception-v1

Integrantes:

- Perca Quispe Joel Cristian
- Espinoza Peñaloza Edgar Alfonso
- Vilchez Molina Misael Svante

CONTENIDOS

01

INTRODUCCIÓN

02

ARQUITECTURA

03

CONJUNTO DE
DATOS

04

APLICACIONES

05

DEMO



Going deeper with convolutions

Christian Szegedy
Google Inc.

Wei Liu
University of North Carolina, Chapel Hill

Yangqing Jia
Google Inc.

Pierre Sermanet
Google Inc.

Scott Reed
University of Michigan

Dragomir Anguelov
Google Inc.

Dumitru Erhan
Google Inc.

Vincent Vanhoucke
Google Inc.

Andrew Rabinovich
Google Inc.

Abstract

We propose a deep convolutional neural network architecture codenamed Inception, which was responsible for setting the new state of the art for classification and detection in the ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC14). The main hallmark of this architecture is the improved utilization of the computing resources inside the network. This was achieved by a carefully crafted design that allows for increasing the depth and width of the network while keeping the computational budget constant. To optimize quality, the architectural decisions were based on the Hebbian principle and the intuition of multi-scale processing. One particular incarnation used in our submission for ILSVRC14 is called GoogLeNet, a 22 layers deep network, the quality of which is assessed in the context of classification and detection.

1 Introduction

In the last three years, mainly due to the advances of deep learning, more concretely convolutional networks [10], the quality of image recognition and object detection has been progressing at a dramatic pace. One encouraging news is that most of this progress is not just the result of more powerful hardware, larger datasets and bigger models, but mainly a consequence of new ideas, algorithms and improved network architectures. No new data sources were used, for example, by the top entries in the ILSVRC 2014 competition besides the classification dataset of the same competition for detection purposes. Our GoogLeNet submission to ILSVRC 2014 actually uses $12\times$ fewer parameters than the winning architecture of Krizhevsky et al [9] from two years ago, while being significantly more accurate. The biggest gains in object-detection have not come from the utilization of deep networks alone or bigger models, but from the synergy of deep architectures and classical computer vision, like the R-CNN algorithm by Girshick et al [6].

Another notable factor is that with the ongoing traction of mobile and embedded computing, the efficiency of our algorithms – especially their power and memory use – gains importance. It is noteworthy that the considerations leading to the design of the deep architecture presented in this paper included this factor rather than having a sheer fixation on accuracy numbers. For most of the experiments, the models were designed to keep a computational budget of 1.5 billion multiply-adds at inference time, so that they do not end up to be a purely academic curiosity, but could be put to real world use, even on large datasets, at a reasonable cost.

1. INTRODUCCIÓN

- El enfoque es abordar el reconocimiento de imágenes utilizando redes convolucionales.
- El paper introduce la arquitectura Inception-v1, que se caracteriza por su enfoque modular y el uso de módulos Inception.

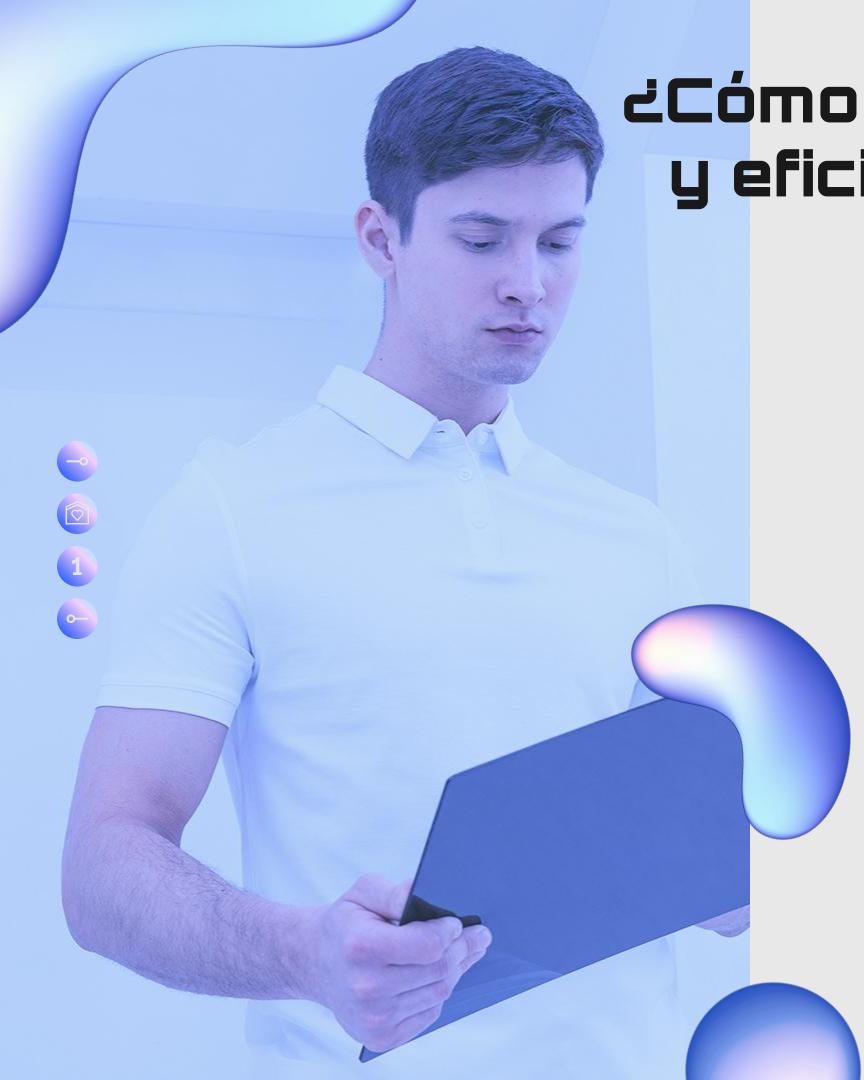
Desafíos que aborda Inception-v1



Dos clases distintas de entre las 1000 clases del ILSVRC 2014.
Se requiere conocimiento del dominio para distinguir entre estas
clases.

- Aumento en la cantidad de parámetros
- Dificultad de entrenamiento a medida que se agregan capas adicionales.



A photograph of a young man with short brown hair, wearing a white button-down shirt. He is looking down at a dark tablet or laptop screen he is holding in his hands. The background is a light blue gradient.

¿Cómo se puede mejorar la precisión y eficiencia en el reconocimiento de imágenes?

La arquitectura de **Inception.v1** utiliza módulos que realizan convoluciones en paralelo con filtros de diferentes tamaños permitiendo: y combinando eficientemente la información Enriquecida en una sola capa.

- Capturar características en diferentes escalas al realizar convoluciones en paralelo con filtros de diferentes tamaños.
- Combinar eficientemente la información Enriquecida en una sola capa.

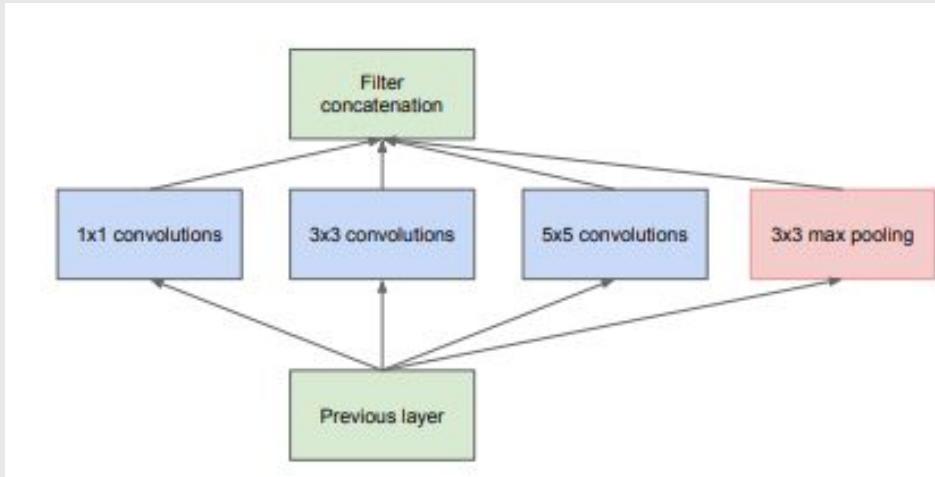
Reducción de dimensionalidad: Capas 1x1

- Uso de capas de reducción de dimensionalidad en Inception-v1.
- Las convoluciones 1x1 se utilizan para disminuir la complejidad computacional y reducir la cantidad de parámetros en la red.
- La reducción de dimensionalidad permite que la red sea más profunda y eficiente sin sacrificar el rendimiento.



ARQUITECTURA

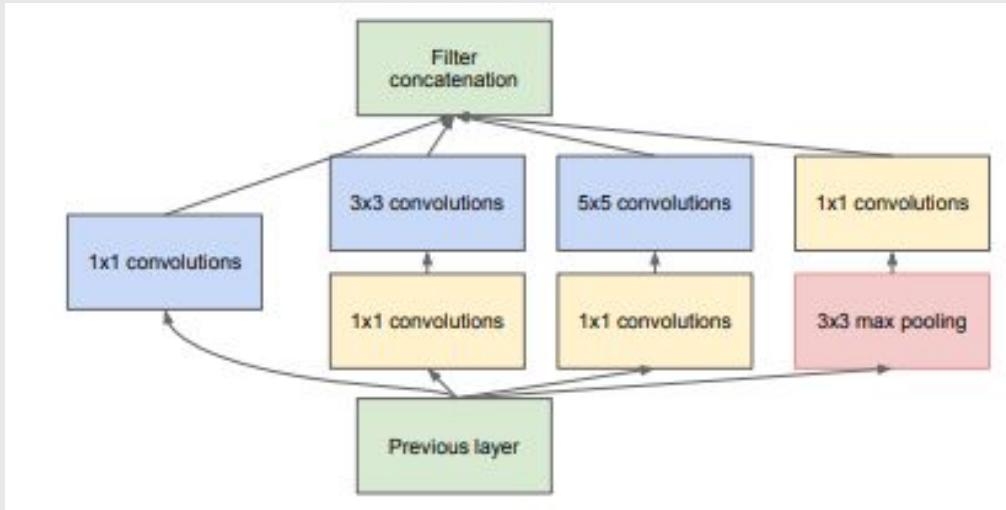
Estructura principal



Módulo Inception naive-version

- Un módulo Inception está compuesto por múltiples rutas de convolución en paralelo con diferentes tamaños de filtro.
- Cada ruta utiliza convoluciones de 1x1, 3x3 y 5x5 para capturar características en diferentes escalas.

ARQUITECTURA



Módulo Inception con reducción de dimensionalidad

Factor de Escala: Convolución 1x1

- Las convoluciones 1x1 se utilizan para ajustar la dimensionalidad y el número de canales antes de las convoluciones de mayor tamaño.
- El factor de escala ayuda a controlar la cantidad de cálculos y parámetros de la red.

ARQUITECTURA

12	20	30	0
8	12	2	0
34	70	37	4
112	100	25	12

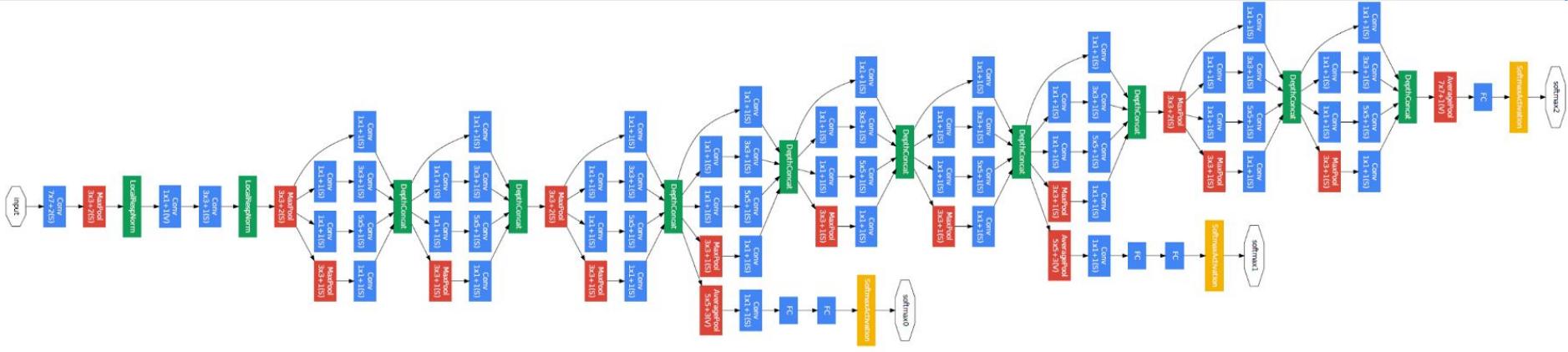
2×2 Max-Pool

20	30
112	37

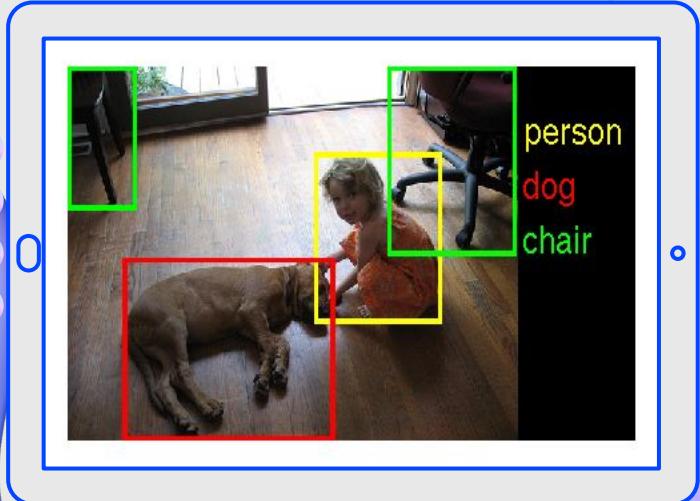
Capa de Pooling: Reducción de Espacio

- La capa de pooling utilizada en Inception-v1 para reducir el tamaño espacial de los mapas de características.
- El pooling reduce la resolución espacial y el costo computacional en comparación con las convoluciones tradicionales.

ARQUITECTURA



CONJUNTO DE DATOS: DETECCIÓN



- 200 categorías de nivel básico completamente anotadas en los datos de prueba.
- Las categorías se eligieron cuidadosamente teniendo en cuenta distintos factores.
- Algunas de las imágenes de prueba no contienen ninguna de las 200 categorías.

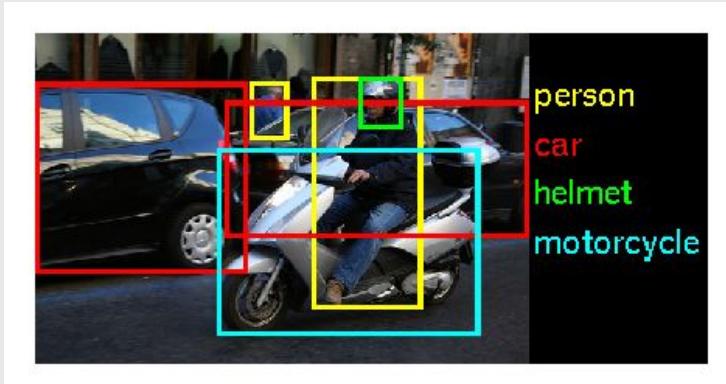
CONJUNTO DE DATOS: DETECCIÓN

	ENTRENAMIENTO		VALIDACIÓN		PRUEBA	
NÚMERO DE CLASES DE OBJETOS	NÚMERO DE IMÁGENES	NÚMERO DE OBJETOS	NÚMERO DE IMÁGENES	NÚMERO DE OBJETOS	NÚMERO DE IMÁGENES	NÚMERO DE OBJETOS
200	395909	345854	20121	55502	40152	---





- **Average image resolution:** 482x415 pixels
- **Average object classes per image:** 1.534
- **Average object instances per image:** 2.758
- **Average object scale:** 0.170



CONJUNTO DE DATOS: CLASIFICACIÓN Y CLASIFICACIÓN CON LOCALIZACIÓN

- Conjunto de validación y test consisten en 150000 fotografías.
- Etiquetas con la presencia o ausencia de 1000 categorías de objetos.
- Subconjunto aleatorio de 50000 de las imágenes con etiquetas se utilizaron como datos de validación junto con una lista de las 1.000 categorías.
- Las imágenes restantes se utilizaron para la evaluación sin proveer sus etiquetas.

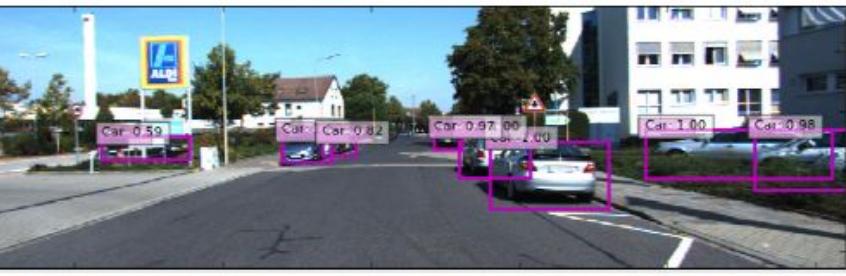
Team	Year	Place	Error (top-5)	Uses external data
SuperVision	2012	1st	16.4%	no
SuperVision	2012	1st	15.3%	Imagenet 22k
Clarifai	2013	1st	11.7%	no
Clarifai	2013	1st	11.2%	Imagenet 22k
MSRA	2014	3rd	7.35%	no
VGG	2014	2nd	7.32%	no
GoogLeNet	2014	1st	6.67%	no

Number of models	Number of Crops	Cost	Top-5 error	compared to base
1	1	1	10.07%	base
1	10	10	9.15%	-0.92%
1	144	144	7.89%	-2.18%
7	1	7	8.09%	-1.98%
7	10	70	7.62%	-2.45%
7	144	1008	6.67%	-3.45%

Desafío ImageNet

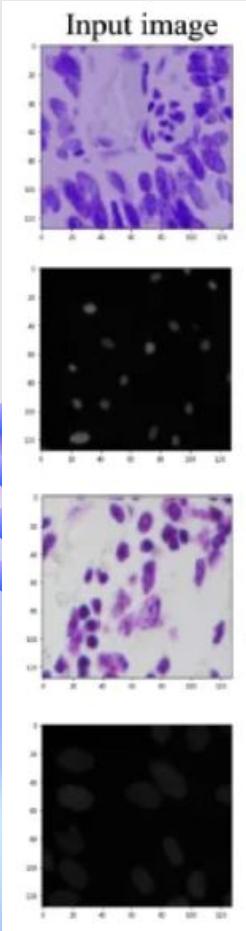
- Inception-v1 ha sido evaluado en el desafío ImageNet, una competencia de clasificación de imágenes.
- Inception-v1 logró una precisión de top-1 del 69.8% y una precisión de top-5 del 89.8% en este desafío.
- Inception-v1 es capaz de clasificar imágenes en diversas categorías, como personas, animales, vehículos, edificios, paisajes, entre otros.

Detección de objetos



- Inception-v1 se ha utilizado para la detección precisa de objetos en imágenes.
- Puede combinarse con técnicas de detección para localizar y etiquetar objetos en imágenes con alta precisión.

Segmentación semántica



- Inception-v1 es aplicable en tareas de segmentación semántica.
- Puede proporcionar mapas detallados y precisos de segmentación al fusionar características de diferentes escalas.

Conclusiones

- La arquitectura Inception-v1 ha logrado un rendimiento sobresaliente en el desafío de clasificación de imágenes ILSVRC 2014, superando a los enfoques existentes.
- Su enfoque de diseño, como el uso de módulos de Inception y la reducción de dimensionalidad, ha demostrado ser efectivo en la captura de características y la mejora de la precisión.
- Inception-v1 ha demostrado ser eficiente en términos de uso de recursos computacionales en comparación con otras arquitecturas.
- La escalabilidad de la arquitectura permite adaptarse a diferentes tamaños de imágenes y requisitos de recursos, lo que facilita su implementación en diversas aplicaciones.

Thanks

LONG SHORT-TERM MEMORY NETWORK

- Jack Christopher Huaihua Huayhua
- Angel Tomas Concha Layme
- Jean Pierre Chavez Guevara

CONTENIDO

01

Artículo

02

Estructura

03

Datos de
entrenamiento

04

Áreas de
aplicación

05

Ventajas y
desventajas

05

Conclusiones





O1

Artículo

LONG SHORT TERM MEMORY

Año: 1997

Medio de publicación: Journal Neural Computation.

Autores: Sepp Hochreiter y Jürgen Schmidhuber.

LONG SHORT-TERM MEMORY

NEURAL COMPUTATION 9(8):1735–1780, 1997

Sepp Hochreiter

Fakultät für Informatik
Technische Universität München
80290 München, Germany
hochreit@informatik.tu-muenchen.de

<http://www7.informatik.tu-muenchen.de/~hochreit>

Jürgen Schmidhuber

IDSIA
Corso Elvezia 36
6900 Lugano, Switzerland
juergen@idsia.ch

<http://www.idsia.ch/~juergen>

MOTIVO

Superar problemas de desvanecimiento y explosión de errores en RNN tradicionales.

IMPACTO

Aplicaciones en reconocimiento de voz, procesamiento del lenguaje natural y predicciones secuenciales.

PUBLICACIÓN

Revista científica especializada en inteligencia artificial, aprendizaje automático o procesamiento de lenguaje natural.

RESULTADOS

Capacidad para resolver problemas complejos con largos retrasos temporales.





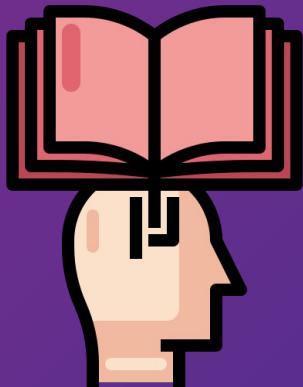
O2

Estructura

Idea Principal

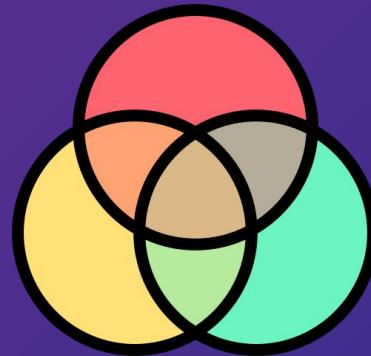
Memorización

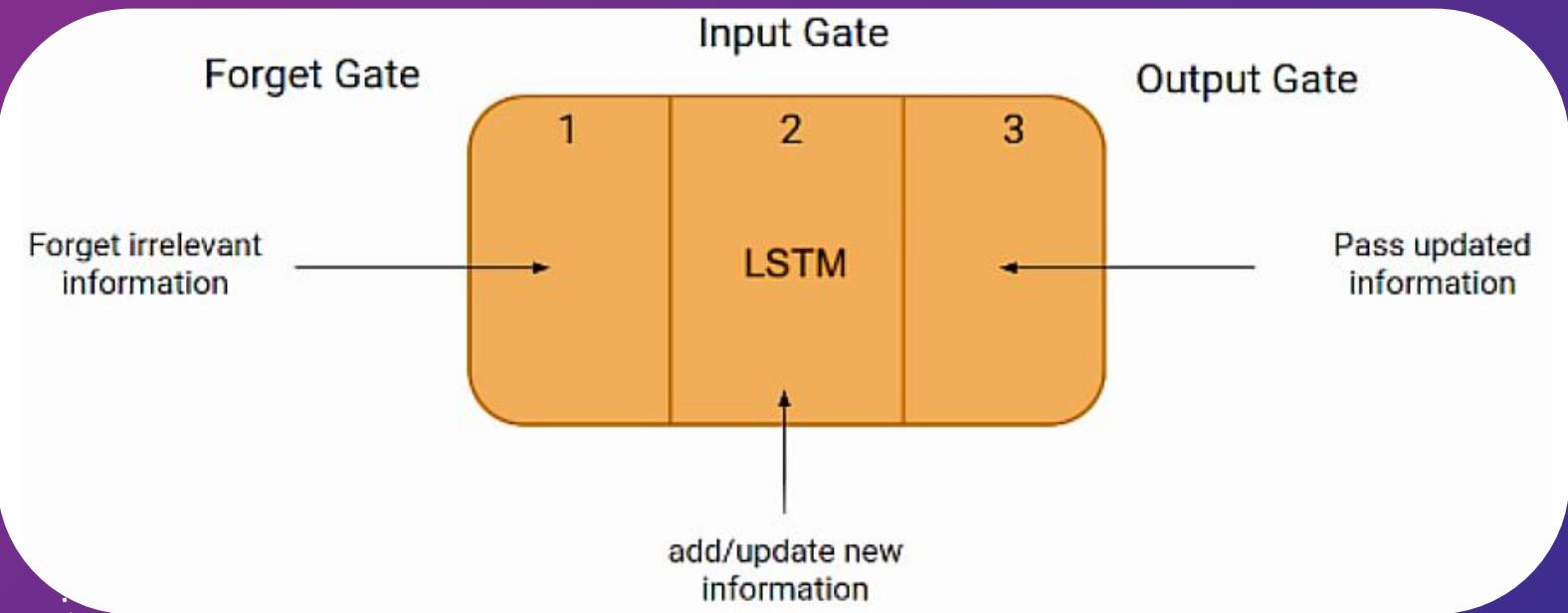
Se gana información → se almacena para uso futuro

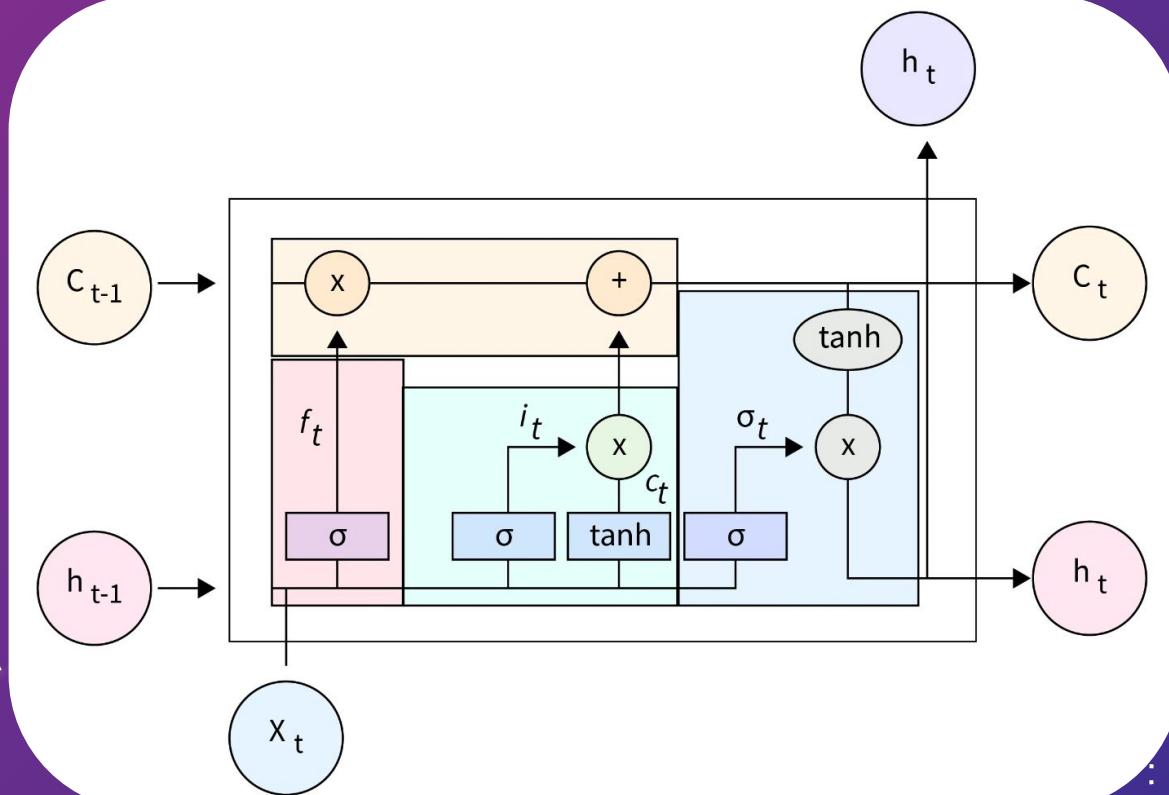


Combinación

Información almacenada + habilidades analíticas





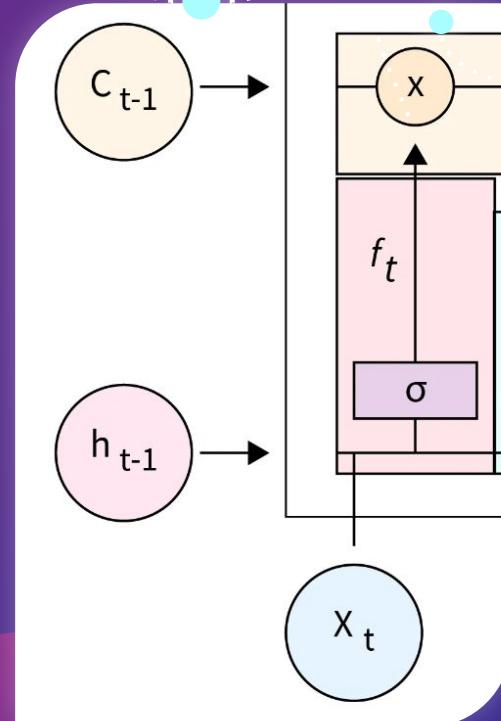


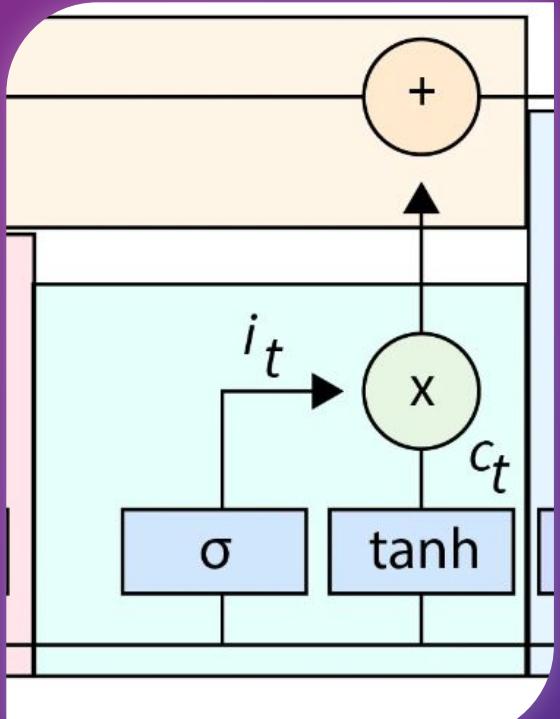
FORGET GATE

Decide qué información debe eliminarse del **estado de la celda**.

Toma el **estado oculto** anterior y la entrada actual y lo pasa a una función de activación Sigmóide.

Genera un valor entre 0 y 1, donde 0 significa olvidar y 1 significa mantener.





INPUT GATE

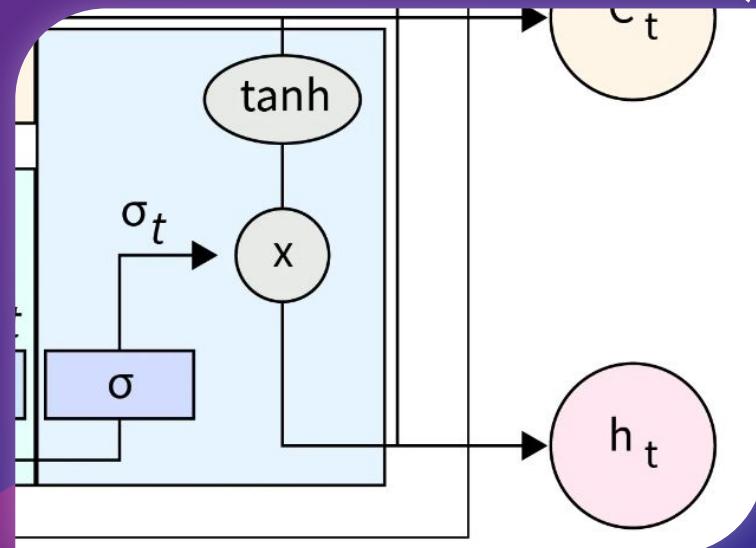
Considera la **entrada actual** y el **estado oculto** anterior para actualizar el valor del **estado de la celda**.

- Función de activación de Sigmoide → Decide qué porcentaje de la información se requiere.
- Función de activación de Tanh → Mapea los datos entre -1 y 1, luego la multiplica por la función Sigmoide.

OUTPUT GATE

Devuelve el **estado oculto** para la próxima ocasión.

- Función de activación Sigmoide → Decide el porcentaje de información relevante requerida.
- Función de activación Tanh → Toma el **estado de la celda** actualizada y la multiplica por la función de activación Sigmoide.

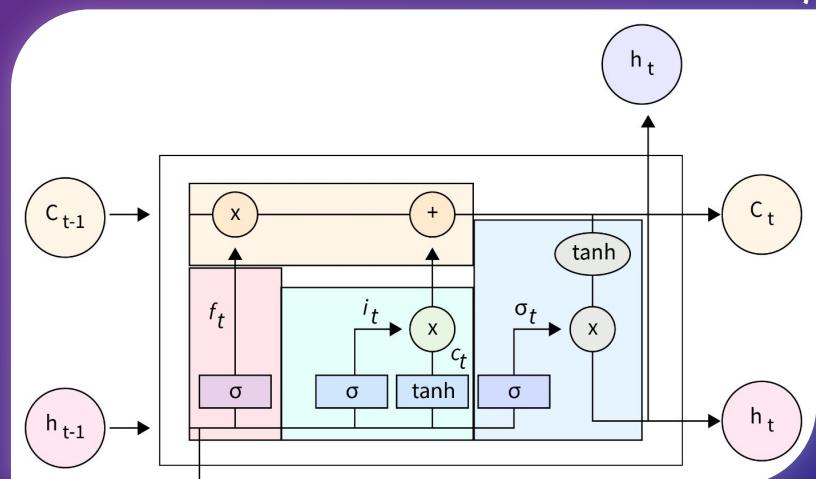


CELL STATE

La **puerta de olvido** y la **puerta de entrada** actualizan la **celda de estado**.

La **celda de estado** anterior se multiplica por la salida de la **puerta de olvido**, se suma con la salida de la **puerta de entrada**.

Este valor se usa luego para calcular el estado oculto en la puerta de salida.





03

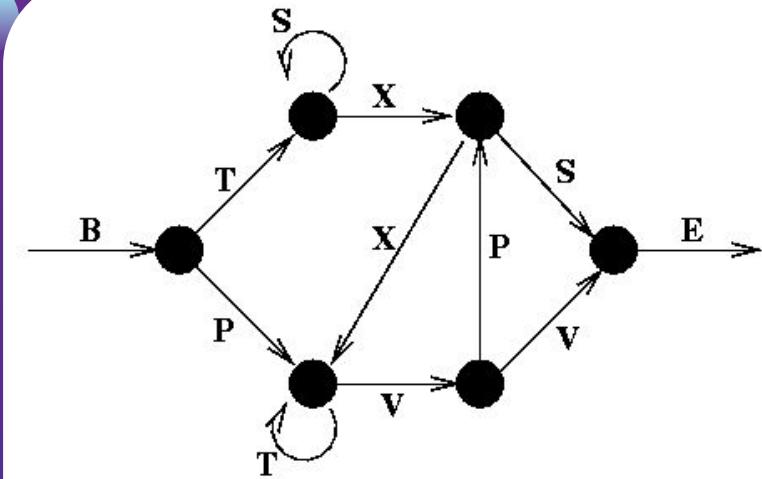
Datos de entrenamiento

Embedded Reber

Es una gramatica

Genera una secuencia de caracteres

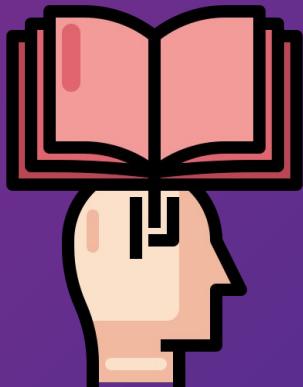
Sigue una logica determinista



LSTM - REBER

APRENDER

Las reglas que usaba la gramática



PREDICIR

Los caracteres que siguen

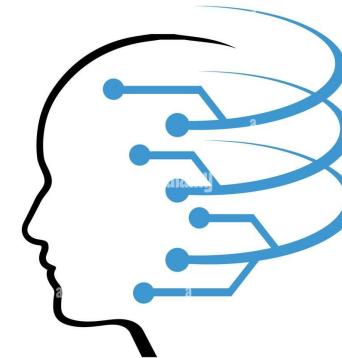


Otros datos

Secuencias de datos como texto, audio, video.

Corpus de noticias, libros, artículos científicos, sitios web

Datos de entrenamiento de Stanford Question Answering Dataset (SQuAD)



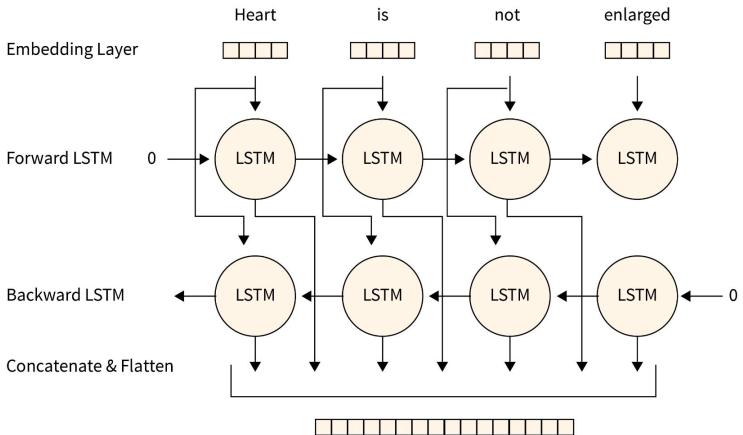
| 04

Áreas de aplicación

Modelado de lenguaje

LSTM es capaz de: crear modelos de lenguaje

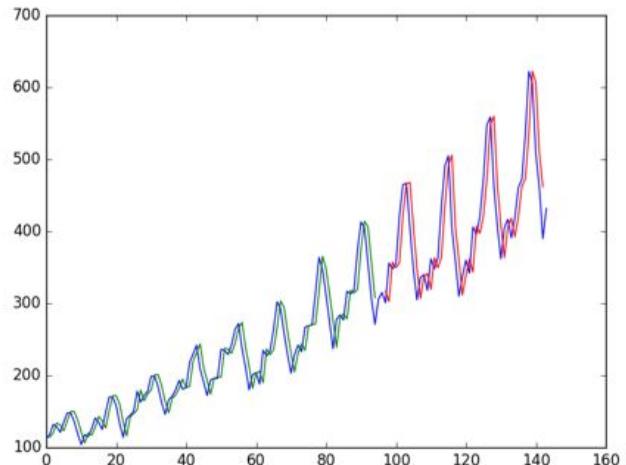
Son usados en: sistemas de traducción automática o chatbots



Predicción de series temporales

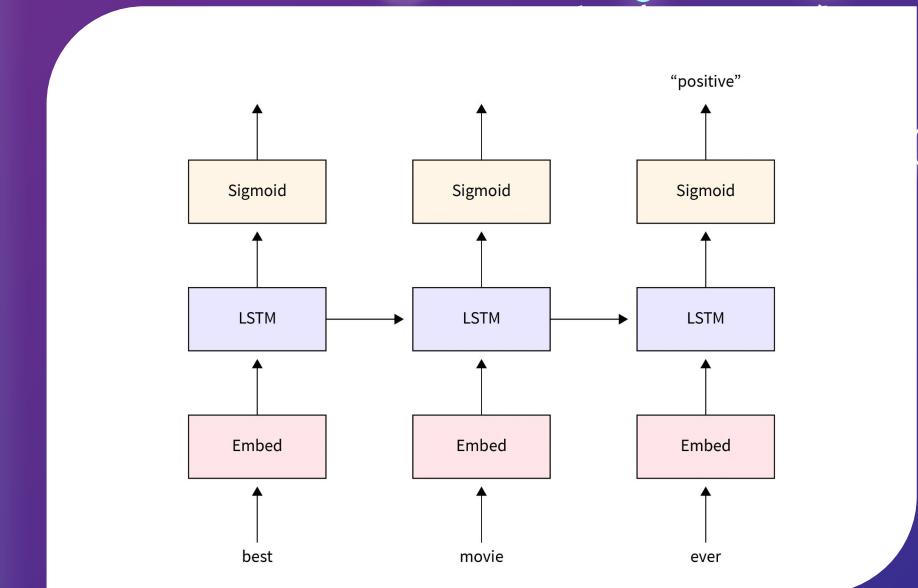
LSTM es capaz de: modelar datos de series temporales y predecir valores futuros en la serie

Son usados para: predecir precios de acciones o patrones de tráfico.



Análisis de sentimientos

LSTM es utilizado para: analizar y clasificar las emociones expresadas en un texto.



Reconocimiento de voz

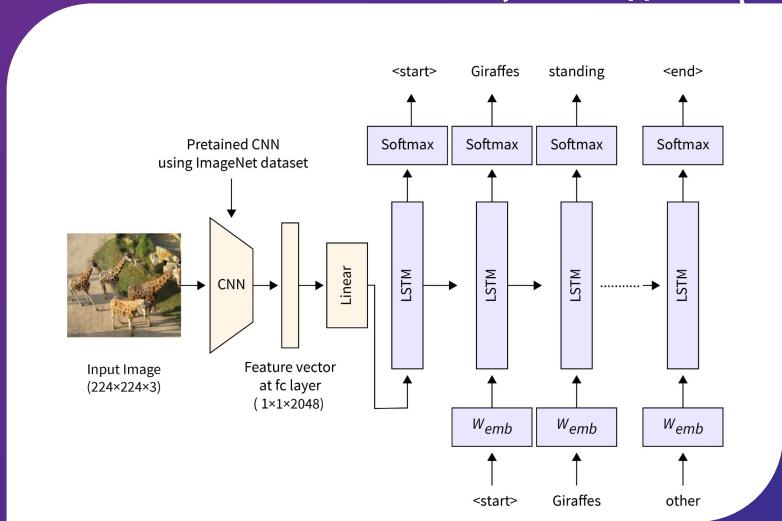
LSTM es capaz de: construir sistemas de reconocimiento de voz

Son usados los siguientes casos: asistentes virtuales , automatización de tareas, accesibilidad y seguridad



Subtítulos de imágenes

LSTM es utilizado para: generar subtítulos descriptivos para imágenes, como en motores de búsqueda de imágenes



05

VENTAJAS Y DESVENTAJAS

VENTAJAS

Captura de dependencias a largo plazo.

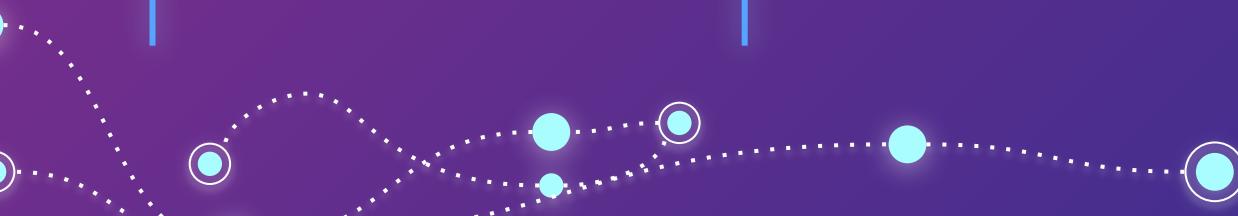
Manejo de secuencias de longitud variable.

Aprendizaje de representaciones jerárquicas.

Memoria de largo plazo.

Resistencia al ruido en los datos.

Adaptabilidad a diferentes dominios.



DESVENTAJAS

Mayor complejidad computacional

Mayor dificultad de interpretación.

Requiere tiempo y experiencia para su configuración.

Mayor consumo de recursos.

Riesgo de sobreajuste.

Sensibilidad a la selección de hiperparámetros

106

CONCLUSIONES

(LSTM) ha sido una arquitectura revolucionaria

Sus logros respaldan su aplicabilidad en diversas áreas

Implementación y tiempo computacional son retos a considerar.