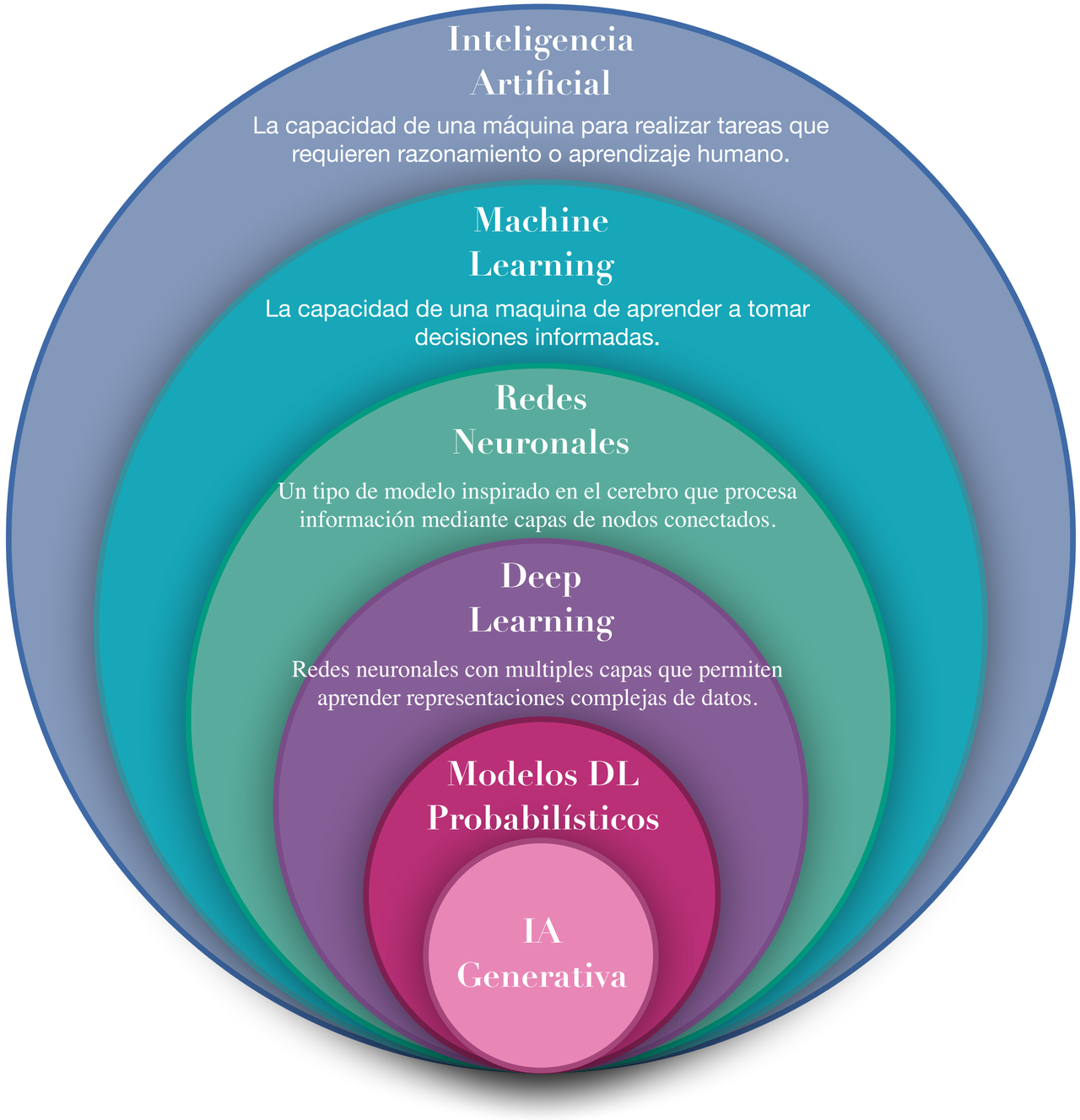
FCEN, UBA

Redes Neuronal Convolucionales

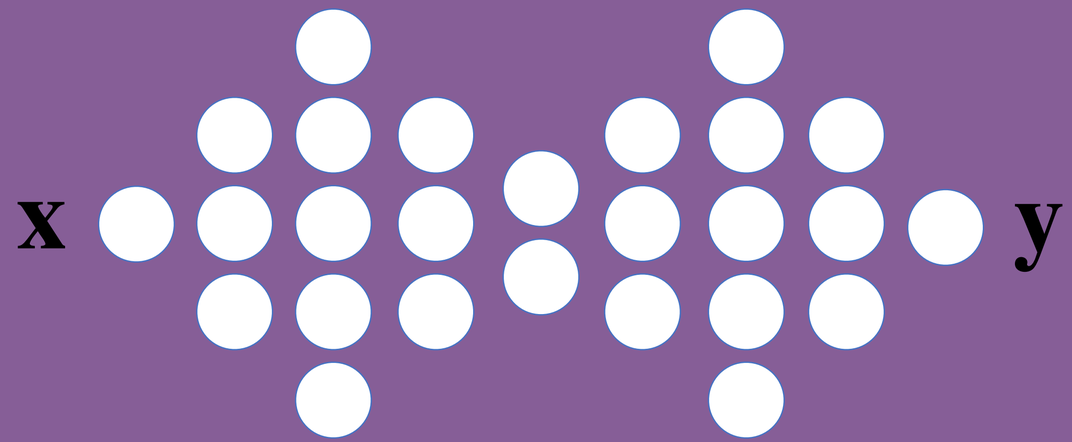
Cecilia Garraffo

Outline

- Desafíos para MLP en Análisis de Imágenes
- De MLP a Redes Neuronales Convolucionales (CNN)
- Como Diseñar y Entrenar una CNN
- Interpretabilidad de CNNs


Inteligencia Artificial
La capacidad de una máquina para realizar tareas que requieren razonamiento o aprendizaje humano.
Machine Learning
La capacidad de una maquina de aprender a tomar decisiones informadas.
Redes Neuronales
Un tipo de modelo inspirado en el cerebro que procesa información mediante capas de nodos conectados.
Deep Learning
Redes neuronales con multiples capas que permiten aprender representaciones complejas de datos.
Modelos DL Probabilísticos
IA Generativa

Deep
Learning

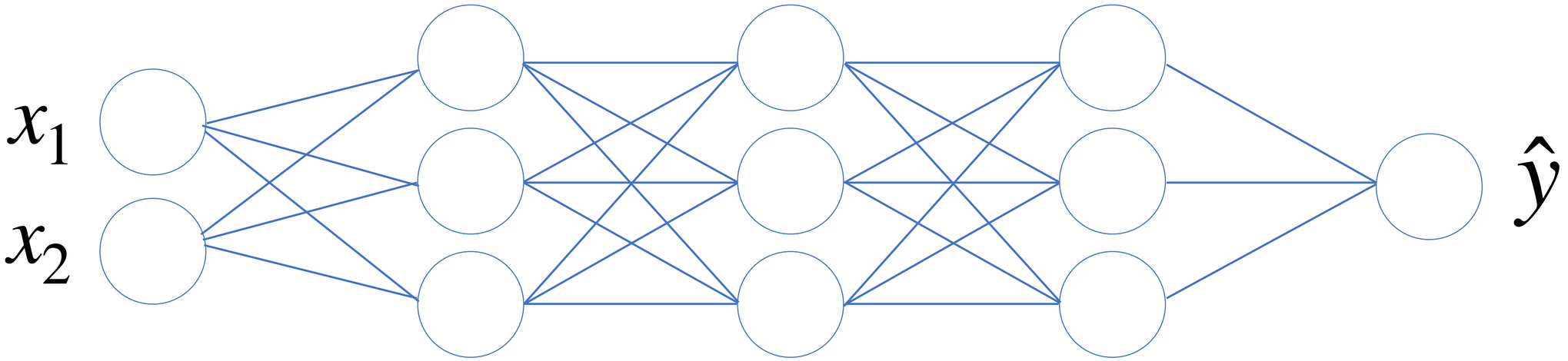


$$\mathbf{y} = \mathbf{F}[\mathbf{x}, \theta]$$

Recap

- Descenso del Gradiente
- Estocasticidad para Funciones de Pérdida no Convexas
- Aprendizaje Adaptativo
- Teorema de Aproximación Universal
- Redes Neuronales Profundas

Redes Neuronales: Profundas



$a^{[l]}$ = activación de la capa l

$W^{[l]}$ = pesos de la capa l

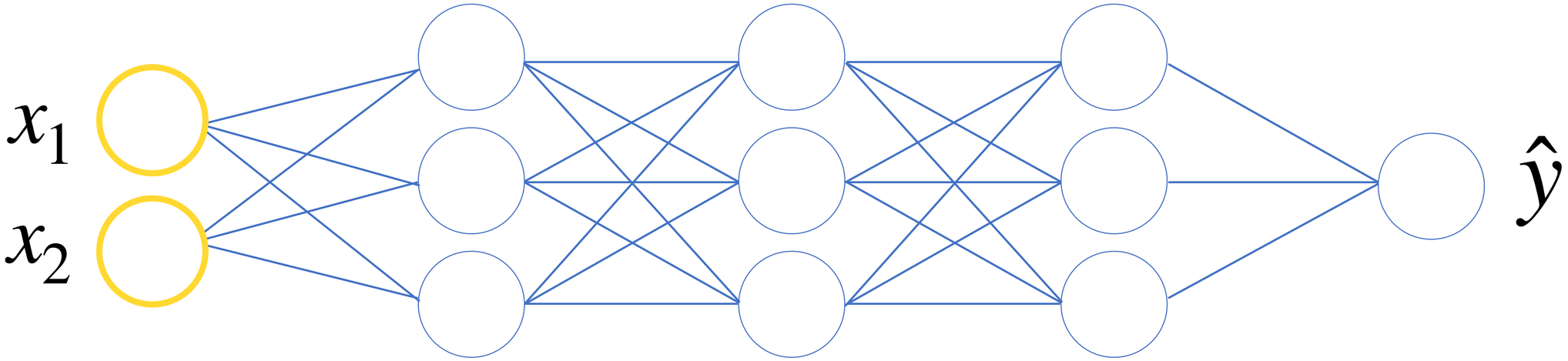
$b^{[l]}$ = sesgos (biases) de la capa l

σ = función de activación (por ejemplo: sigmoide, tanh, ReLU)

Redes Neuronales: Profundas

Forward Pass

$$a^{[l]} = \sigma\left(W^{[l]}a^{[l-1]} + b^{[l]}\right)$$



$a^{[l]}$ = activación de la capa l

$W^{[l]}$ = pesos de la capa l

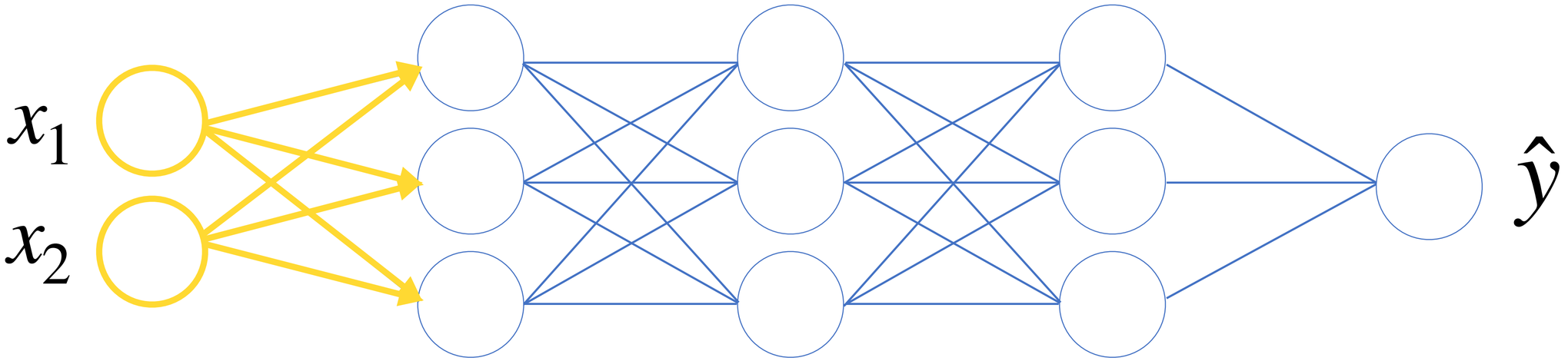
$b^{[l]}$ = sesgos (biases) de la capa l

σ = función de activación (por ejemplo: sigmoide, tanh, ReLU)

Redes Neuronales: Profundas

Forward Pass

$$a^{[l]} = \sigma\left(W^{[l]}a^{[l-1]} + b^{[l]}\right)$$



$a^{[l]}$ = activación de la capa l

$W^{[l]}$ = pesos de la capa l

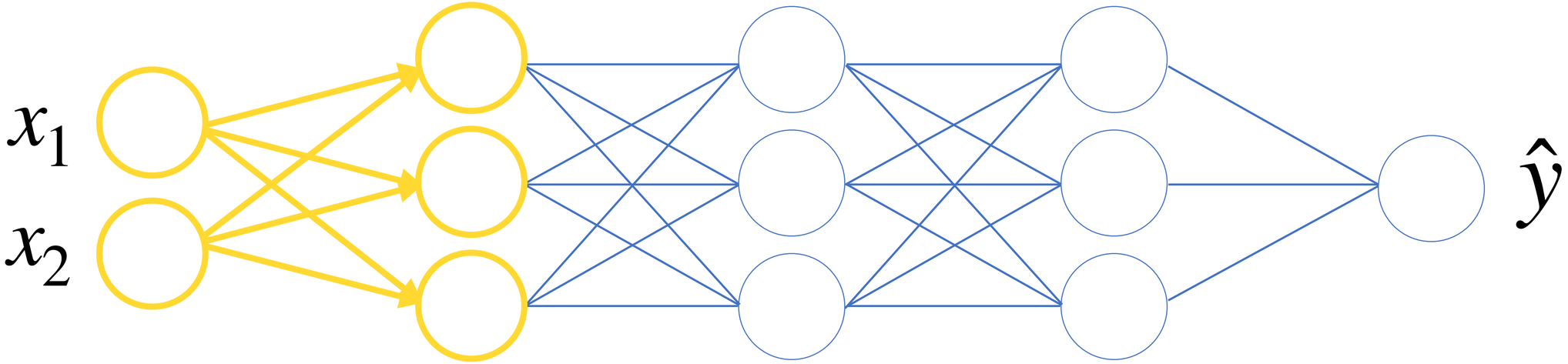
$b^{[l]}$ = sesgos (biases) de la capa l

σ = función de activación (por ejemplo: sigmoide, tanh, ReLU)

Redes Neuronales: Profundas

Forward Pass

$$a^{[l]} = \sigma\left(W^{[l]}a^{[l-1]} + b^{[l]}\right)$$



$a^{[l]}$ = activación de la capa l

$W^{[l]}$ = pesos de la capa l

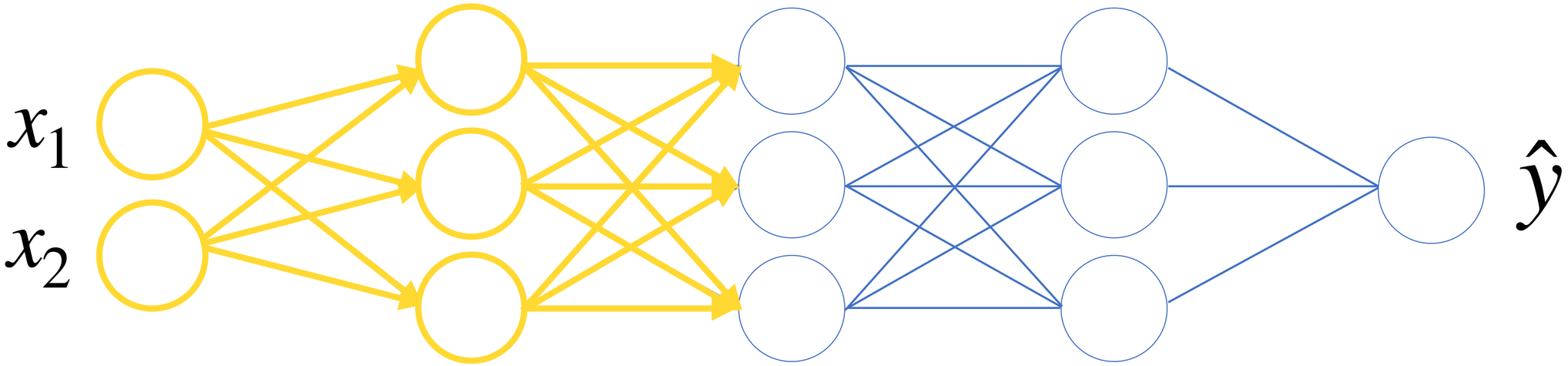
$b^{[l]}$ = sesgos (biases) de la capa l

σ = función de activación (por ejemplo: sigmoide, tanh, ReLU)

Redes Neuronales: Profundas

Forward Pass

$$a^{[l]} = \sigma\left(W^{[l]}a^{[l-1]} + b^{[l]}\right)$$



$a^{[l]}$ = activación de la capa l

$W^{[l]}$ = pesos de la capa l

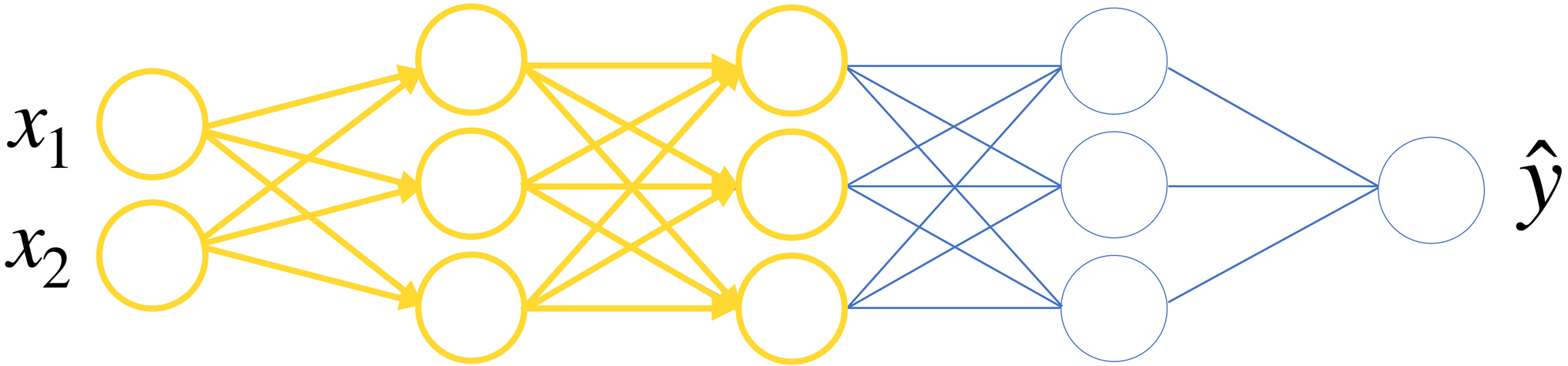
$b^{[l]}$ = sesgos (biases) de la capa l

σ = función de activación (por ejemplo: sigmoide, tanh, ReLU)

Redes Neuronales: Profundas

Forward Pass

$$a^{[l]} = \sigma\left(W^{[l]}a^{[l-1]} + b^{[l]}\right)$$



$a^{[l]}$ = activación de la capa l

$W^{[l]}$ = pesos de la capa l

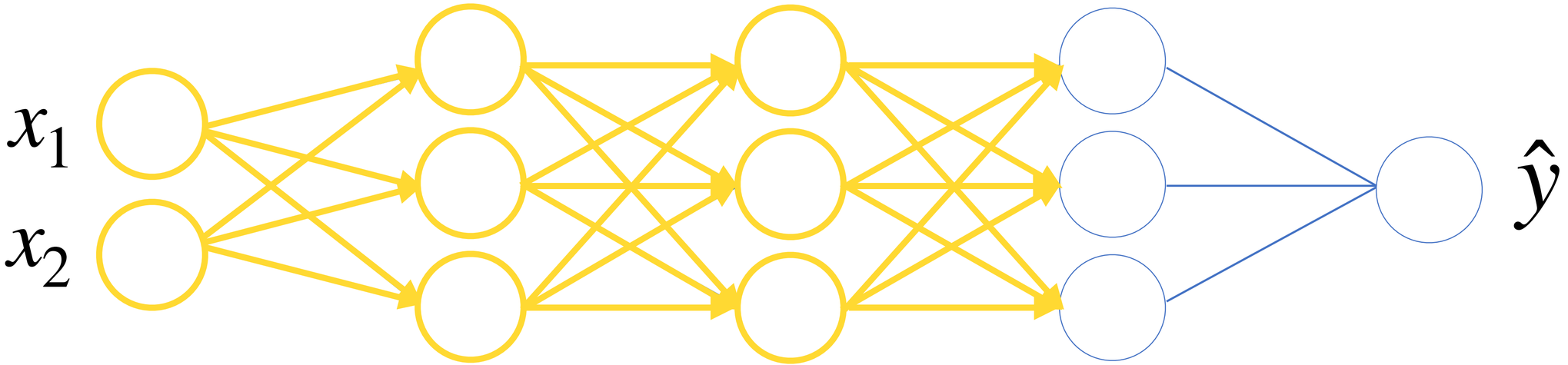
$b^{[l]}$ = sesgos (biases) de la capa l

σ = función de activación (por ejemplo: sigmoide, tanh, ReLU)

Redes Neuronales: Profundas

Forward Pass

$$a^{[l]} = \sigma\left(W^{[l]}a^{[l-1]} + b^{[l]}\right)$$



$a^{[l]}$ = activación de la capa l

$W^{[l]}$ = pesos de la capa l

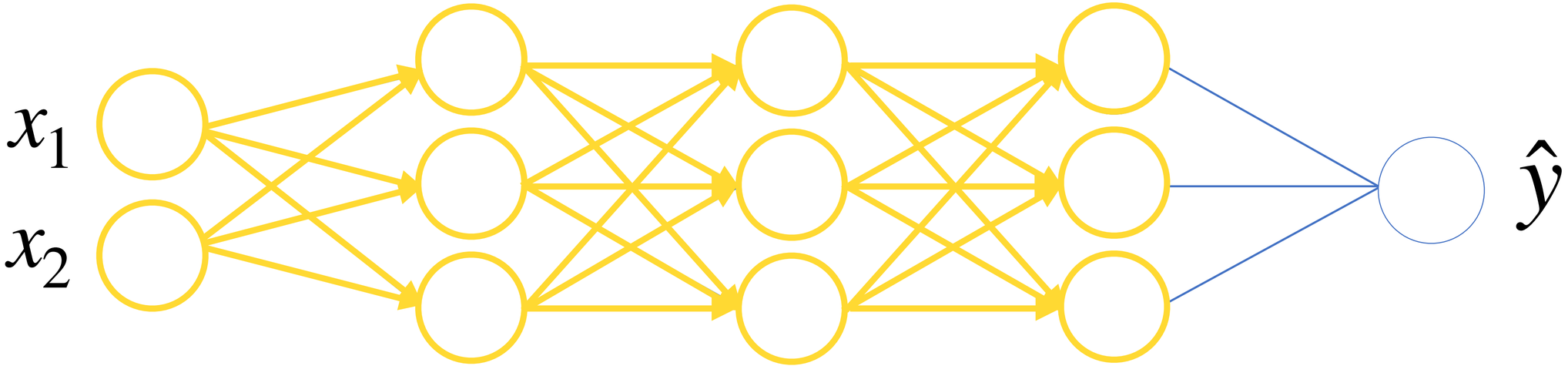
$b^{[l]}$ = sesgos (biases) de la capa l

σ = función de activación (por ejemplo: sigmoide, tanh, ReLU)

Redes Neuronales: Profundas

Forward Pass

$$a^{[l]} = \sigma\left(W^{[l]}a^{[l-1]} + b^{[l]}\right)$$



$a^{[l]}$ = activación de la capa l

$W^{[l]}$ = pesos de la capa l

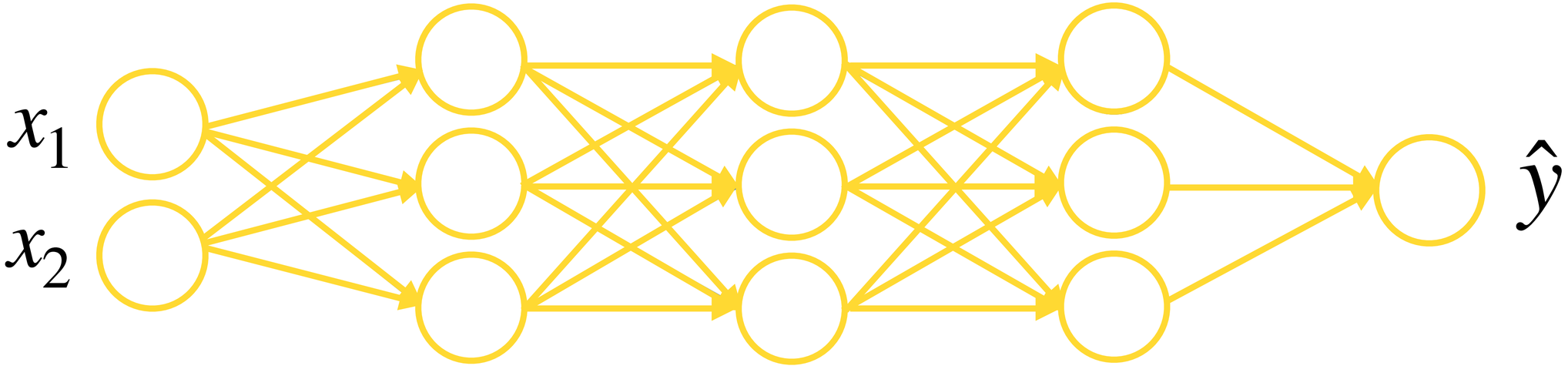
$b^{[l]}$ = sesgos (biases) de la capa l

σ = función de activación (por ejemplo: sigmoide, tanh, ReLU)

Redes Neuronales: Profundas

Forward Pass

$$a^{[l]} = \sigma\left(W^{[l]}a^{[l-1]} + b^{[l]}\right)$$



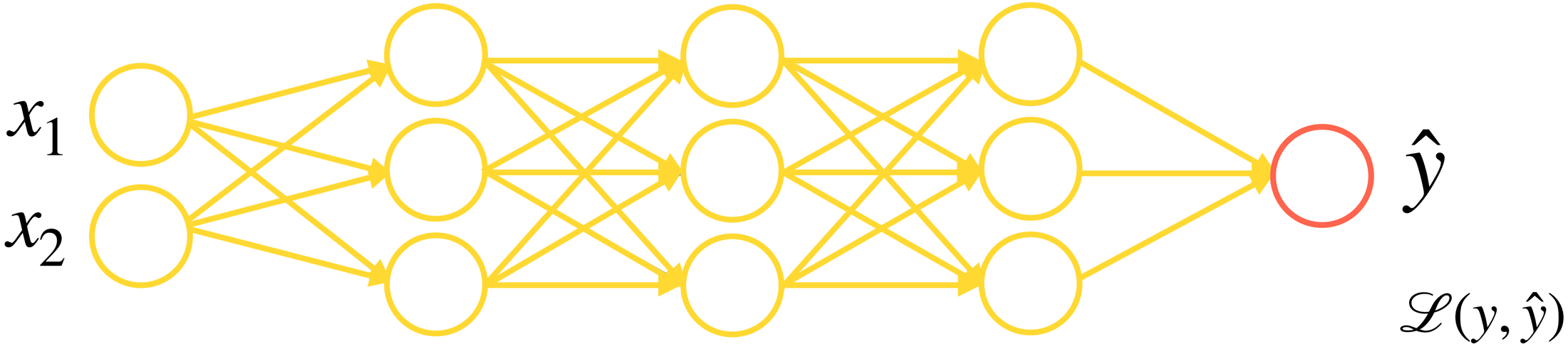
$a^{[l]}$ = activación de la capa l

$W^{[l]}$ = pesos de la capa l

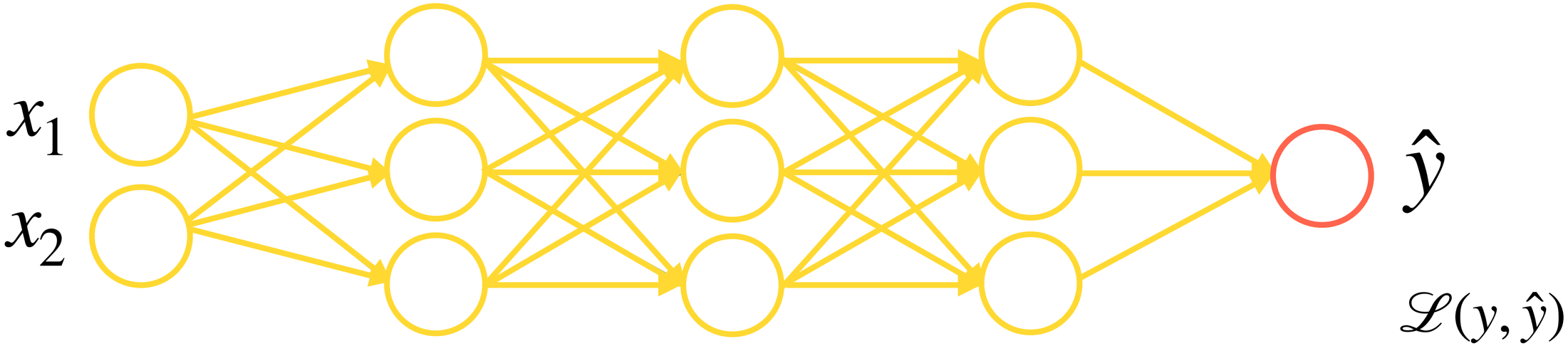
$b^{[l]}$ = sesgos (biases) de la capa l

σ = función de activación (por ejemplo: sigmoide, tanh, ReLU)

Redes Neuronales: Profundas

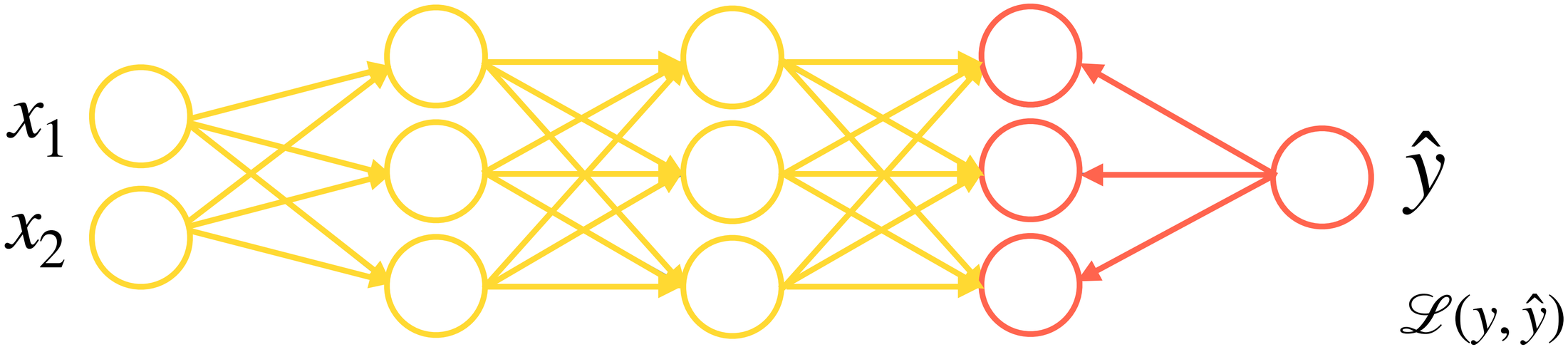


Redes Neuronales: Profundas



$$\nabla_\theta \mathscr{L}(\theta) = \frac{\partial \mathscr{L}(\theta)}{\partial \theta} \qquad \theta_{new} = \theta_{old} - \eta \nabla_\theta \mathscr{L}(\theta_{old})$$

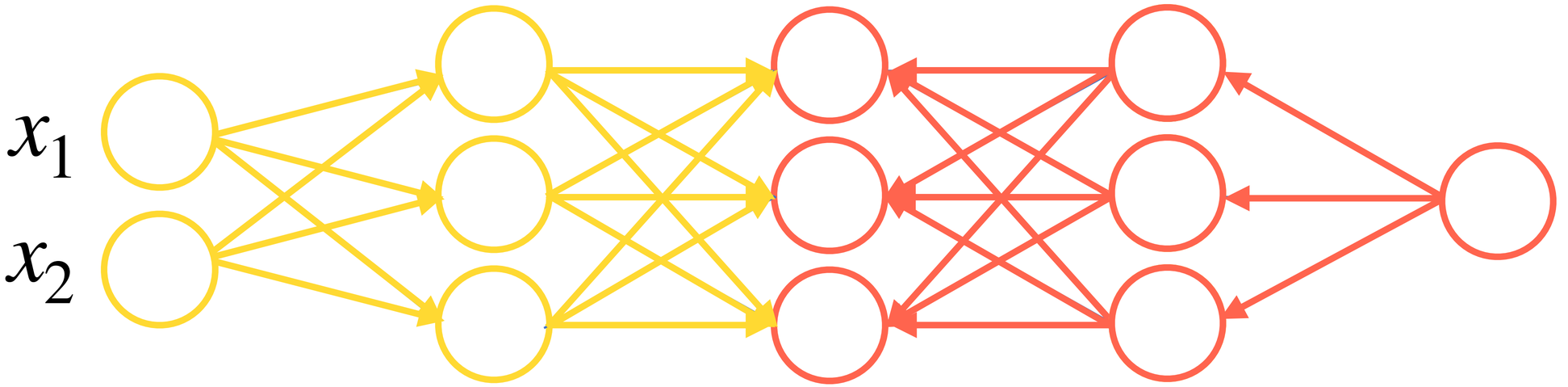
Redes Neuronales: Profundas



Back Propagate

$$\nabla_\theta \mathscr{L}(\theta) = \frac{\partial \mathscr{L}(\theta)}{\partial \theta} \qquad \theta_{new} = \theta_{old} - \eta \nabla_\theta \mathscr{L}(\theta_{old})$$

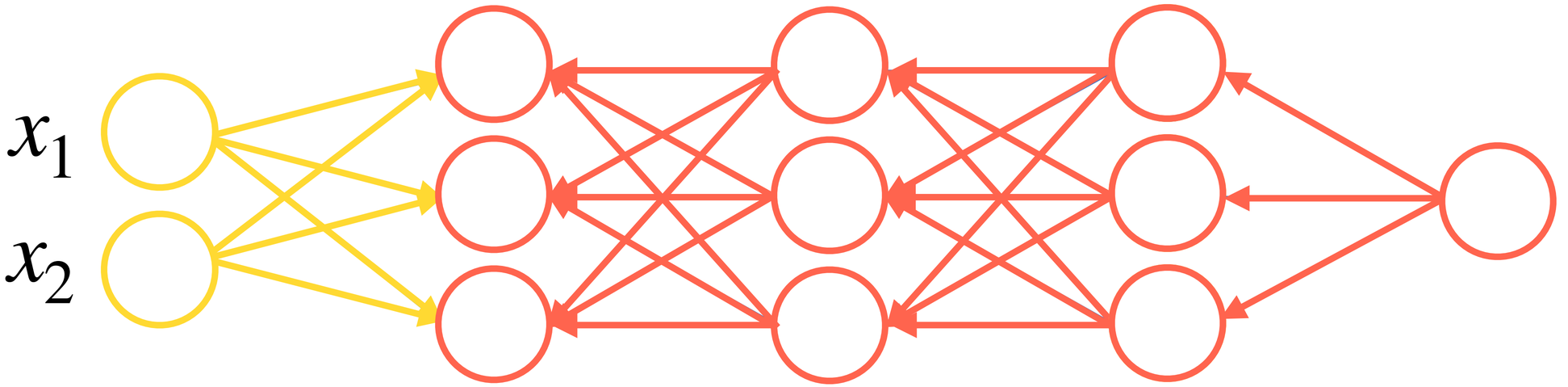
Redes Neuronales: Profundas



Back Propagate

$$\nabla_\theta \mathscr{L}(\theta) = \frac{\partial \mathscr{L}(\theta)}{\partial \theta} \qquad \theta_{new} = \theta_{old} - \eta \nabla_\theta \mathscr{L}(\theta_{old})$$

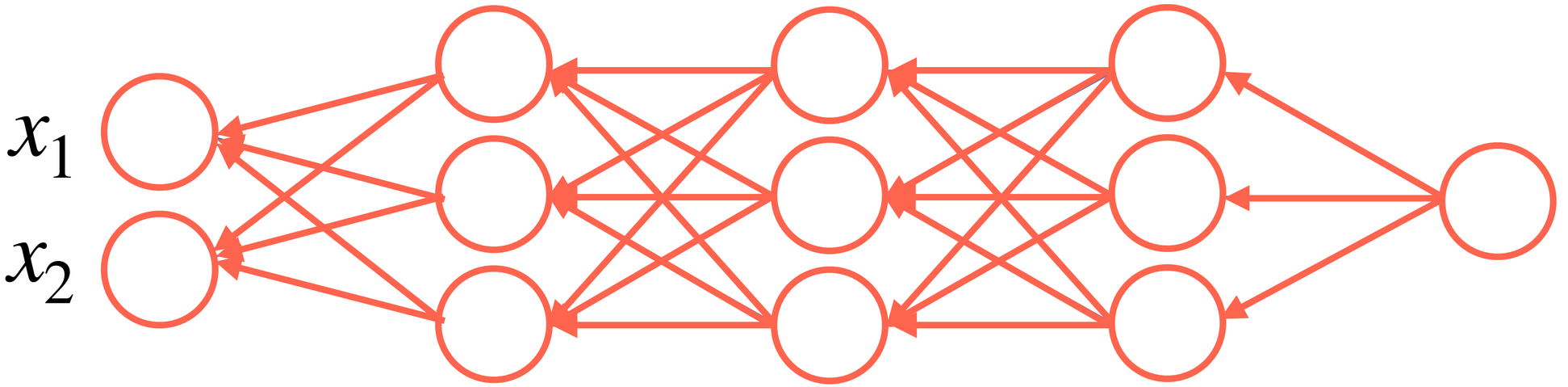
Redes Neuronales: Profundas



Back Propagate

$$\nabla_\theta \mathscr{L}(\theta) = \frac{\partial \mathscr{L}(\theta)}{\partial \theta} \qquad \theta_{new} = \theta_{old} - \eta \nabla_\theta \mathscr{L}(\theta_{old})$$

Redes Neuronales: Profundas

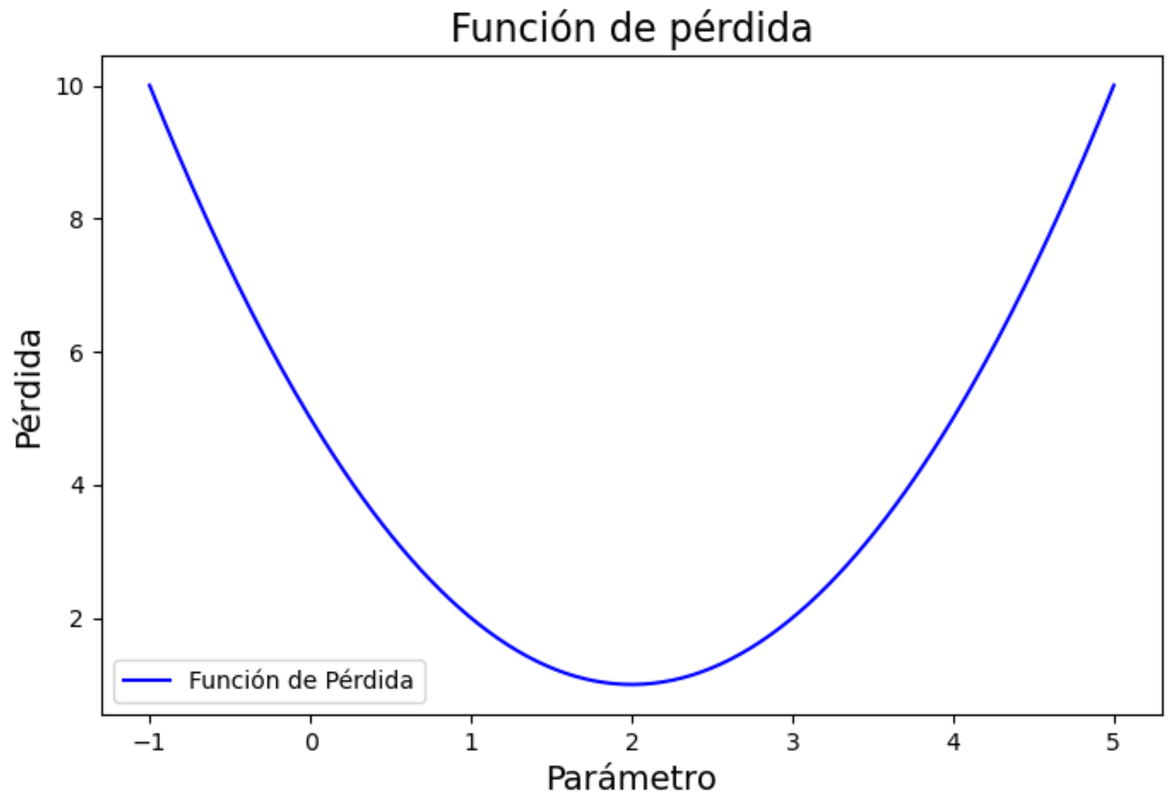


Back Propagate

$$\nabla_\theta \mathscr{L}(\theta) = \frac{\partial \mathscr{L}(\theta)}{\partial \theta} \qquad \theta_{new} = \theta_{old} - \eta \nabla_\theta \mathscr{L}(\theta_{old})$$

Redes Neuronales: Descenso del Gradiente

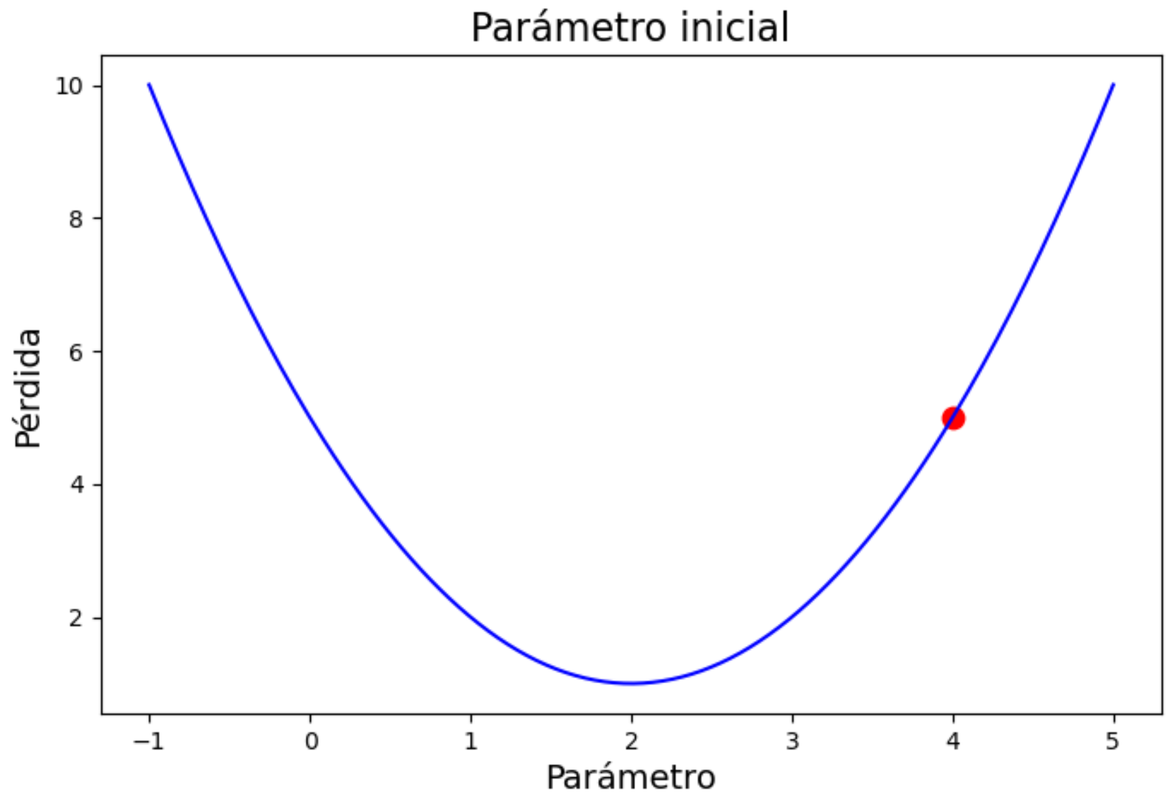
$$\mathscr{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}(x_i;\theta))^2$$



Redes Neuronales: Descenso del Gradiente

$$\mathscr{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}(x_i;\theta))^2$$

$$\mathscr{L}(\theta_{ini}) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}(x_i;\theta_{ini})\right)^2$$

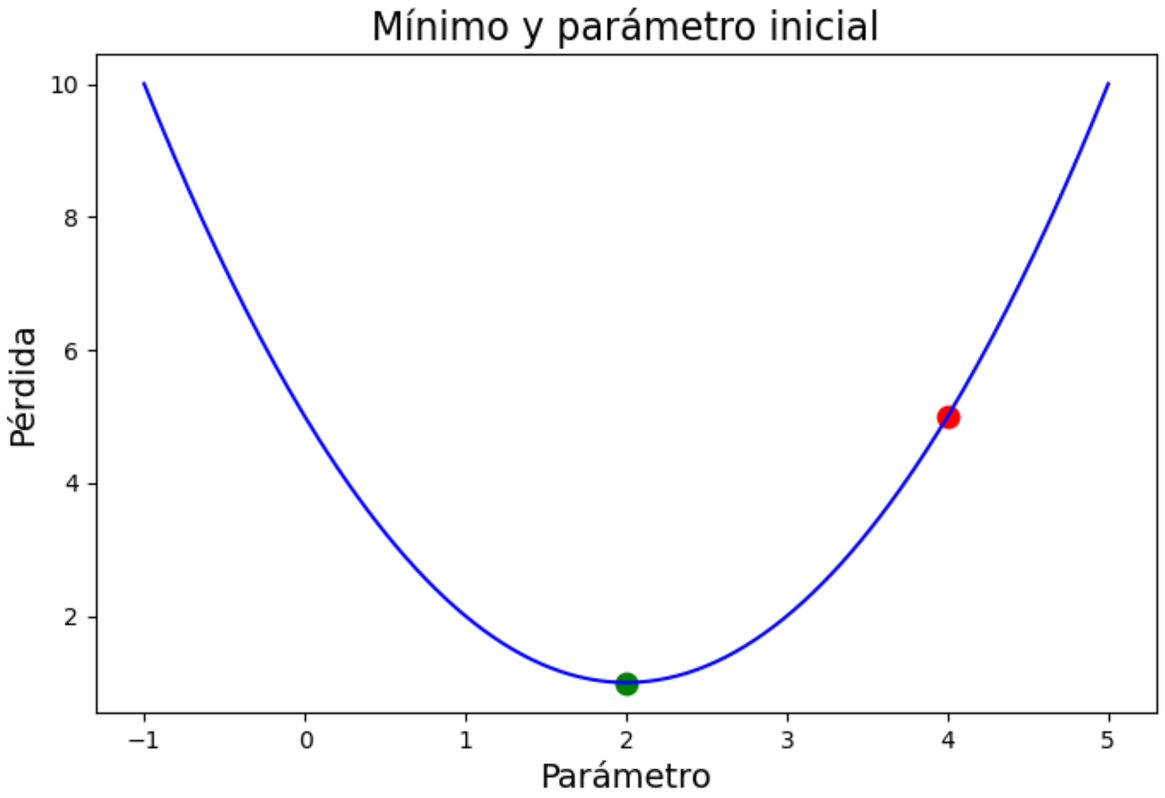


Redes Neuronales: Descenso del Gradiente

$$\mathscr{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}(x_i;\theta))^2$$

$$\mathscr{L}(\theta_{ini}) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}(x_i;\theta_{ini})\right)^2$$

$$\mathscr{L}(\theta_{min}) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}(x_i;\theta_{min})\right)^2$$



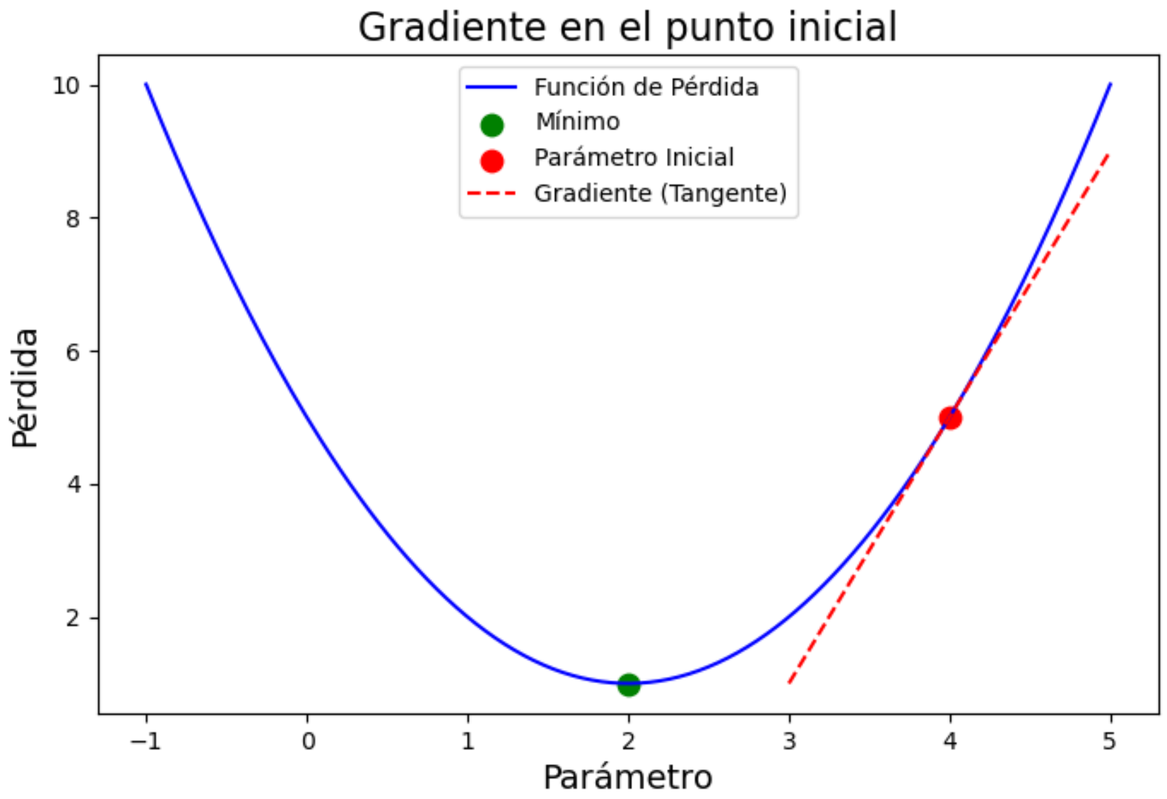
Redes Neuronales: Descenso del Gradiente

$$\mathscr{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}(x_i;\theta))^2$$

$$\mathscr{L}(\theta_{ini}) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}(x_i;\theta_{ini})\right)^2$$

$$\mathscr{L}(\theta_{min}) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}(x_i;\theta_{min})\right)^2$$

$$\nabla_\theta \mathscr{L}(\theta) = \frac{\partial \mathscr{L}(\theta)}{\partial \theta}$$



Redes Neuronales: Descenso del Gradiente

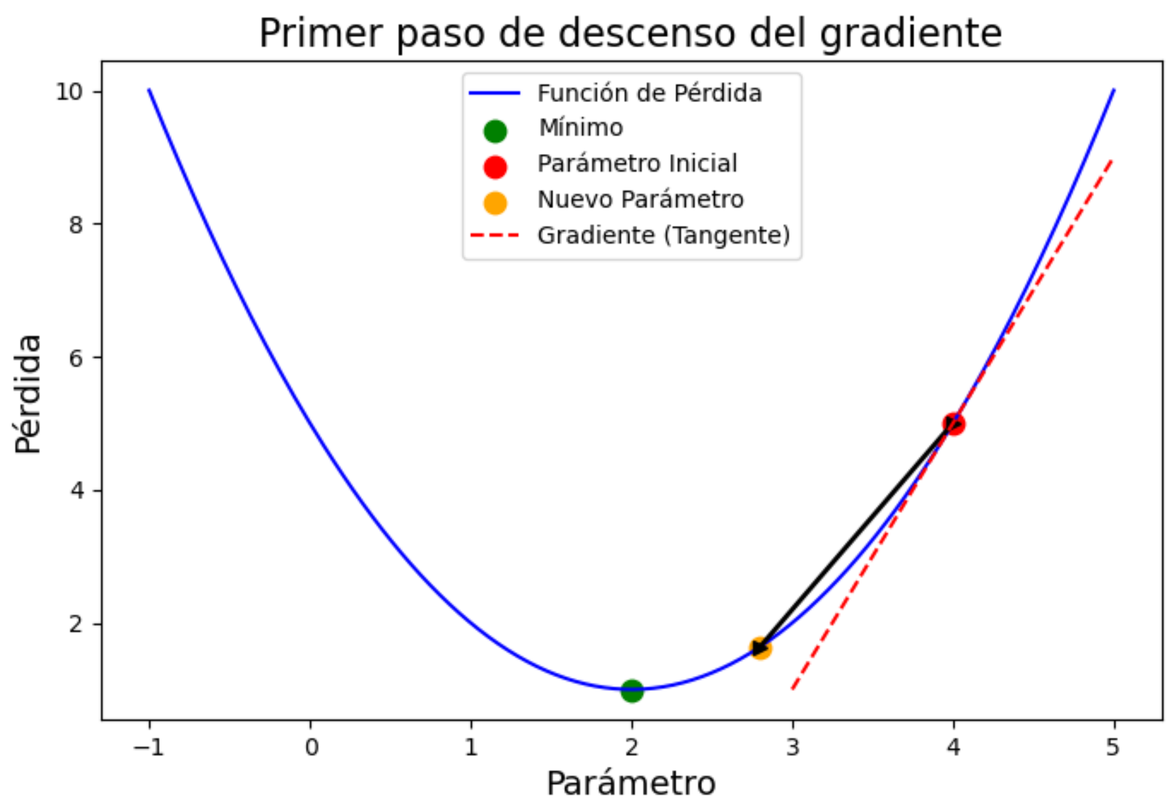
$$\mathscr{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}(x_i;\theta))^2$$

$$\mathscr{L}(\theta_{ini}) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}(x_i;\theta_{ini})\right)^2$$

$$\mathscr{L}(\theta_{min}) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}(x_i;\theta_{min})\right)^2$$

$$\nabla_\theta \mathscr{L}(\theta) = \frac{\partial \mathscr{L}(\theta)}{\partial \theta}$$

$$\theta_{new} = \theta_{old} - \eta \nabla_\theta \mathscr{L}(\theta_{old})$$



Redes Neuronales: Descenso del Gradiente

$$\mathscr{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}(x_i;\theta))^2$$

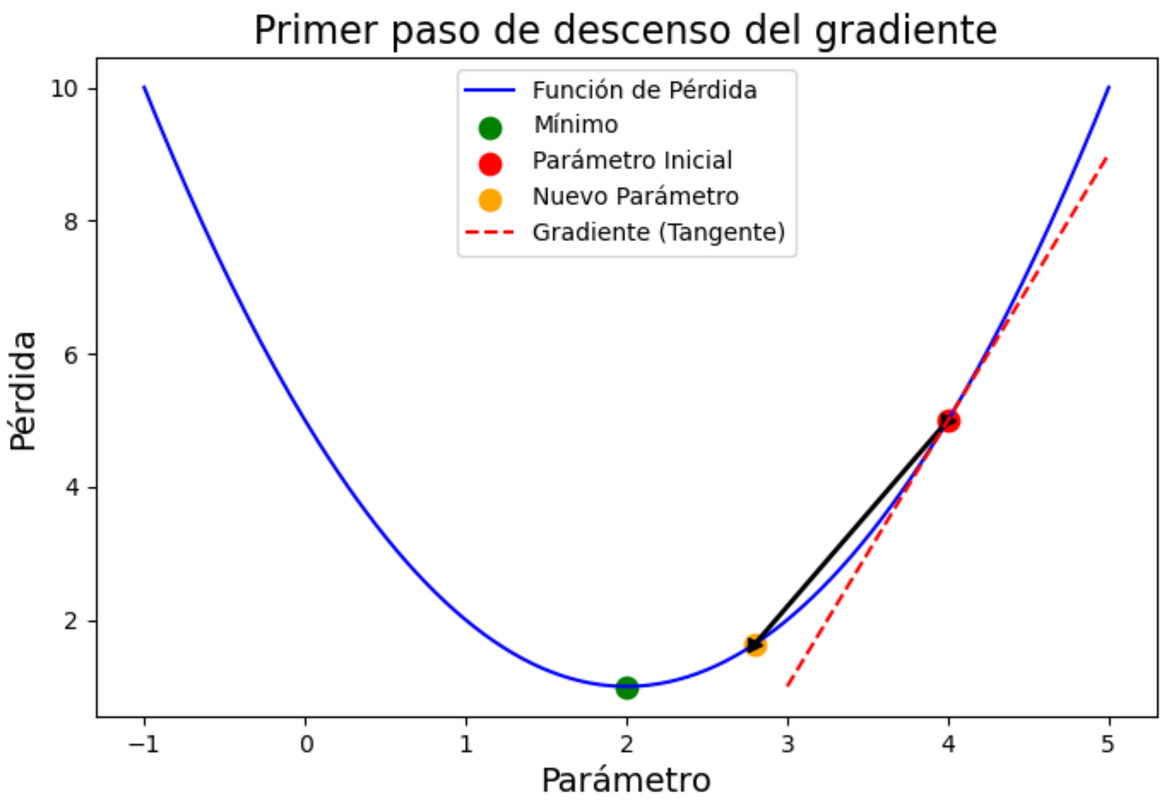
$$\mathscr{L}(\theta_{ini}) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}(x_i;\theta_{ini})\right)^2$$

$$\mathscr{L}(\theta_{min}) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}(x_i;\theta_{min})\right)^2$$

$$\nabla_\theta \mathscr{L}(\theta) = \frac{\partial \mathscr{L}(\theta)}{\partial \theta}$$

$$\theta_{new} = \theta_{old} - \eta \nabla_\theta \mathscr{L}(\theta_{old})$$

$\theta_{ini} = = \theta_{new}$



Redes Neuronales: Descenso del Gradiente

$$\mathscr{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}(x_i;\theta))^2$$

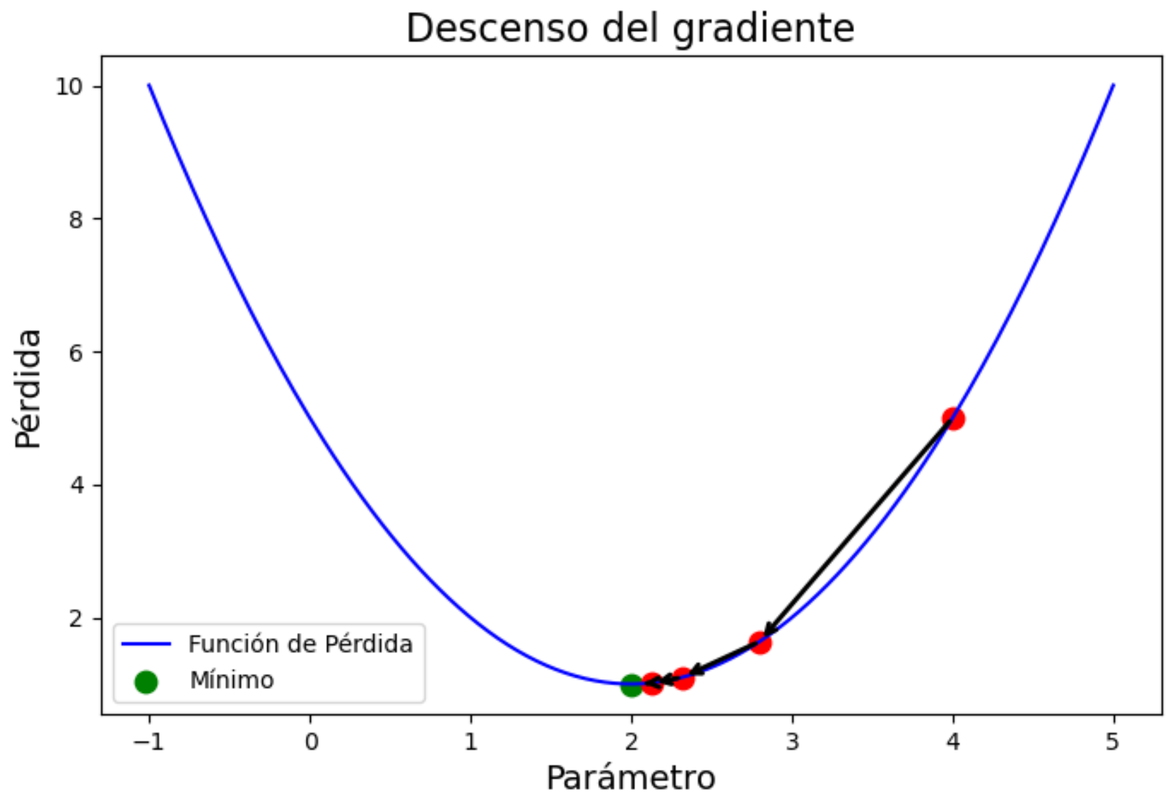
$$\mathscr{L}(\theta_{ini}) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}(x_i;\theta_{ini})\right)^2$$

$$\mathscr{L}(\theta_{min}) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}(x_i;\theta_{min})\right)^2$$

$$\nabla_\theta \mathscr{L}(\theta) = \frac{\partial \mathscr{L}(\theta)}{\partial \theta}$$

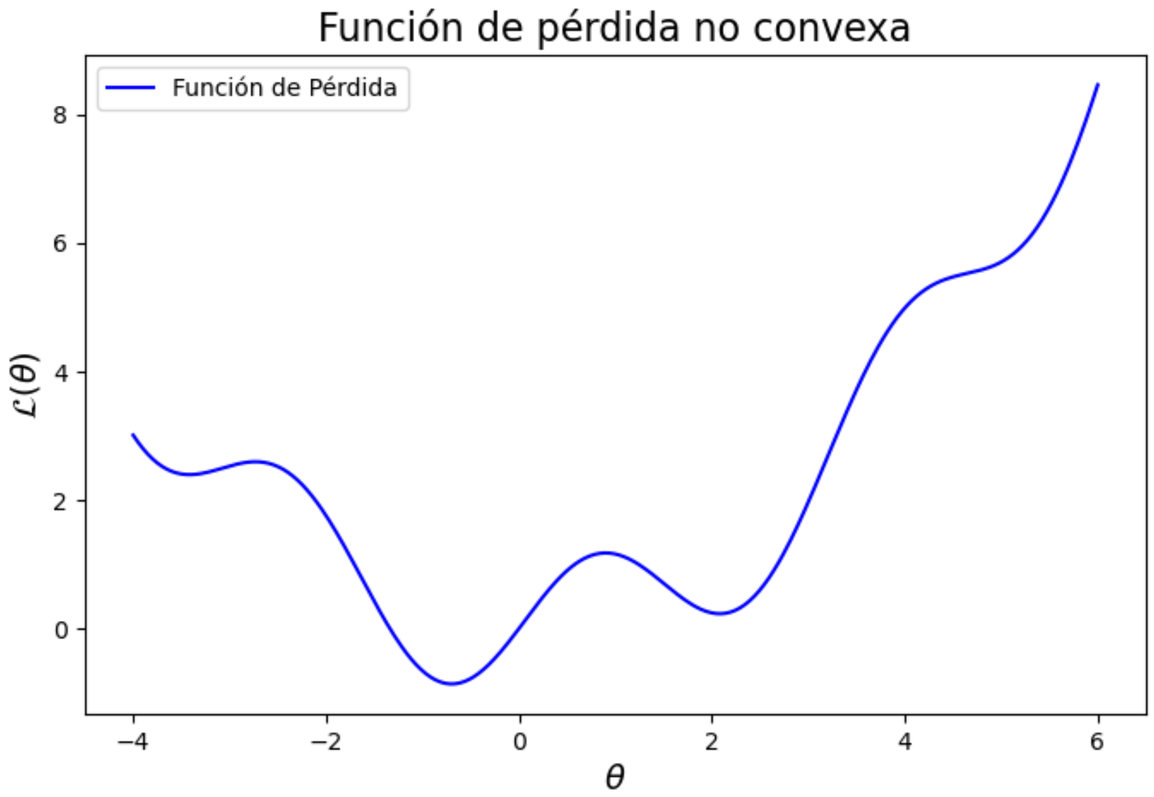
$$\theta_{new} = \theta_{old} - \eta \nabla_\theta \mathscr{L}(\theta_{old})$$

$\theta_{ini} = = \theta_{new}$



Redes Neuronales: Descenso del Gradiente

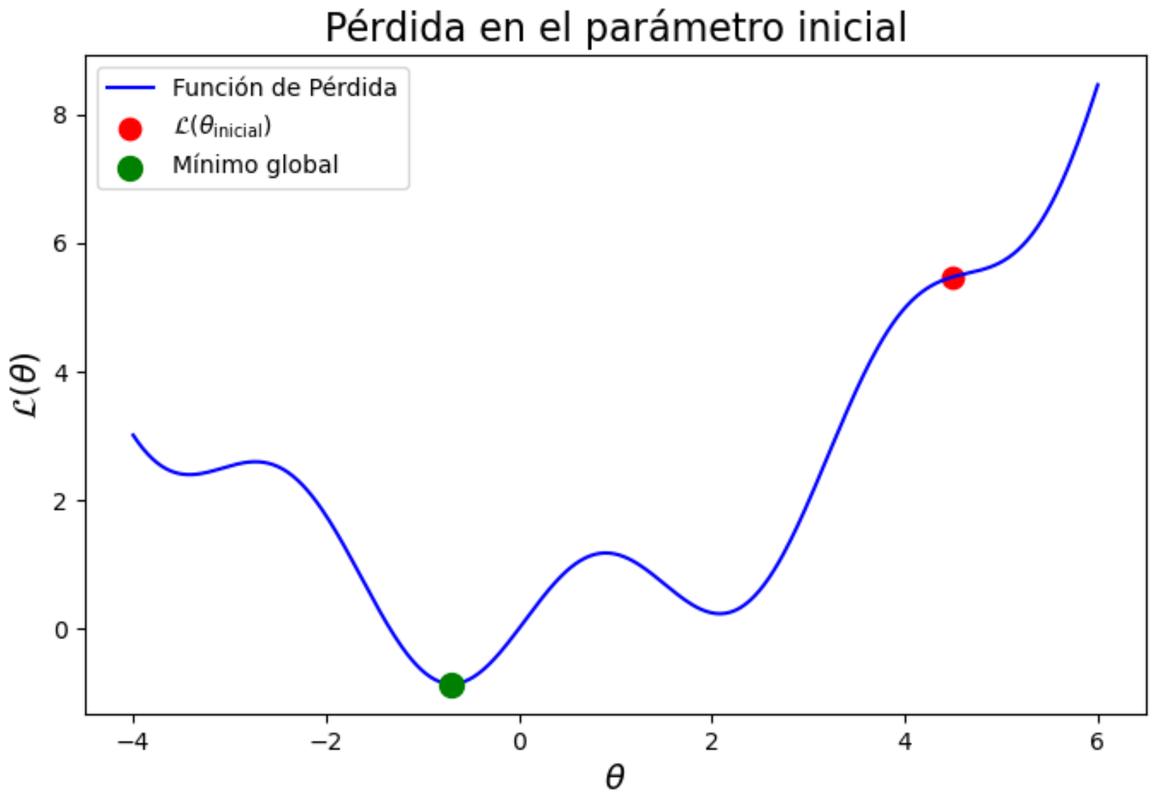
$$\mathscr{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}(x_i;\theta))^2$$



Redes Neuronales: Descenso del Gradiente

$$\mathscr{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}(x_i;\theta))^2$$

$$\mathscr{L}(\theta_{ini}) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}(x_i;\theta_{ini})\right)^2$$



Redes Neuronales: Función de Pérdida no Convexa

$$\mathscr{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}(x_i;\theta))^2$$

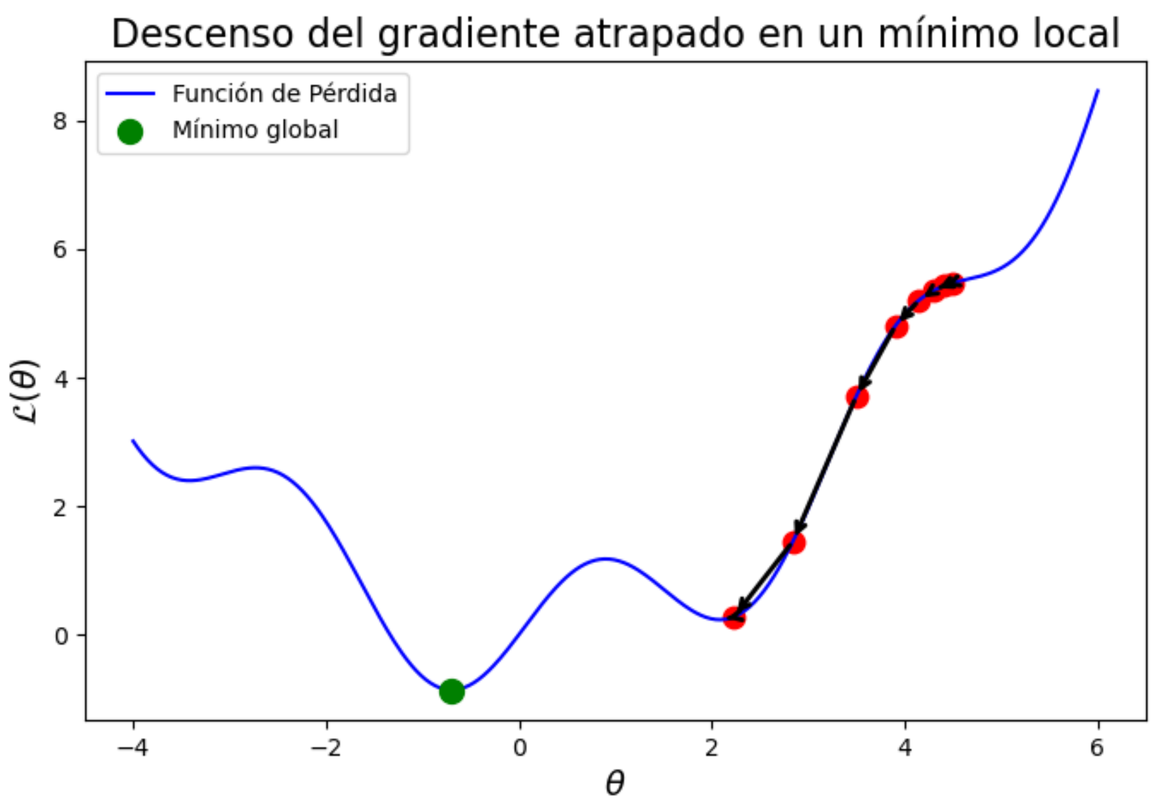
$$\mathscr{L}(\theta_{ini}) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}(x_i;\theta_{ini})\right)^2$$

$$\mathscr{L}(\theta_{min}) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}(x_i;\theta_{min})\right)^2$$

$$\nabla_\theta \mathscr{L}(\theta) = \frac{\partial \mathscr{L}(\theta)}{\partial \theta}$$

$$\theta_{new} = \theta_{old} - \eta \nabla_\theta \mathscr{L}(\theta_{old})$$

$\theta_{ini} = = \theta_{new}$



Redes Neuronales: Estocasticidad

$$\mathscr{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}(x_i;\theta))^2$$

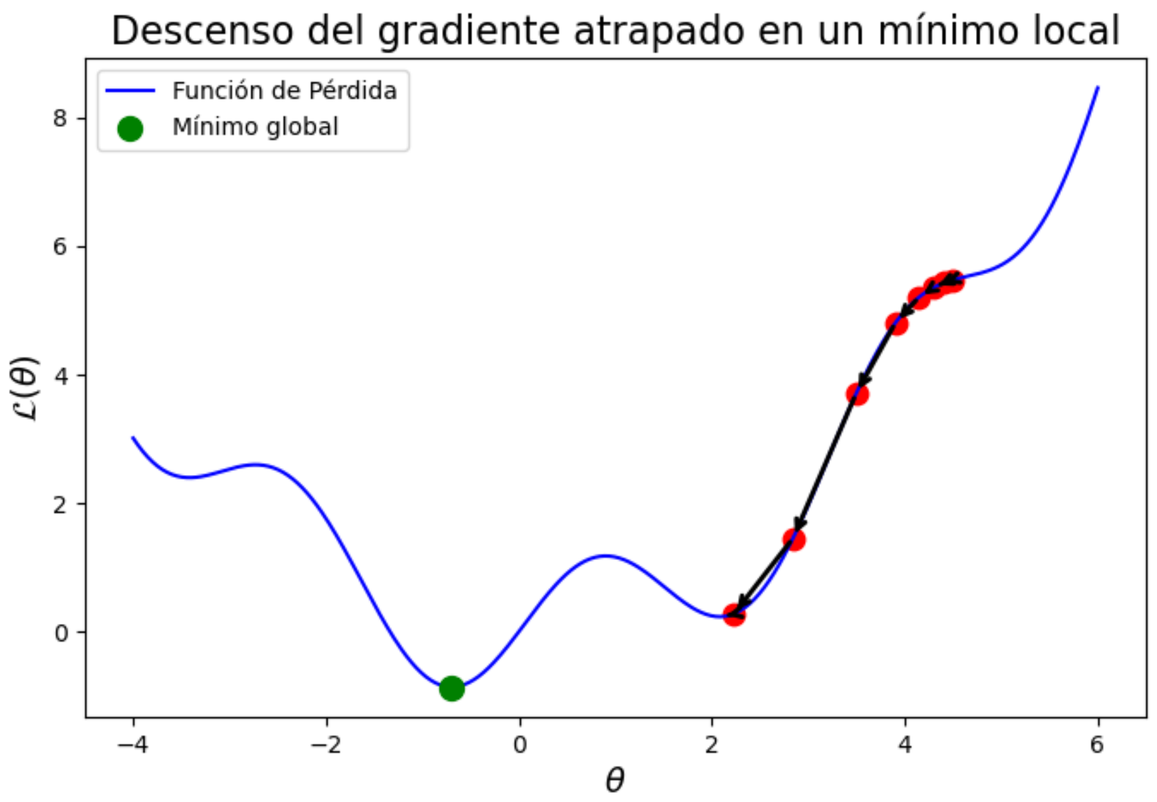
$$\mathscr{L}(\theta_{ini}) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}(x_i;\theta_{ini})\right)^2$$

$$\mathscr{L}(\theta_{min}) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}(x_i;\theta_{min})\right)^2$$

$$\nabla_\theta \mathscr{L}(\theta) = \frac{\partial \mathscr{L}(\theta)}{\partial \theta}$$

$$\theta_{new} = \theta_{old} - \eta \nabla_\theta \mathscr{L}(\theta_{old})$$

$\theta_{ini} = = \theta_{new}$



Redes Neuronales: Estocasticidad

$$\mathscr{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}(x_i;\theta))^2$$

$$\mathscr{L}(\theta_{ini}) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}(x_i;\theta_{ini})\right)^2$$

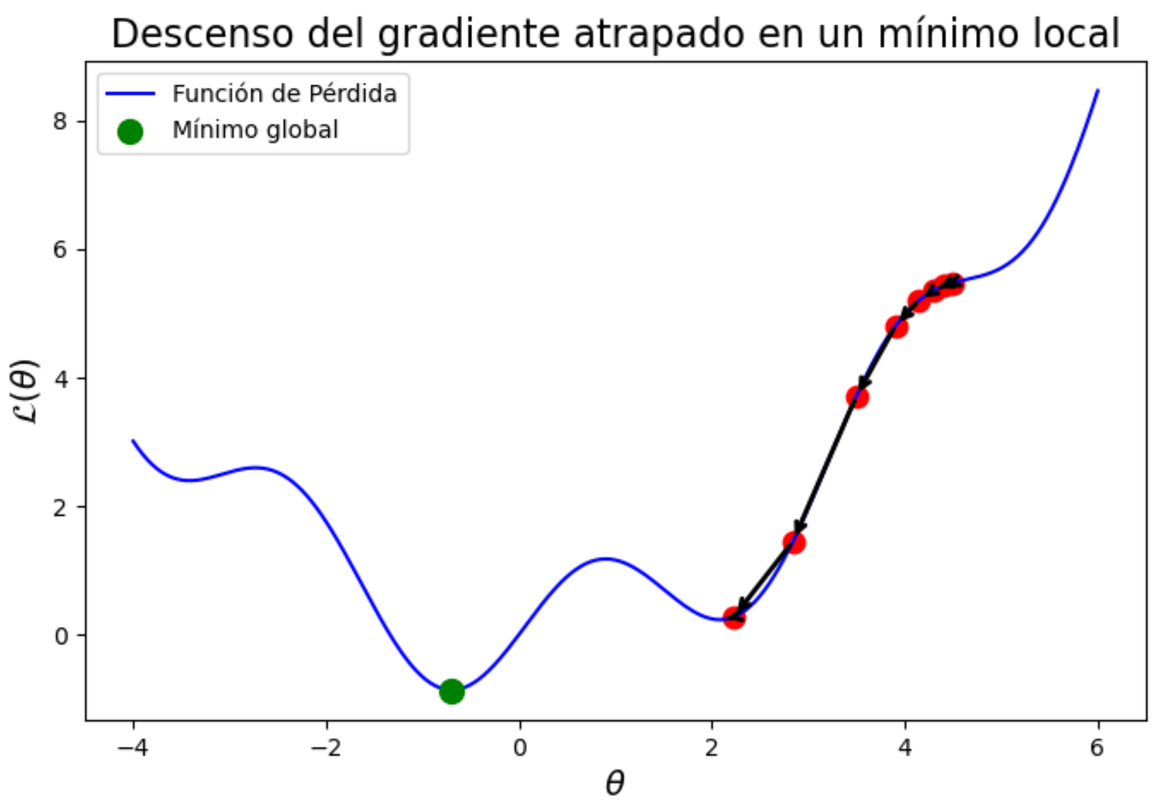
$$\mathscr{L}(\theta_{min}) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}(x_i;\theta_{min})\right)^2$$

$\theta_{ini} = = \theta_{new}$

$$\nabla_\theta \mathscr{L}(\theta) = \frac{\partial \mathscr{L}(\theta)}{\partial \theta}$$

$$\theta_{new} = \theta_{old} - \eta \nabla_\theta \mathscr{L}(\theta_{old})$$

$$\theta_{new} = \theta_{old} - \eta \nabla_\theta \mathscr{L}(\theta_{old}) + \mathcal{N}(0,\sigma^2)$$



Redes Neuronales: Estocasticidad

$$\mathscr{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}(x_i;\theta))^2$$

$$\mathscr{L}(\theta_{ini}) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}(x_i;\theta_{ini})\right)^2$$

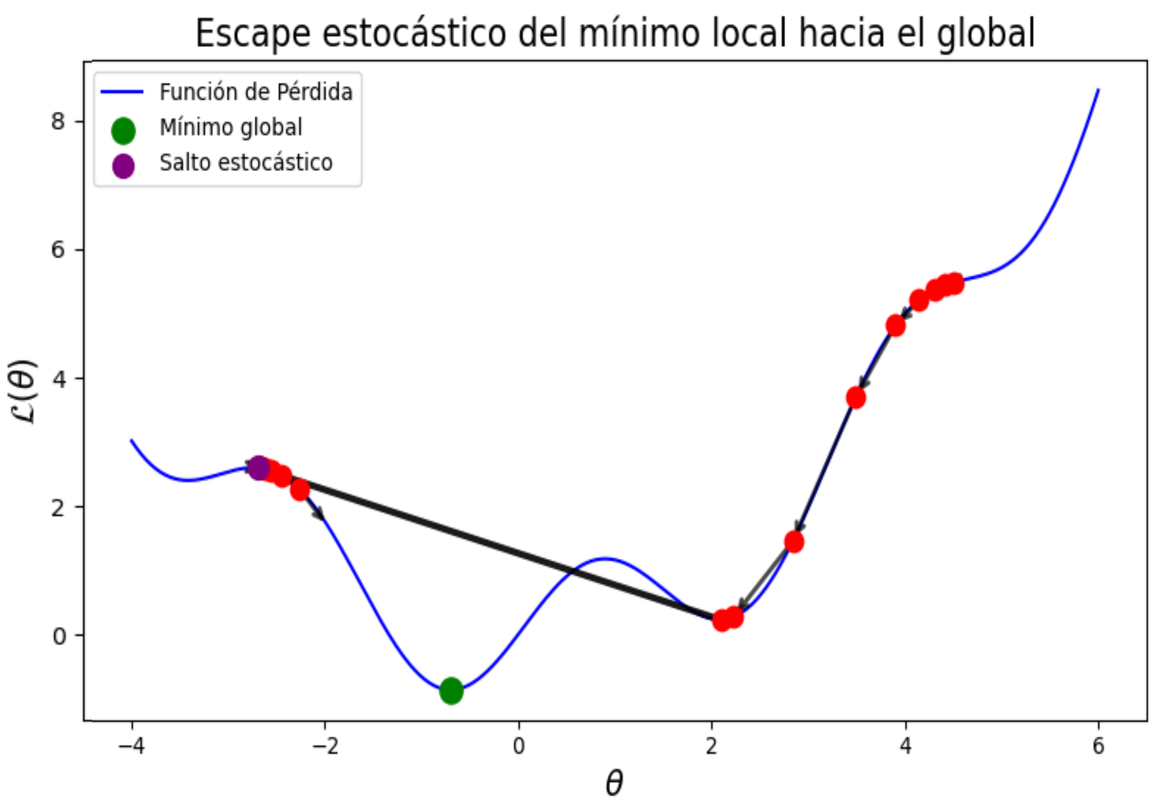
$$\mathscr{L}(\theta_{min}) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}(x_i;\theta_{min})\right)^2$$

$\theta_{ini} = = \theta_{new}$

$$\nabla_\theta \mathscr{L}(\theta) = \frac{\partial \mathscr{L}(\theta)}{\partial \theta}$$

$$\theta_{new} = \theta_{old} - \eta \nabla_\theta \mathscr{L}(\theta_{old})$$

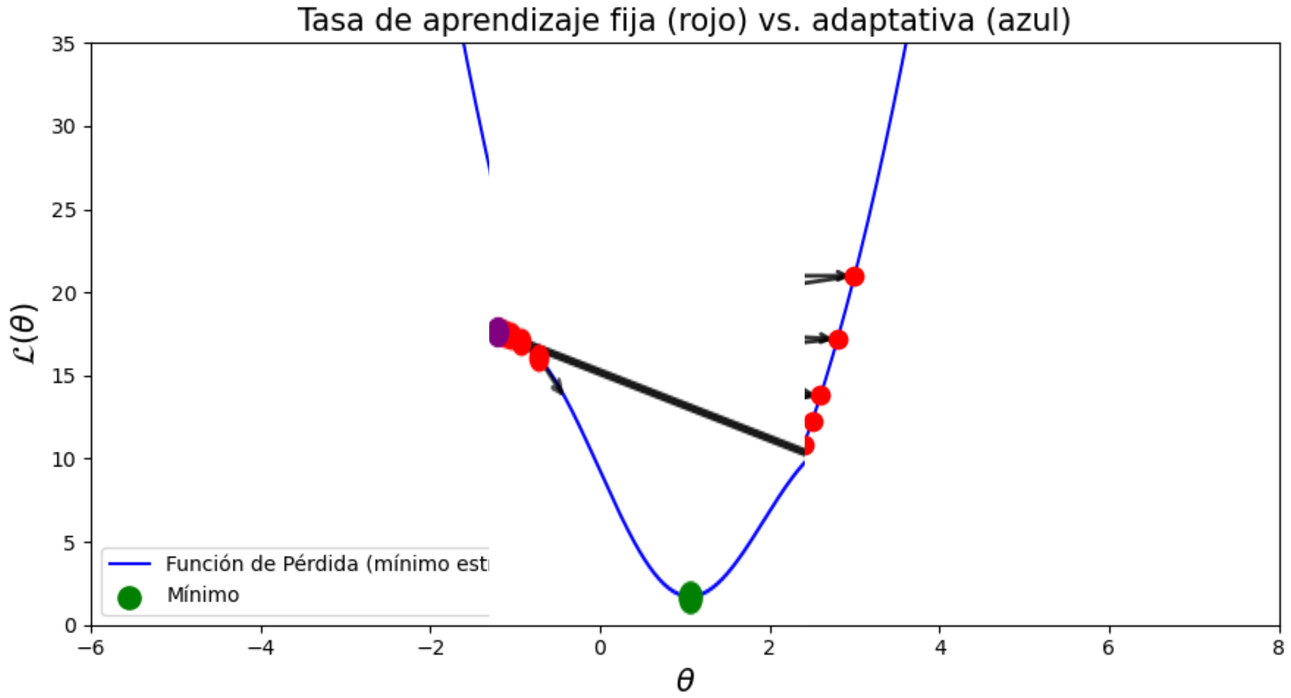
$$\theta_{new} = \theta_{old} - \eta \nabla_\theta \mathscr{L}(\theta_{old}) + \mathcal{N}(0,\sigma^2)$$



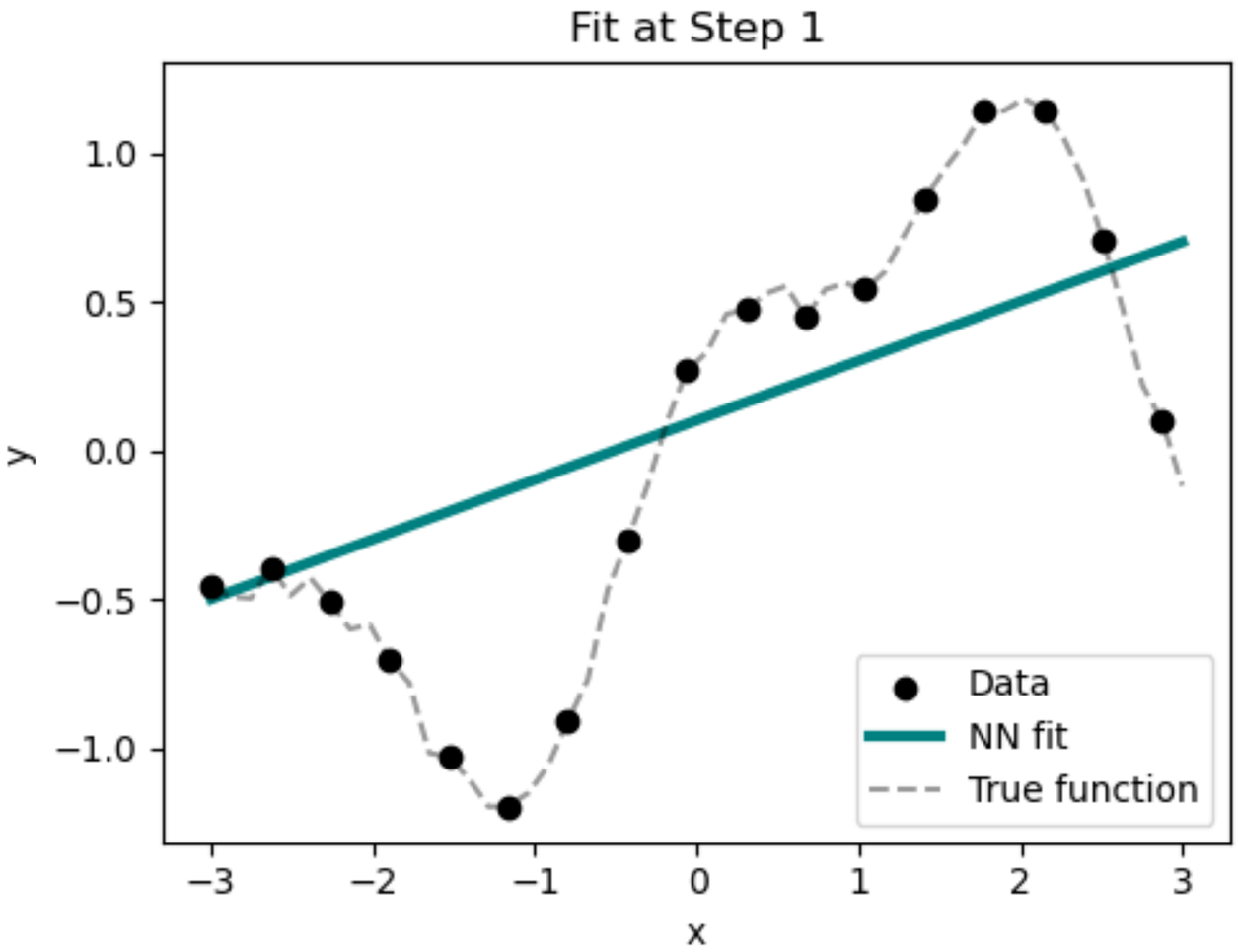
Redes Neuronales: Aprendizaje Adaptativo

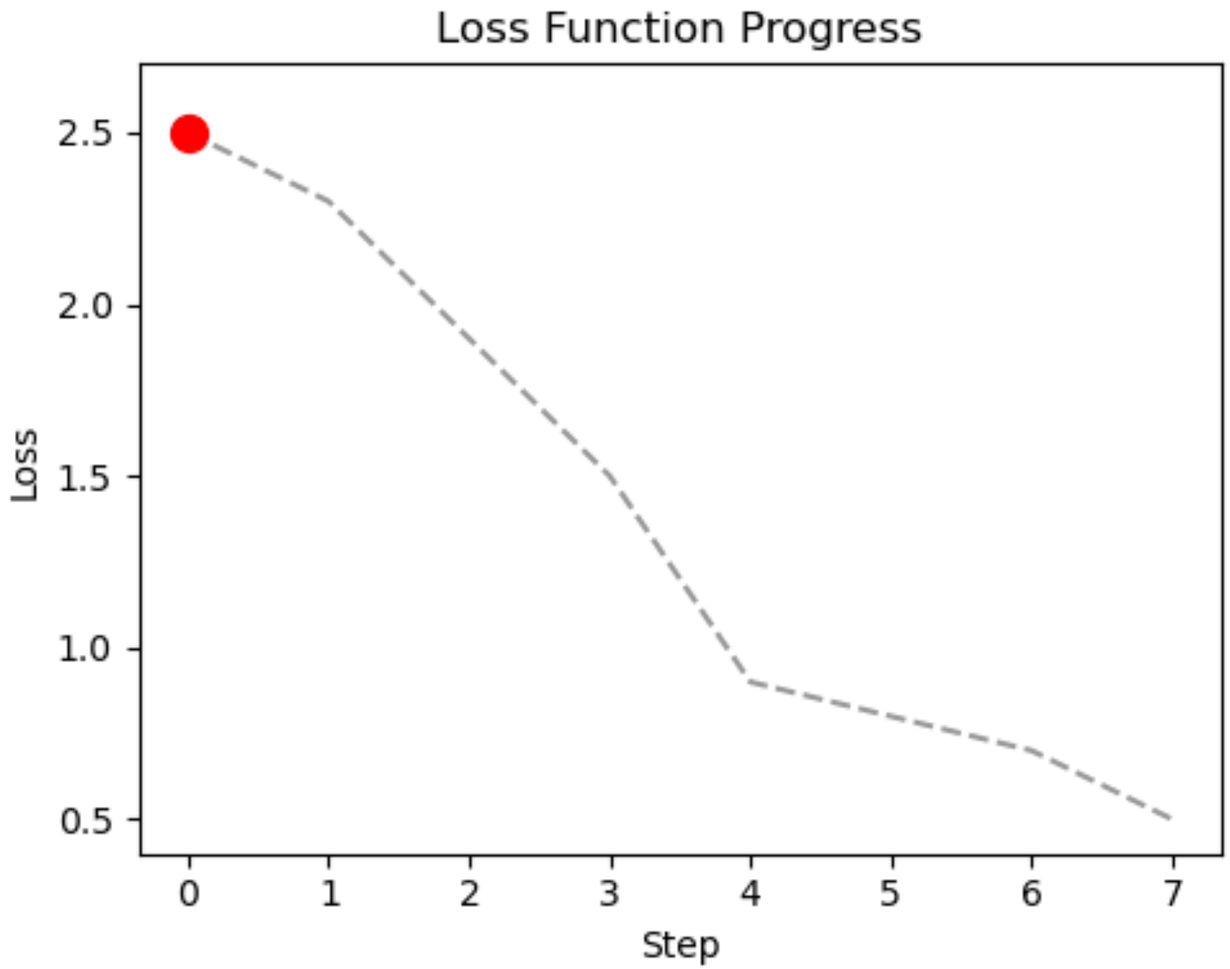
$$\theta_{t+1} = \theta_t - \eta \nabla_\theta \mathcal{L}(\theta_t)$$

η : tasa de aprendizaje fija
(puede causar oscilaciones si es muy grande)

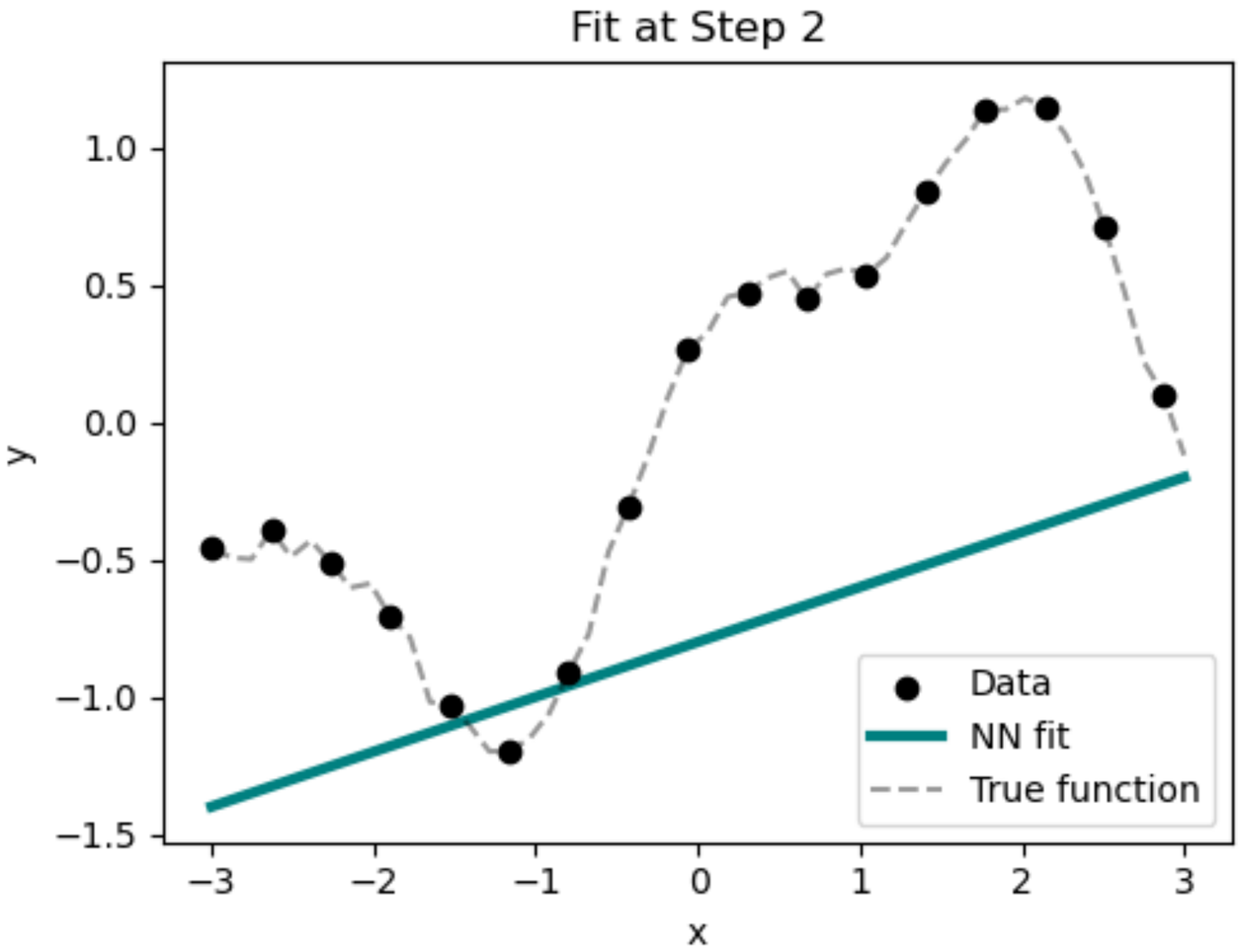


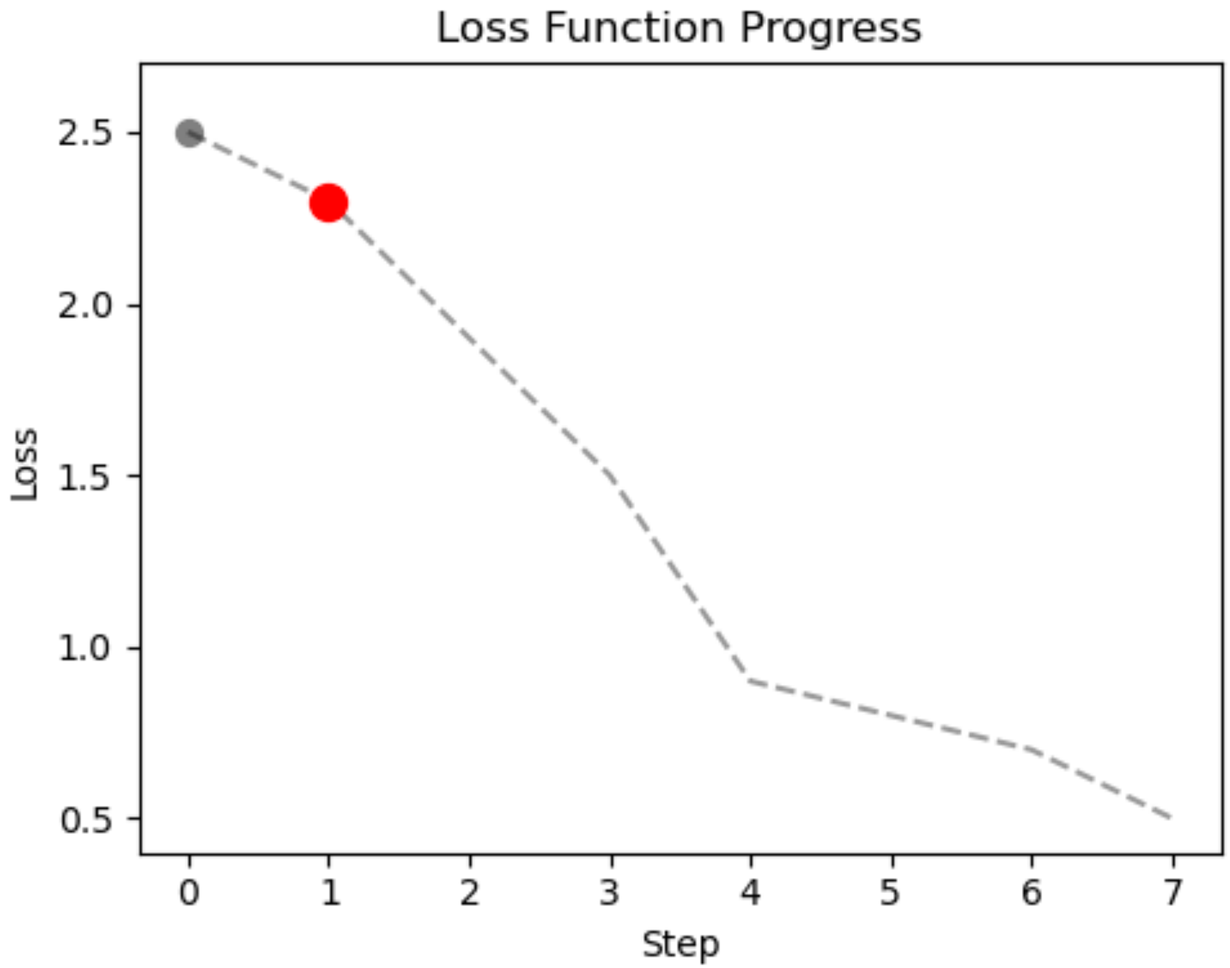
Redes Neuronales: Entrenamiento



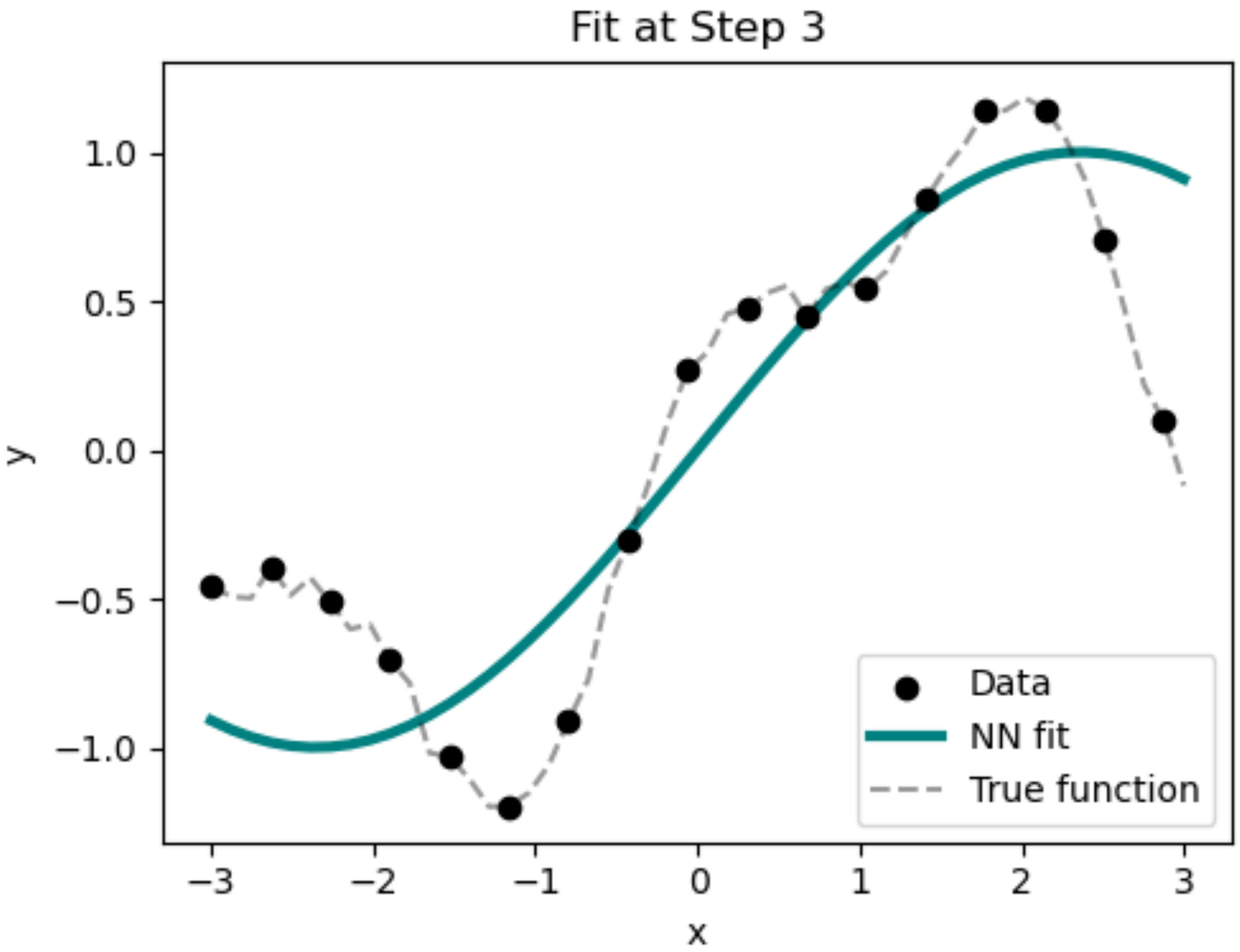


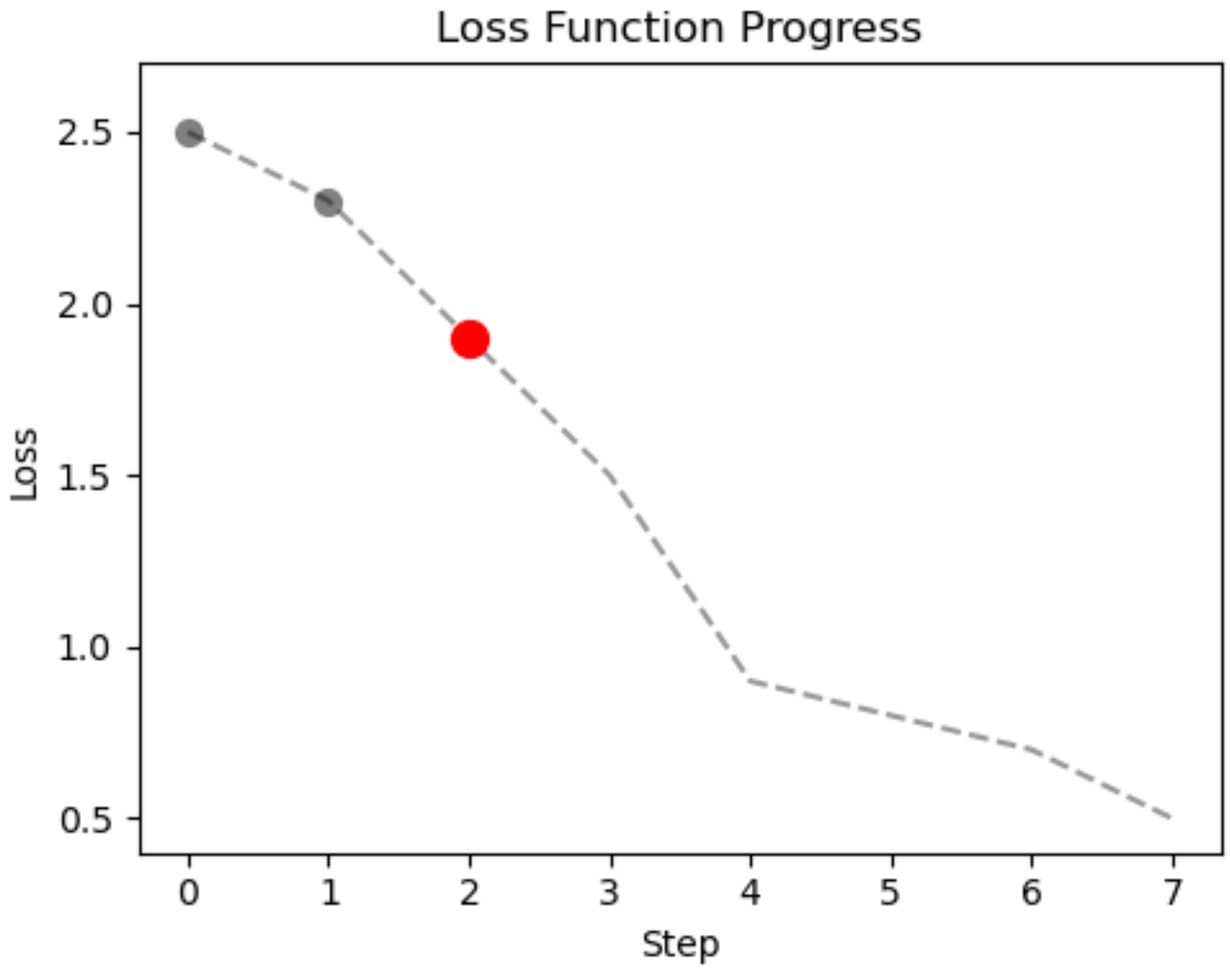
Redes Neuronales: Entrenamiento



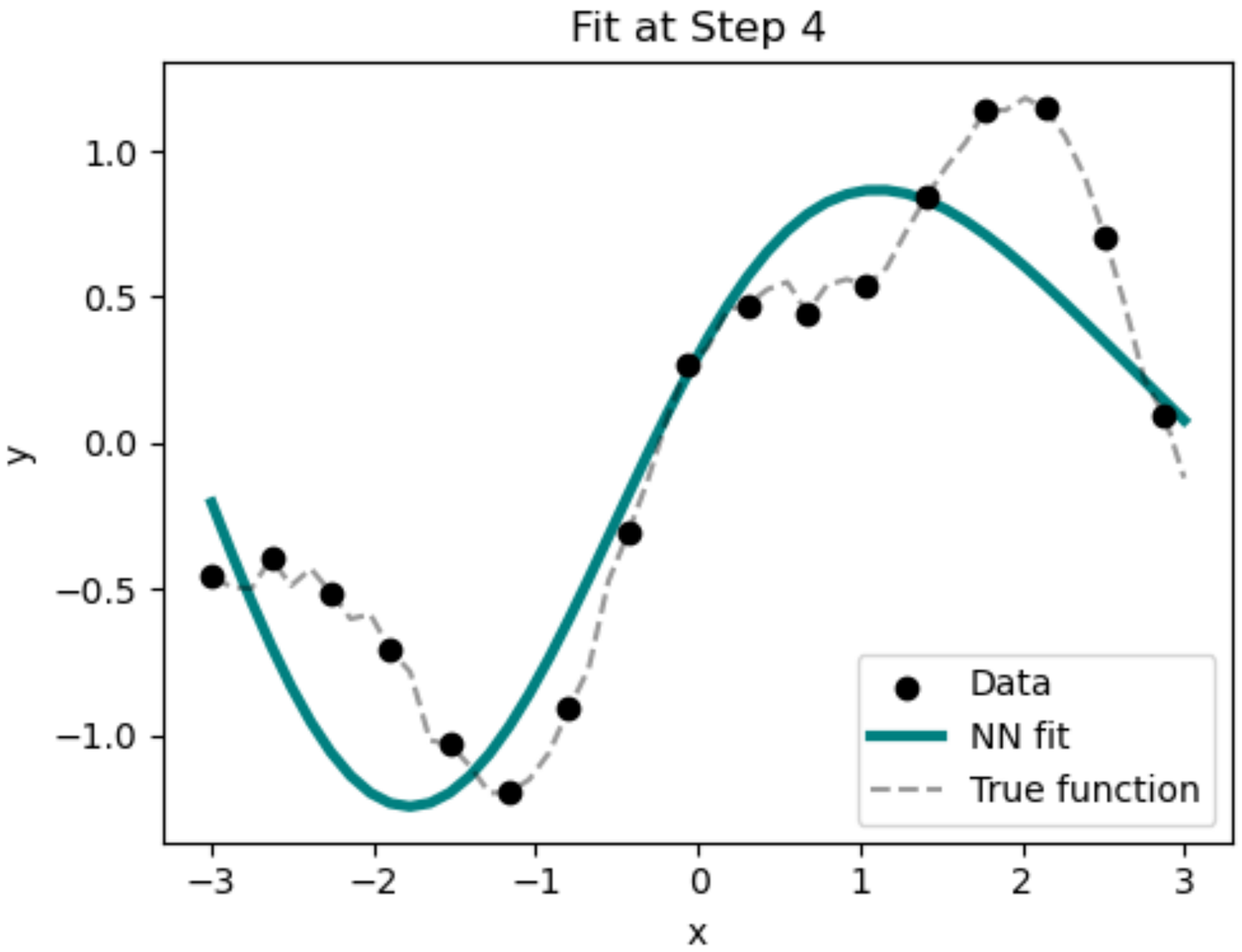


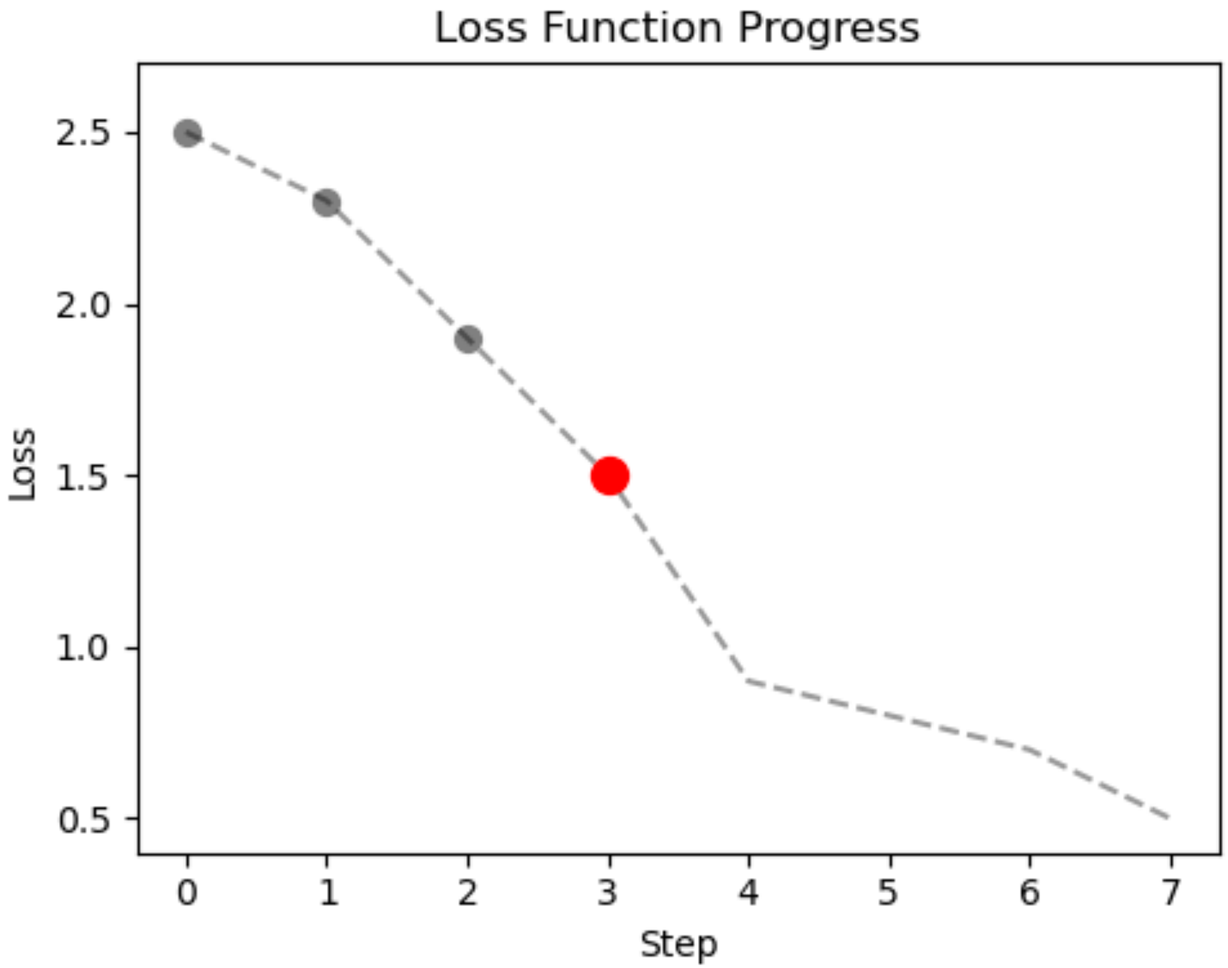
Redes Neuronales: Entrenamiento



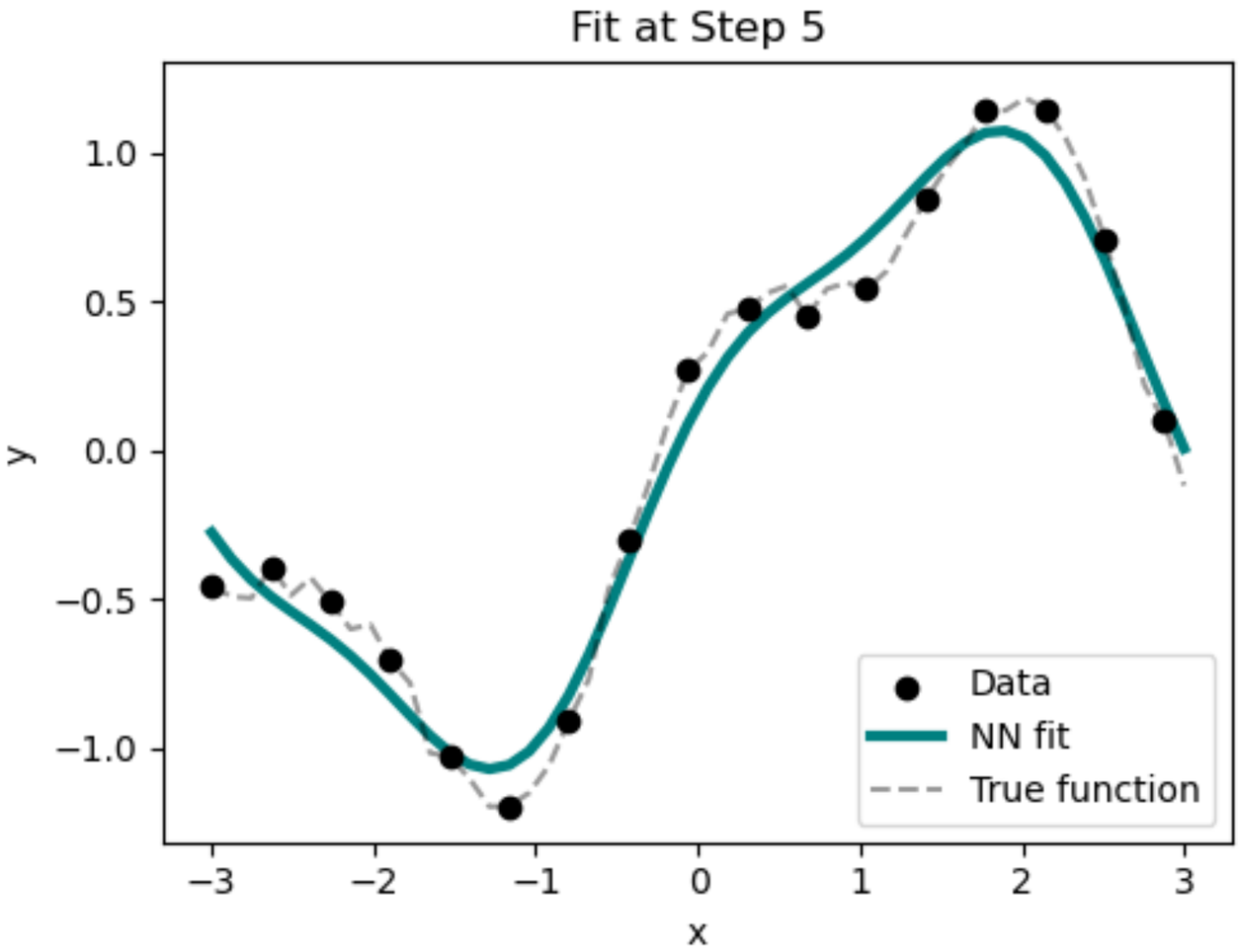


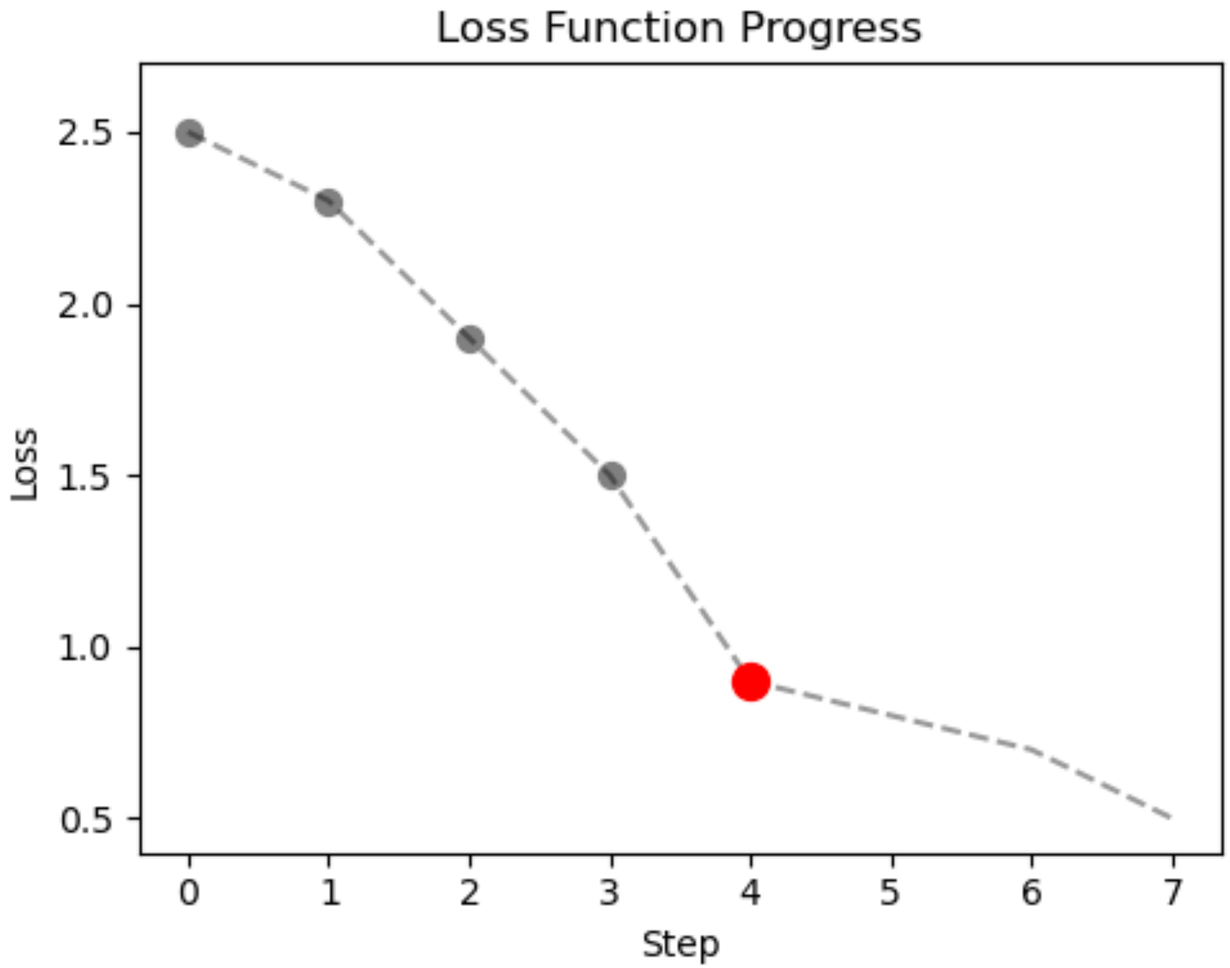
Redes Neuronales: Entrenamiento



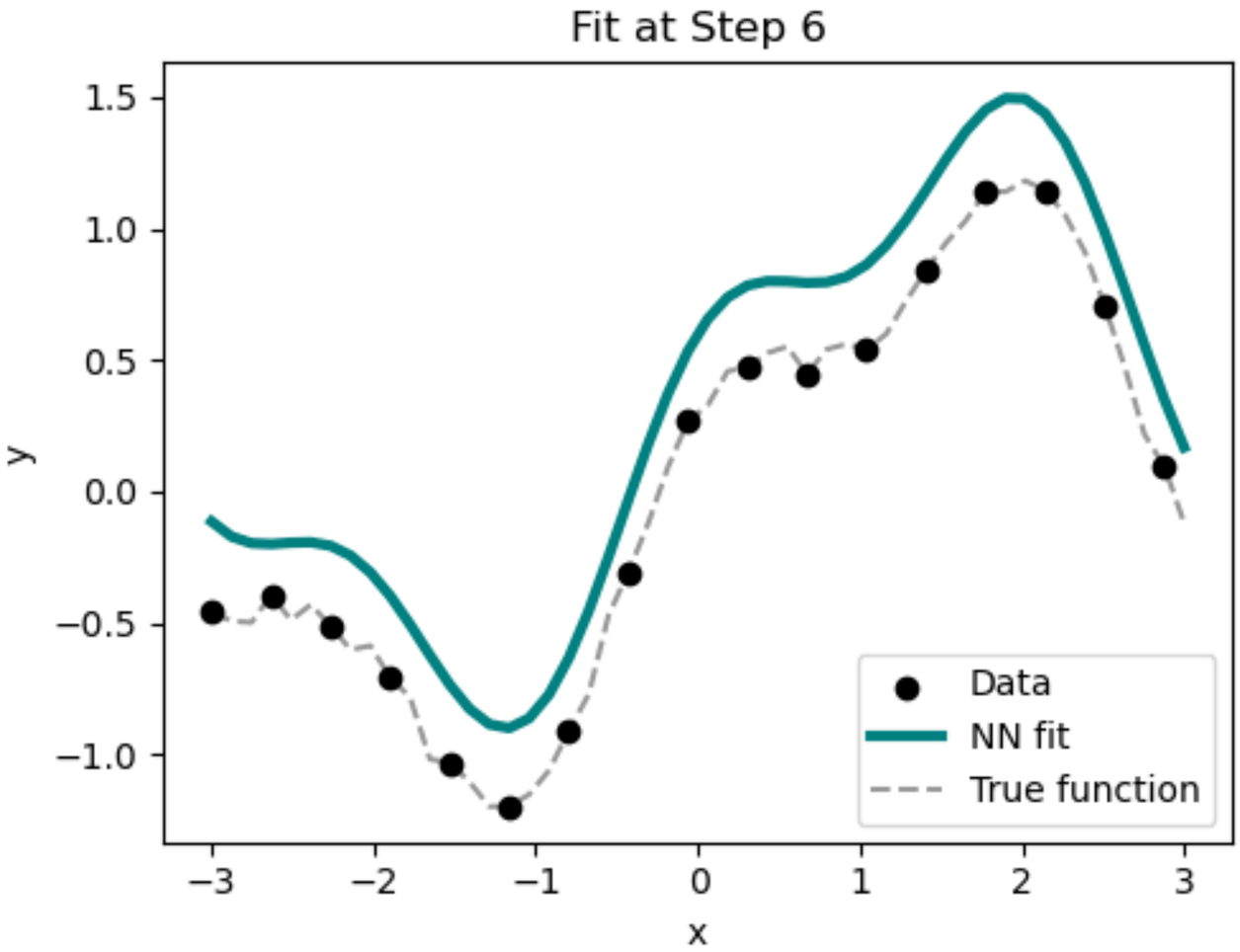


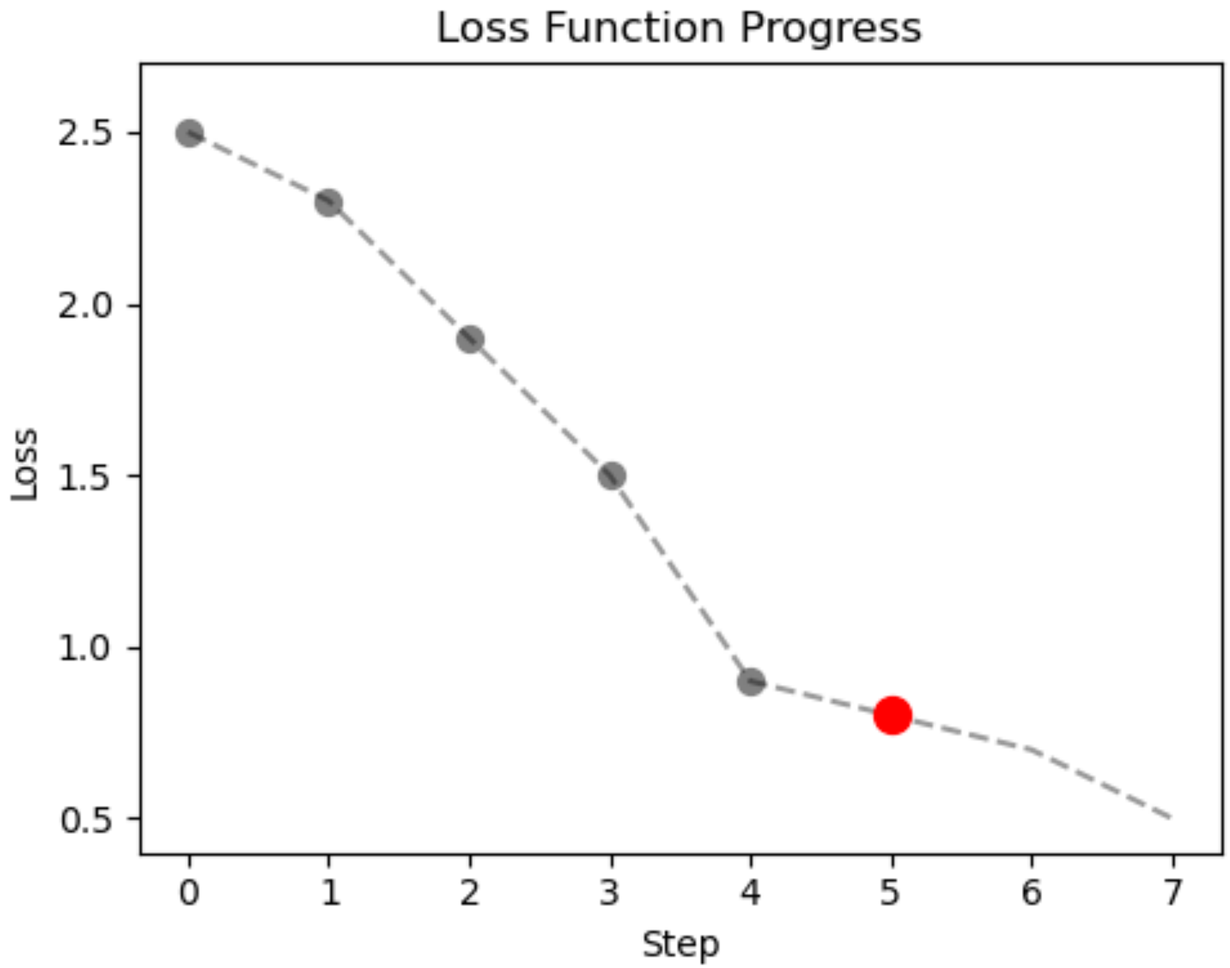
Redes Neuronales: Entrenamiento



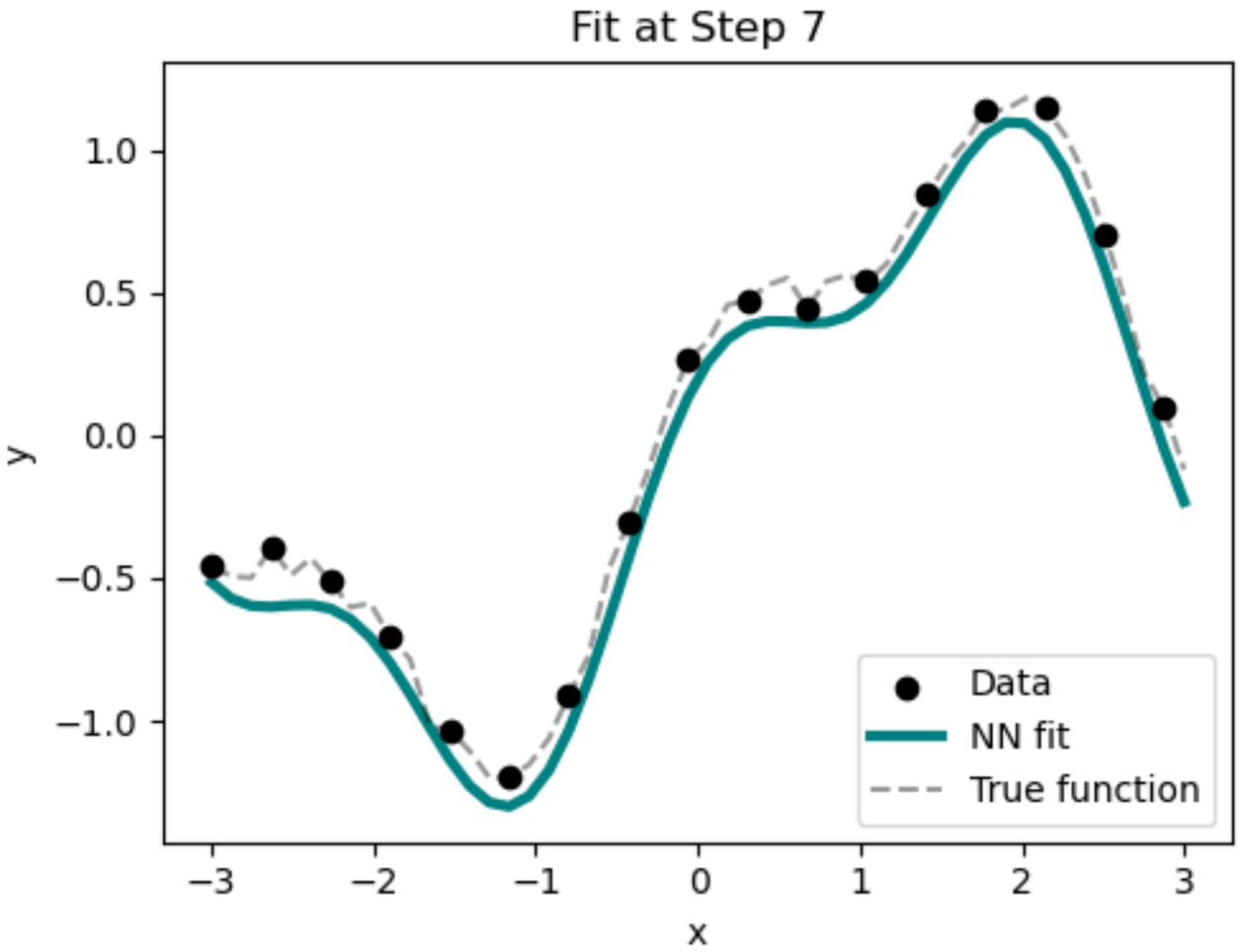


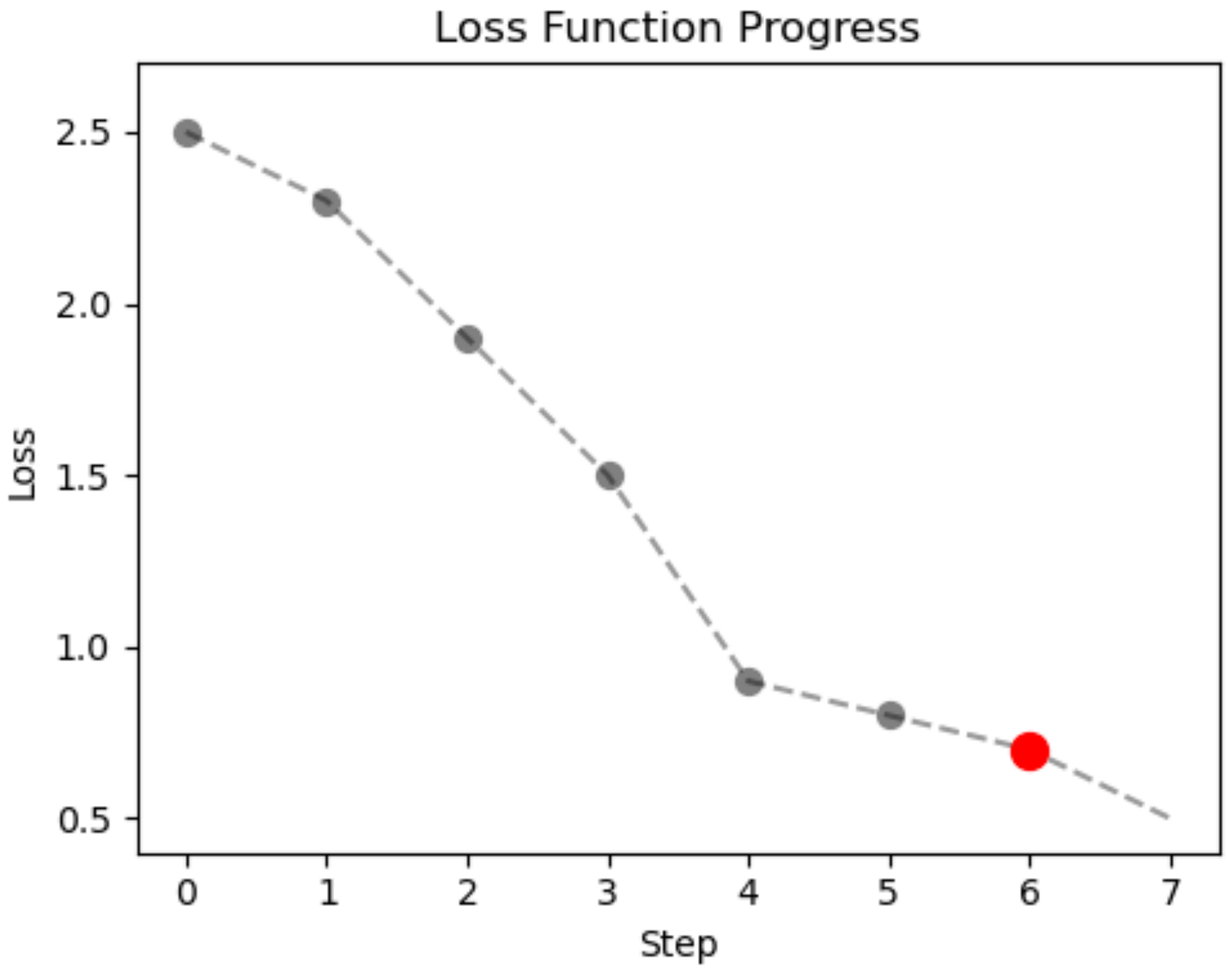
Redes Neuronales: Entrenamiento



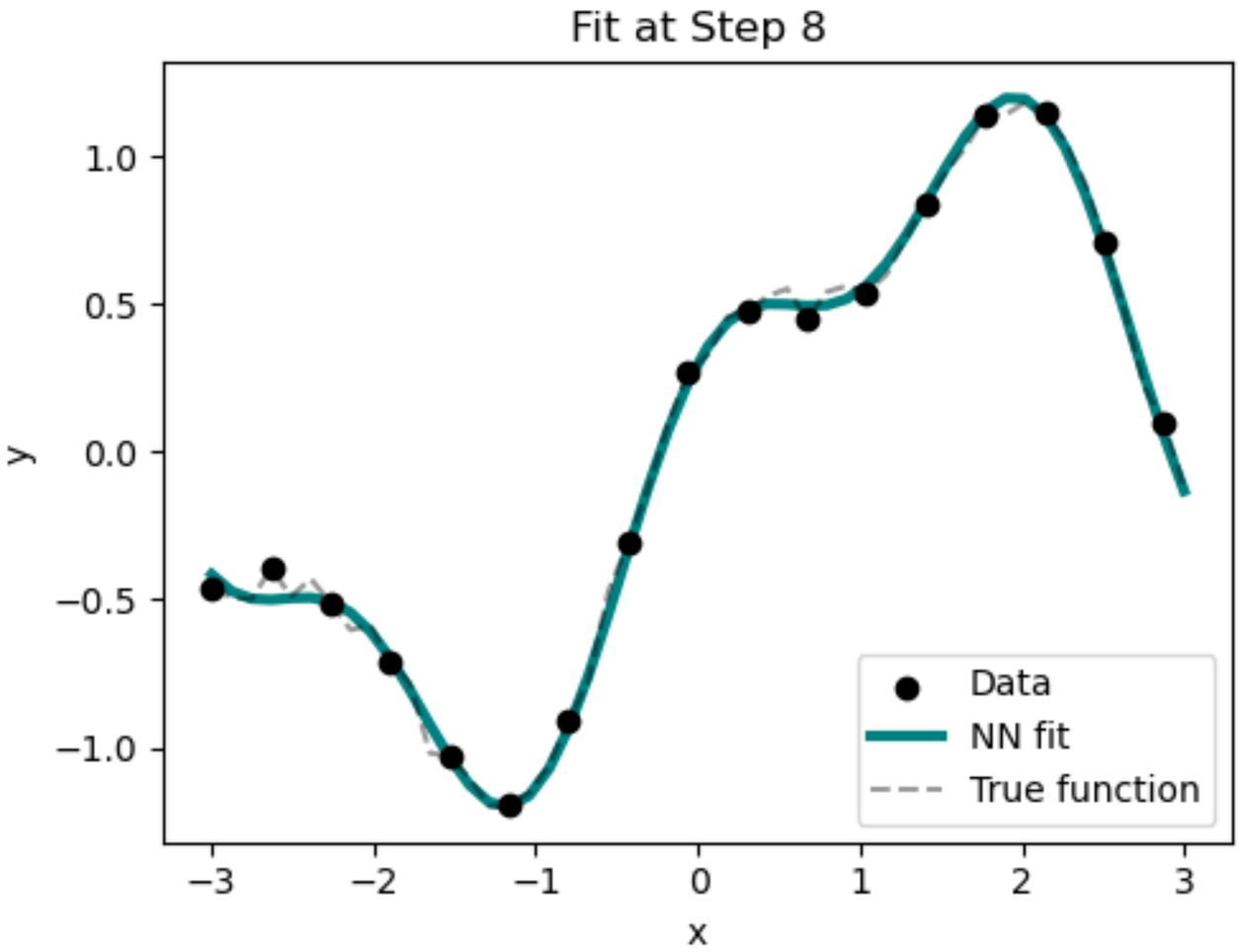


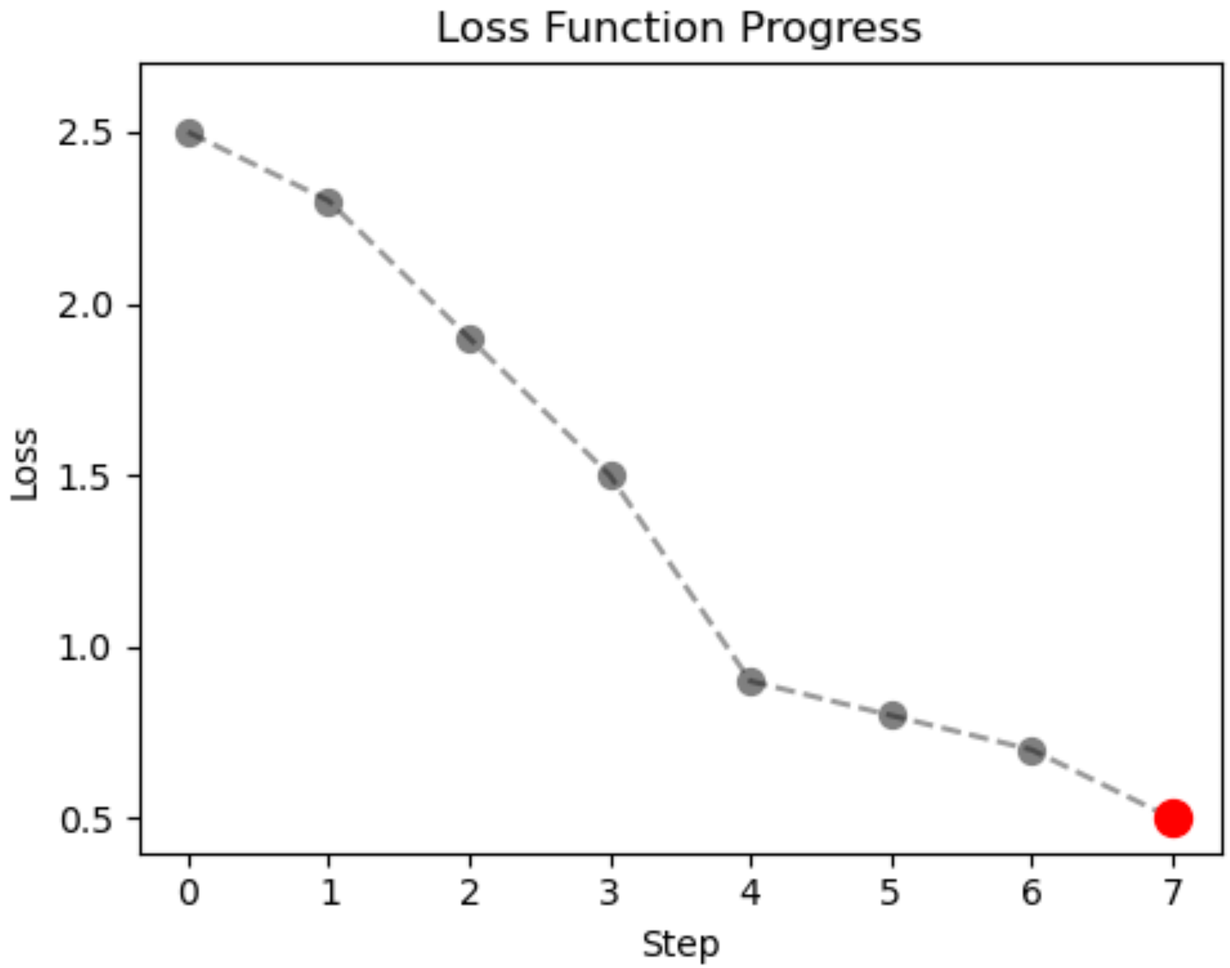
Redes Neuronales: Entrenamiento





Redes Neuronales: Entrenamiento





Fin de Recap

Análisis de Imágenes: classico

Characteristics

- Cuello largo
- Cuatro patas flacas
- Cuernos
- Manchas
- Nariz puntiaguda
- Cola larga con punta peluda

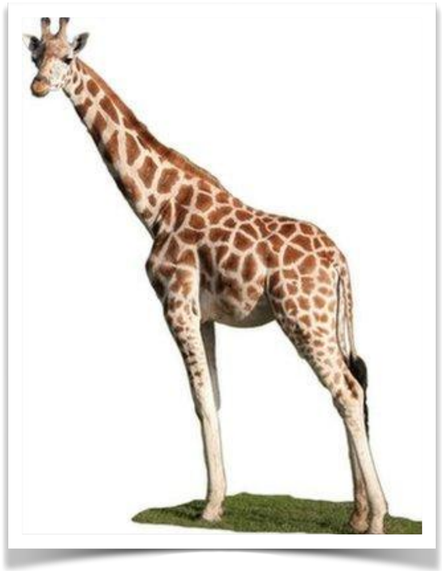
Clasificador

Y/N

función de perdida

Análysis de Imágenes: desafíos

Elección de características



Análysis de Imágenes: desafíos

Casos sutiles, casos special, y falsos positivos

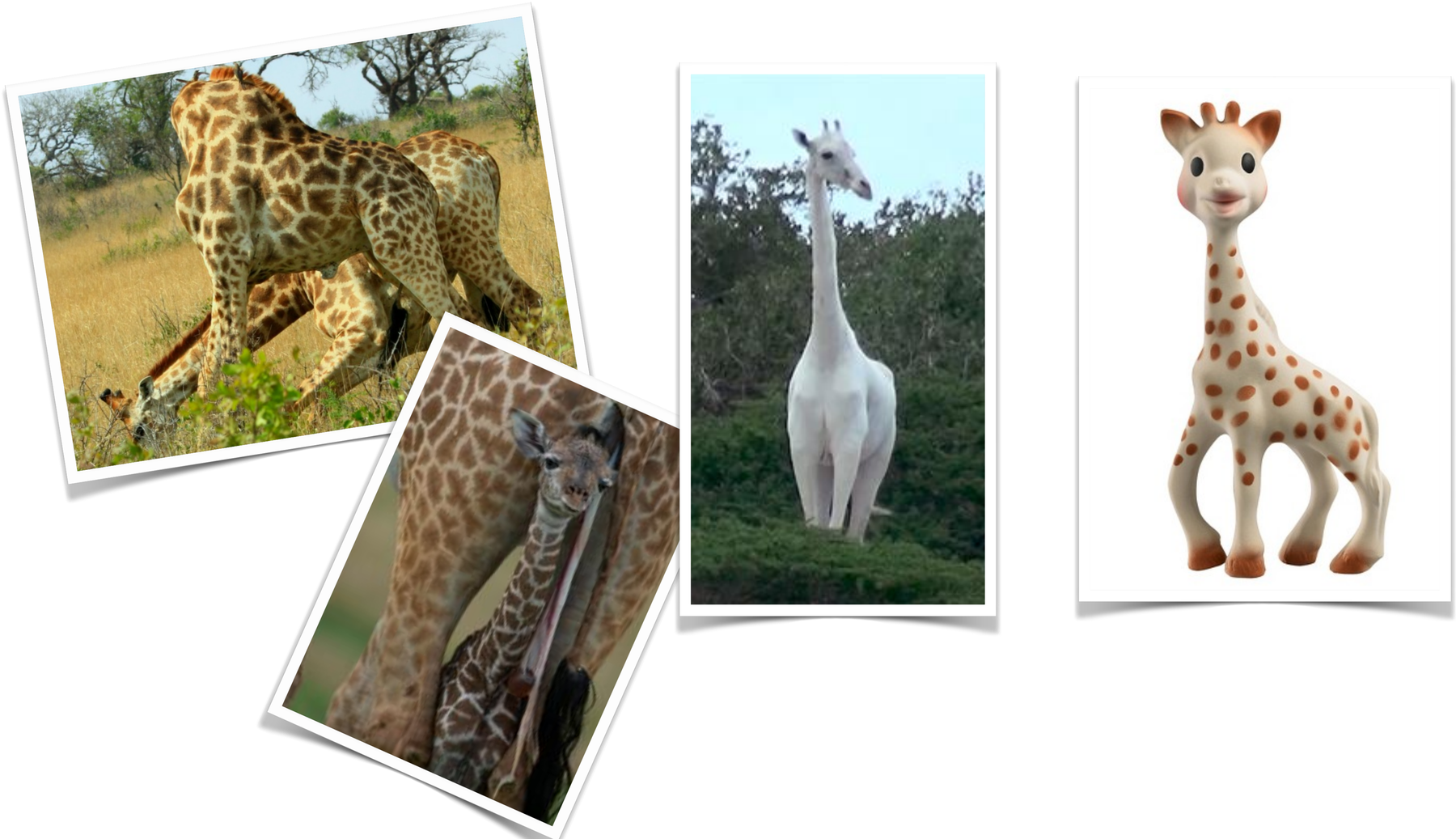


Image Analysis: representation learning

Caracteristicas

- Cuello largo
- Cuatro patas flacas
- Carros features
- Manchas
- Nariz puntiaguda
- Cola larga con punta peluda

Classificador

Y/N

Función de Pérdida

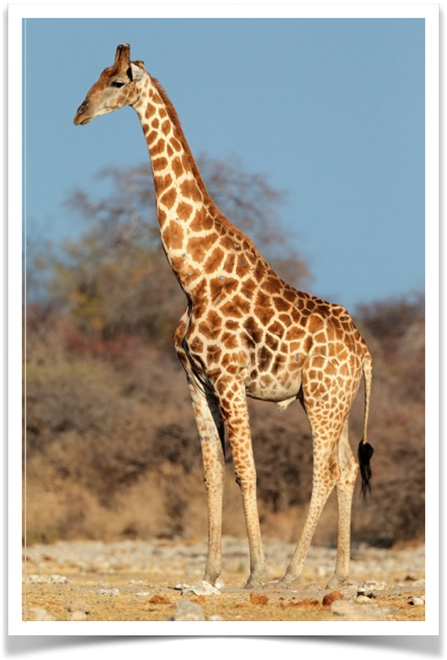
Back propagation

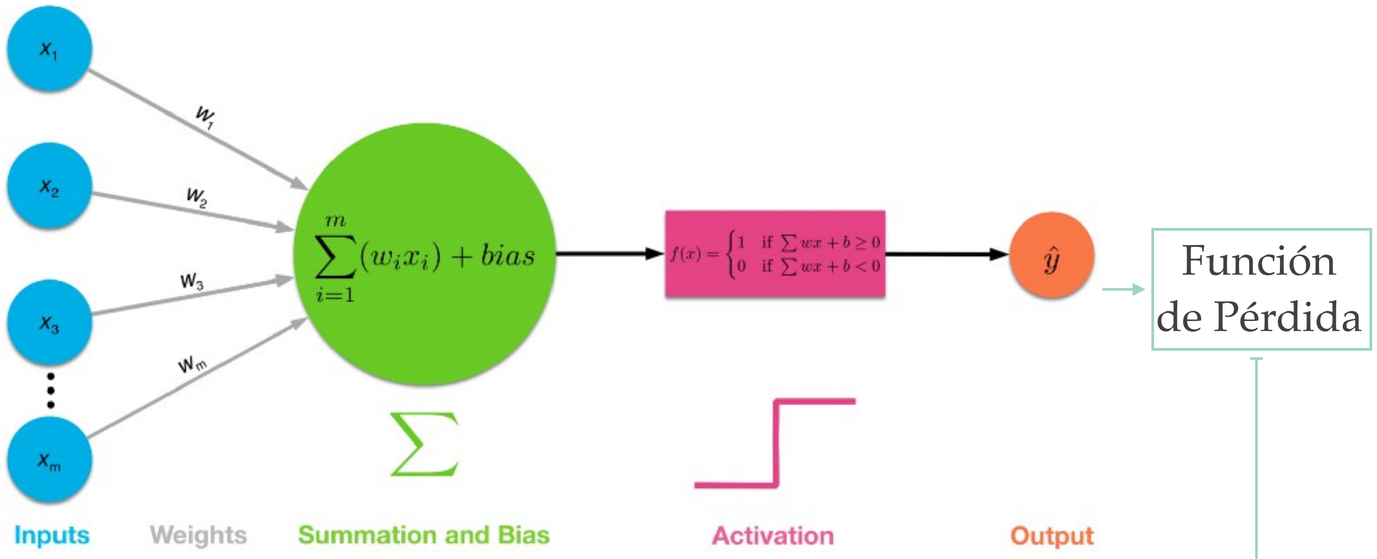
Learning

Ingeniería de Características

Redes NN

MLP drawbacks

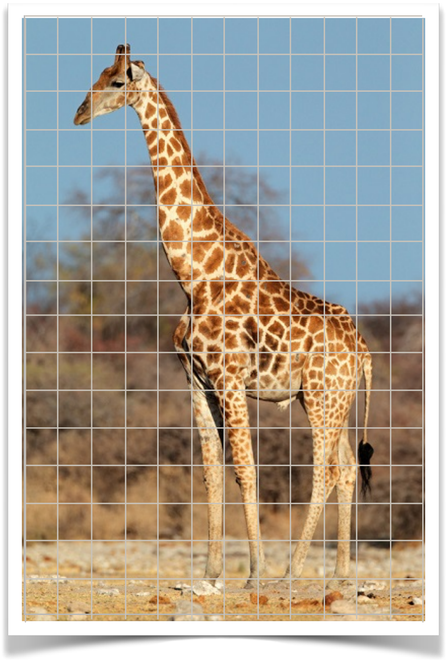


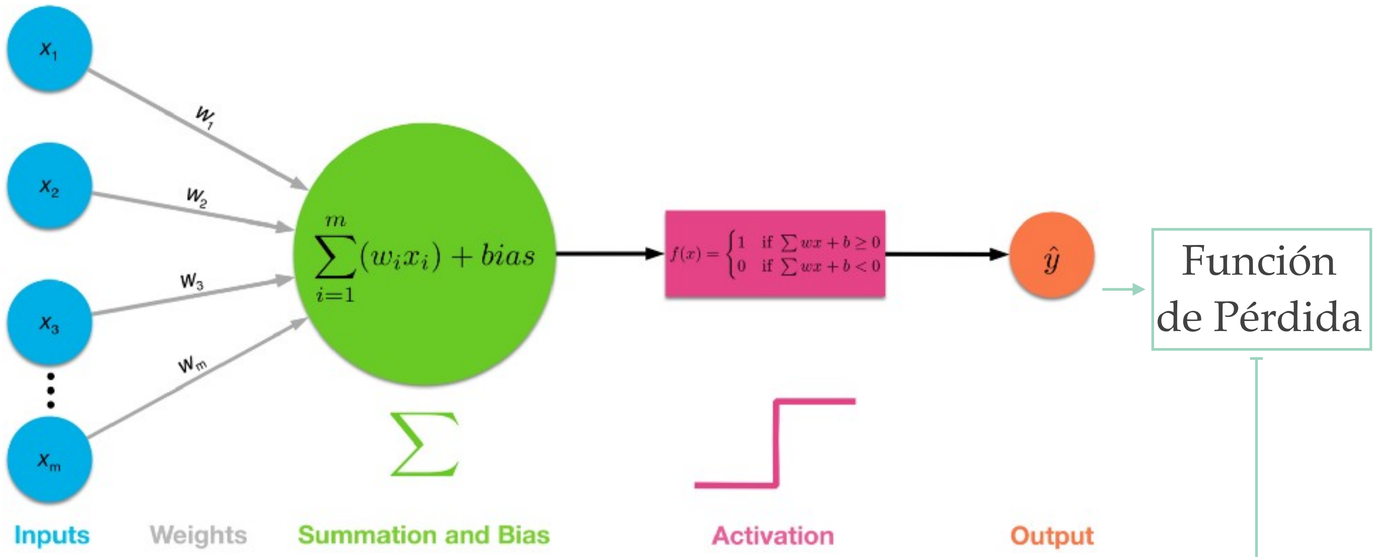


Función de Pérdida

Back propagation

MLP drawbacks



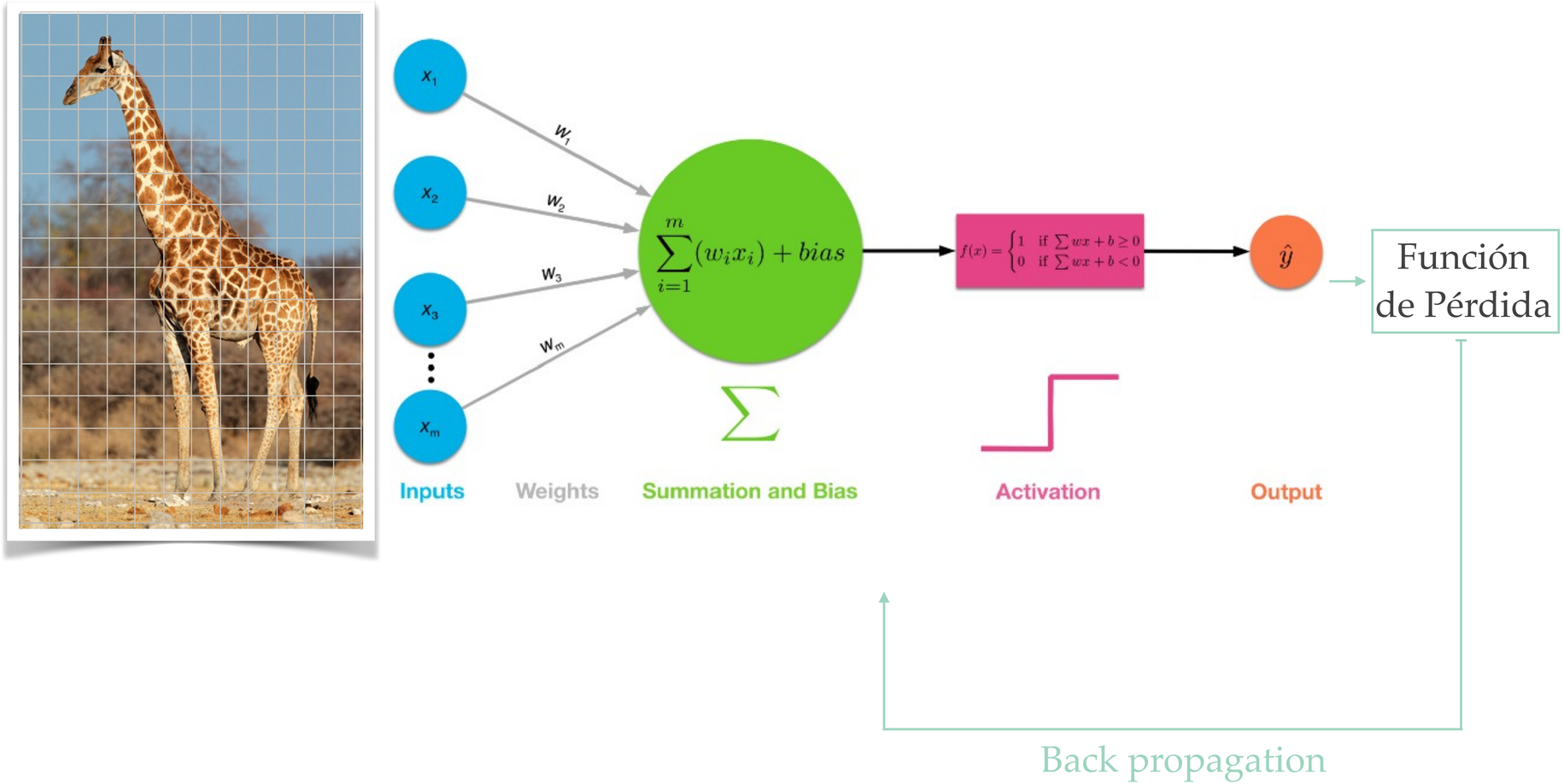


Función de Pérdida

Back propagation

MLP drawbacks: Quiz!

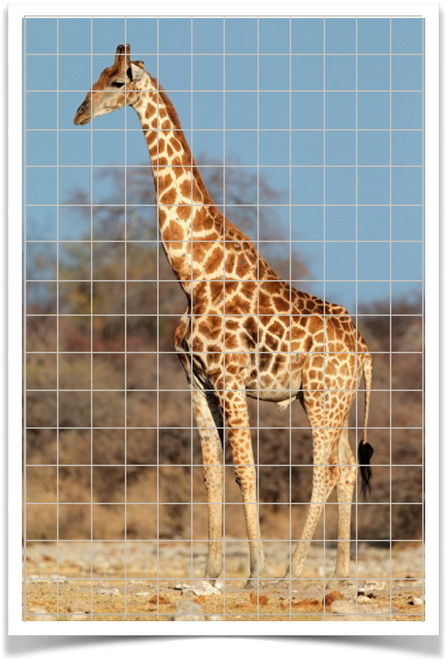
Cuantos parámetros necesitamos para imágenes ImageNet tienen 224x224, con tres colores (rojo, azul y verde), para una red con una sóla capa de 128 neuronas?

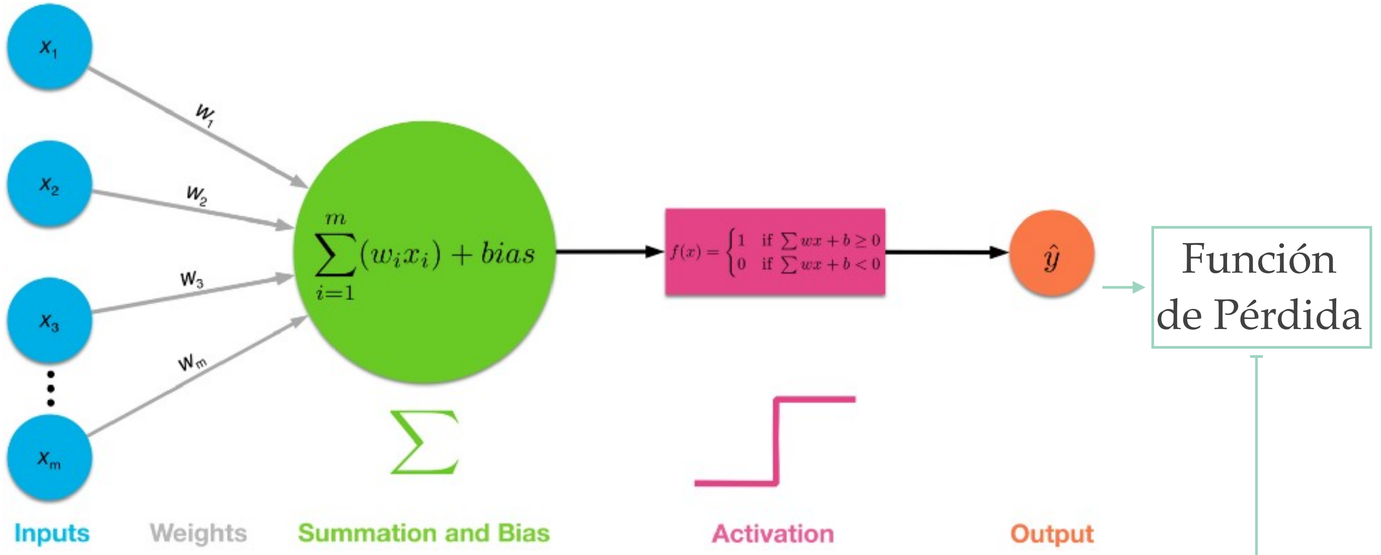


MLP drawbacks: Quiz!

Cuantos parámetros necesitamos para imágenes ImageNet tienen 224x224, con tres colores (rojo, azul y verde), para una red con una sóla capa de 128 neuronas?

(224x224x3+1)x128 = 19,267,712





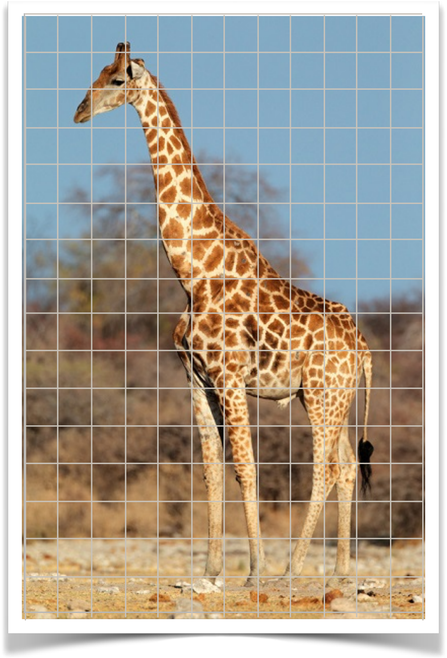
Función de Pérdida

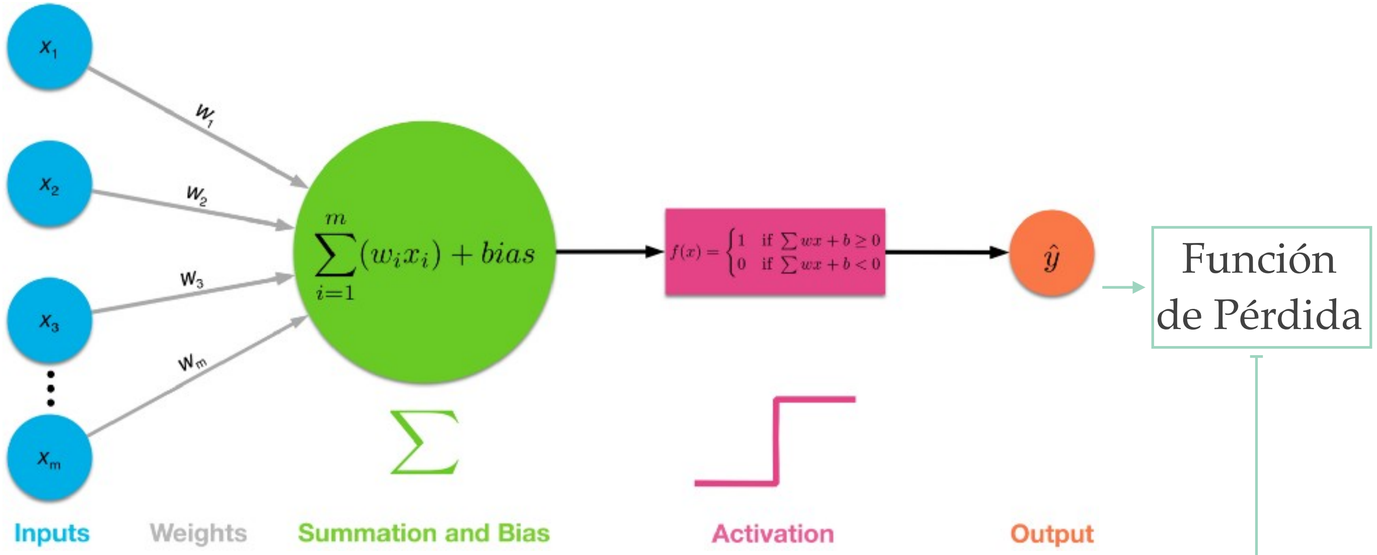
Back propagation

MLP drawbacks: Quiz!

Cuantos parámetros necesitamos para imágenes ImageNet tienen 224x224, con tres colores (rojo, azul y verde), para una red con una sóla capa de 128 neuronas?

(224x224x3+1)x128 +128+1





Función de Pérdida

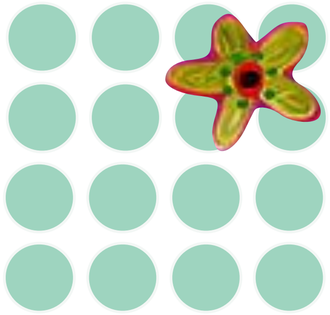
- Gran número de parámetros

Back propagation

MLP drawbacks

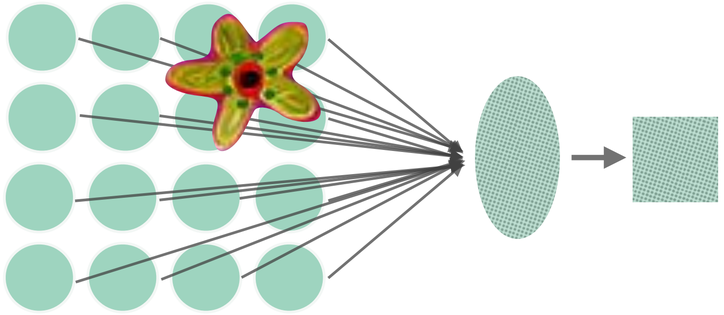
- Gran numero de parámetros
- Redundante, sin atributos compartidos

MLP drawbacks



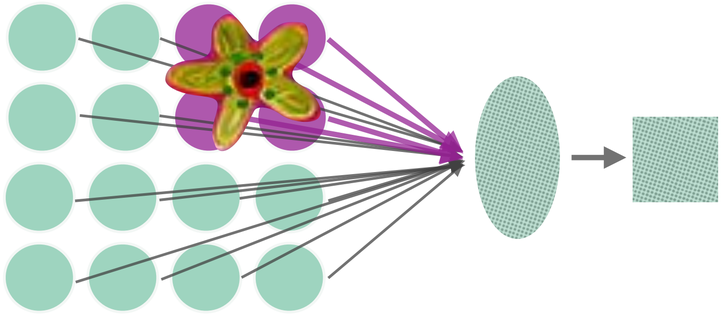
- Gran numero de parámetros
- Redundante, sin atributos compartidos

MLP drawbacks



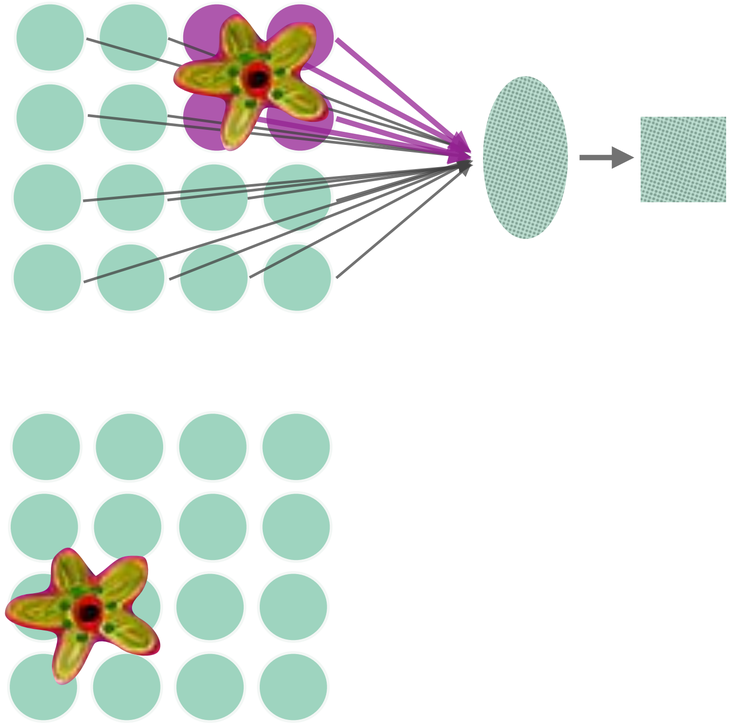
- Gran numero de parámetros
- Redundante, sin atributos compartidos

MLP drawbacks



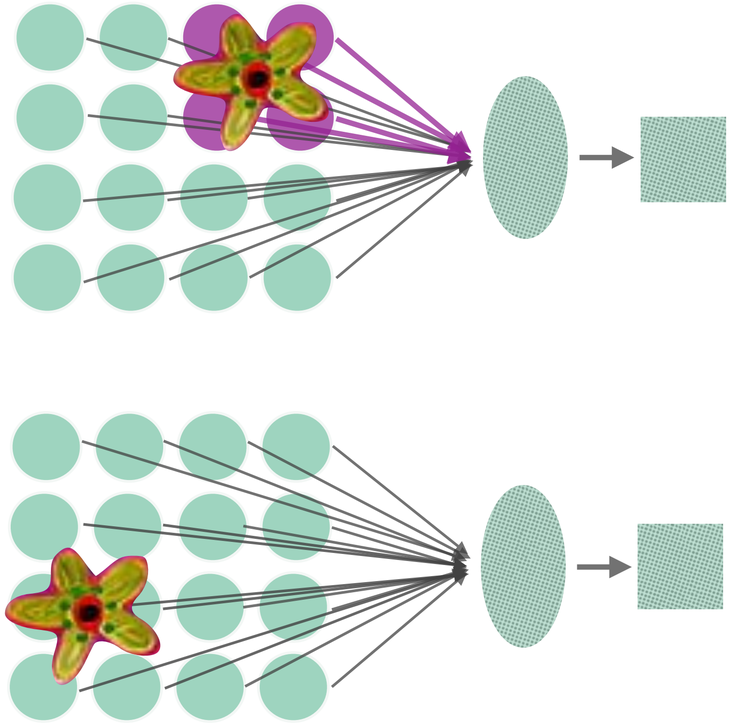
- Gran numero de parámetros
- Redundante, sin atributos compartidos

MLP drawbacks



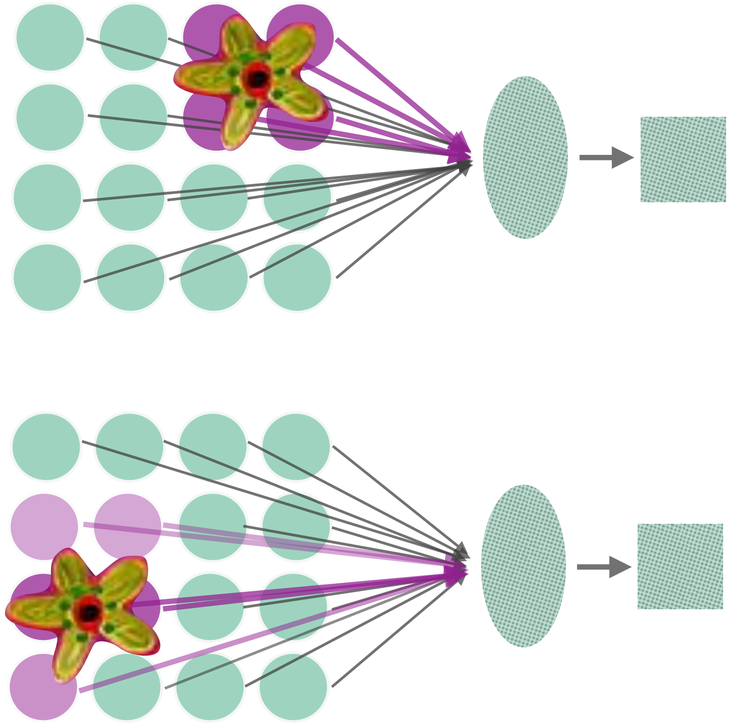
- Gran numero de parámetros
- Redundante, sin atributos compartidos

MLP drawbacks



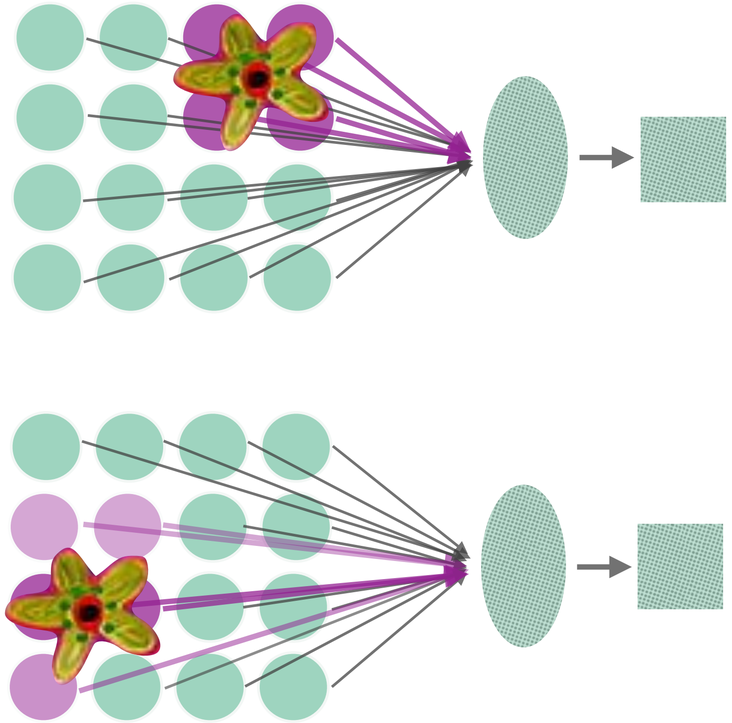
- Gran numero de parámetros
- Redundante, sin atributos compartidos

MLP drawbacks



- Gran numero de parámetros
- Redundante, sin atributos compartidos

MLP drawbacks



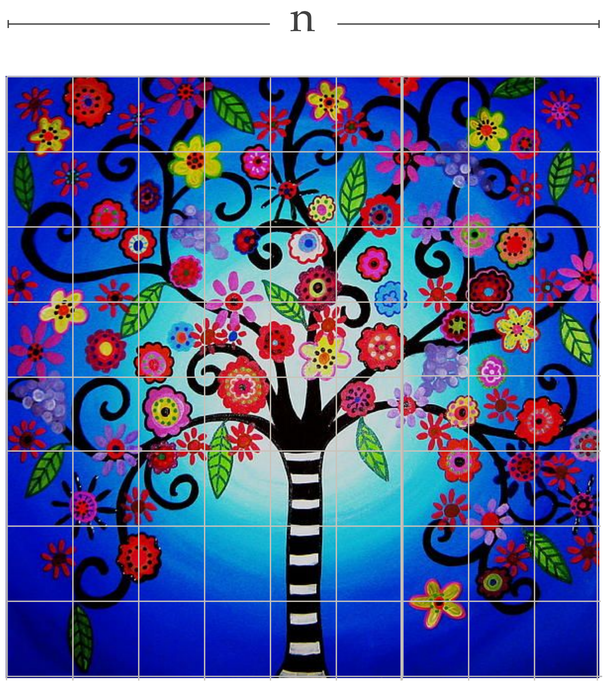
- Gran numero de parámetros
- Redundante, sin atributos compartidos
- Sin invariancia de traslación

→ Computacionalmente costoso , ineficiente, no robusto

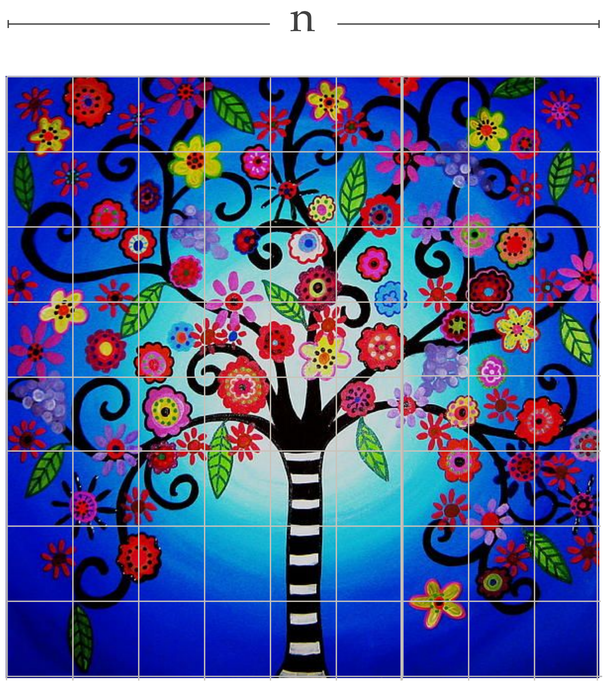
From MLP to CNNs

n

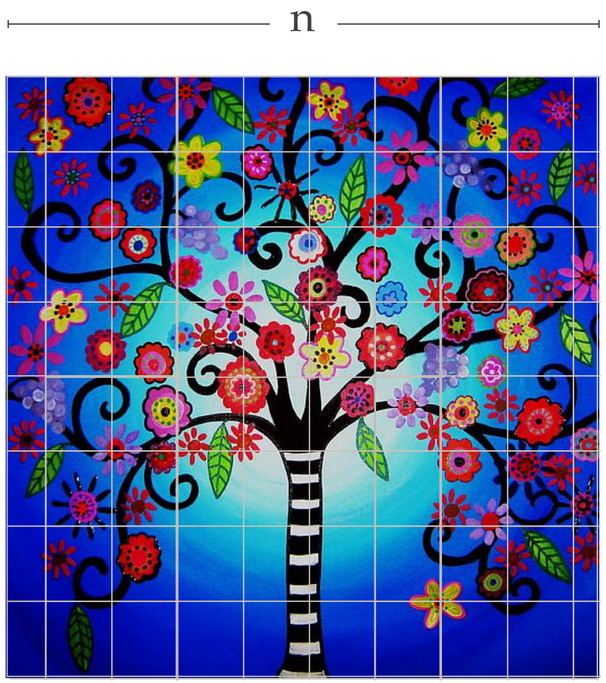
From MLP to CNNs



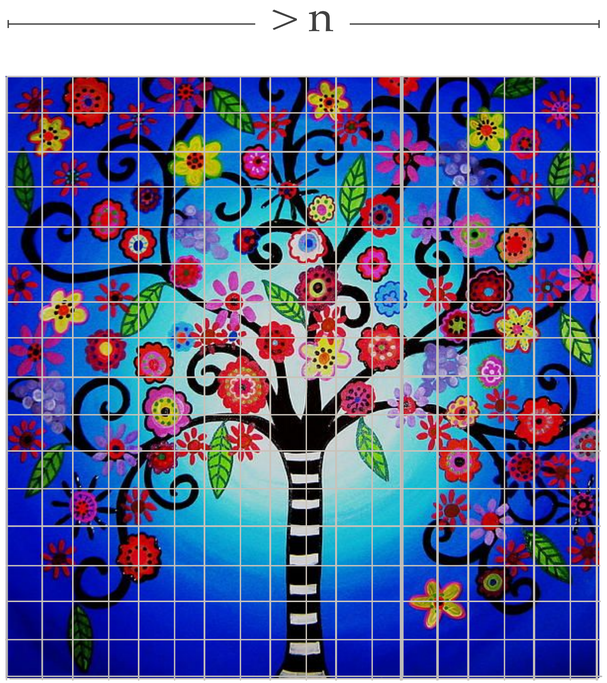
From MLP to CNNs



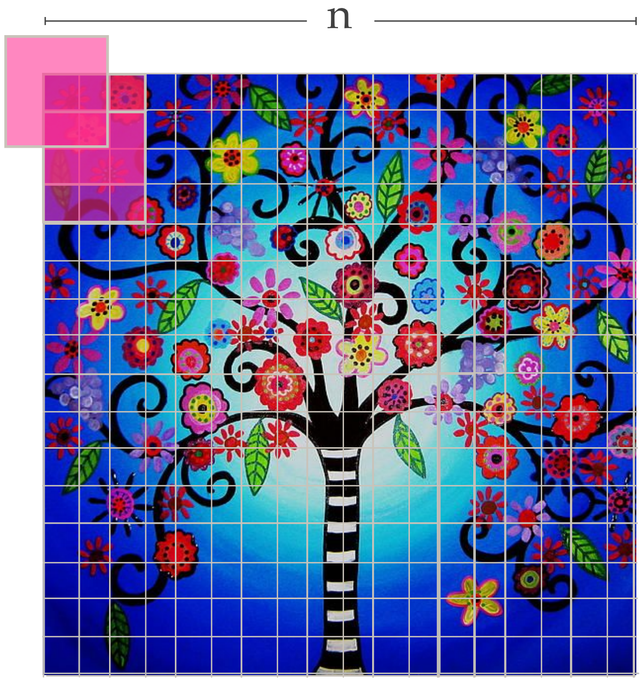
From MLP to CNNs



From MLP to CNNs



From MLP to CNNs



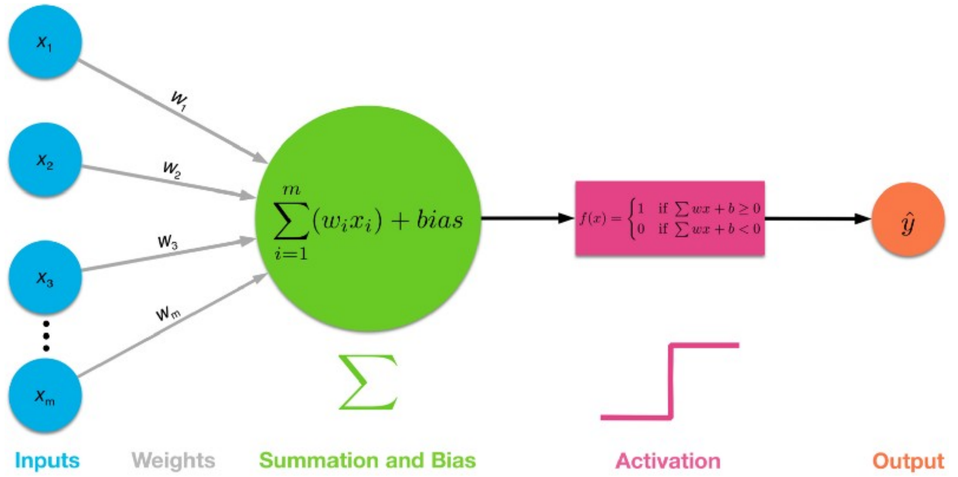
From MLP to CNNs: Quiz!

Cuántas imágenes obtenemos?

From MLP to CNNs

Características compartidas

Tenemos
(n-2)x(n-2)
3x3 images



Para cada neurona aprendemos 9 pesos and un bias:

$$G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} f[u,v]F[i+u, j+v]$$

FILTER

W_1	W_2	W_3
W_4	W_5	W_6
W_7	W_8	W_9

From MLP to CNNs

$$G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} f[u,v]F[i+u,j+v]$$

From MLP to CNNs

$$G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} f[u,v]F[i+u,j+v]$$

From MLP to CNNs

100	81	25	94	21	12	210	26	167
45	30	7	28	79	12	10	15	138
70	81	25	94	5	12	231	26	160
101	90	25	82	79	189	210	120	180
98	81	250	85	75	186	199	167	145
92	99	233	74	62	191	205	190	167
101	108	210	94	76	192	210	222	117

$$G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} f[u,v]F[i+u, j+v]$$

From MLP to CNNs

100	81	25	94	21	12	210	26	167
45	30	7	28	79	12	10	15	138
70	81	25	94	5	12	231	26	160
101	90	25	82	79	189	210	120	180
98	81	250	85	75	186	199	167	145
92	99	233	74	62	191	205	190	167
101	108	210	94	76	192	210	222	117

f1 =

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

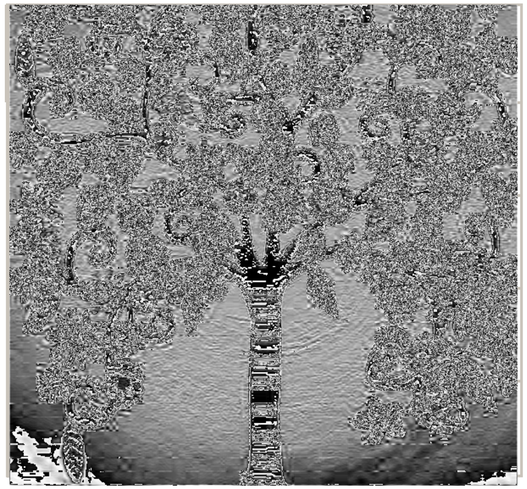
$$G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} f[u,v]F[i+u, j+v]$$

From MLP to CNNs

100 -1	81 -2	25 -1	94	21	12	210	26	167
45 0	30 0	7 0	28	79	12	10	15	138
70 1	81 2	25 1	94	5	12	231	26	160
101 0	90 0	25 0	82	79	189	210	120	180
98 0	81 0	250 0	85	75	186	199	167	145
92 0	99 0	233 0	74	62	191	205	190	167
101 1	108 2	210 1	94	76	192	210	222	117

f1 =

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

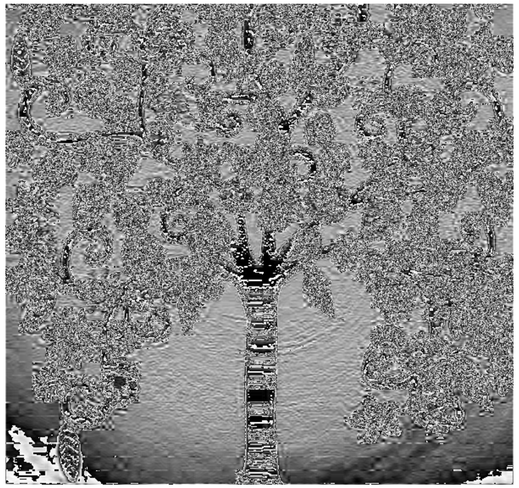


$$G[i,j] = \sum_{u=0}^{k-1} \sum_{v=0}^{k-1} f[u,v] F[i+u, j+v]$$

From MLP to CNNs

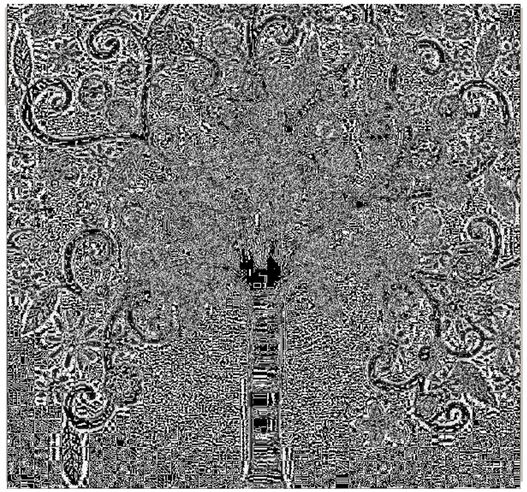
f1 =

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



f2 =

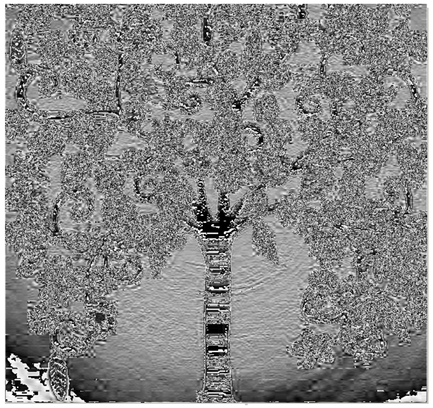
0	1	0
1	-4	1
0	1	0



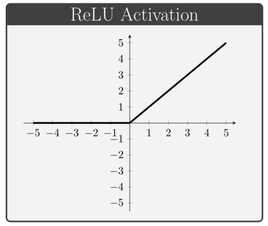
$$G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} f[u,v]F[i+u,j+v]$$

From MLP to CNNs

f_1

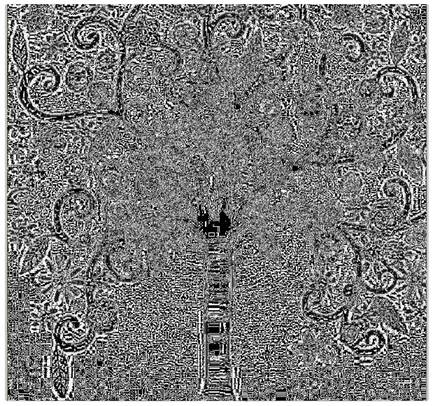


+ bias
&

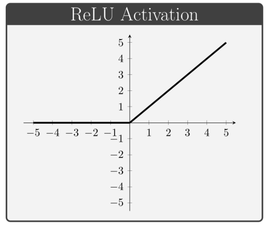


0	0	0	0	5	42	0
194	150	126	231	554	682	452
253	441	277	305	386	251	235
217	417	175	0	0	62	122
17	0	0	17	24	83	93

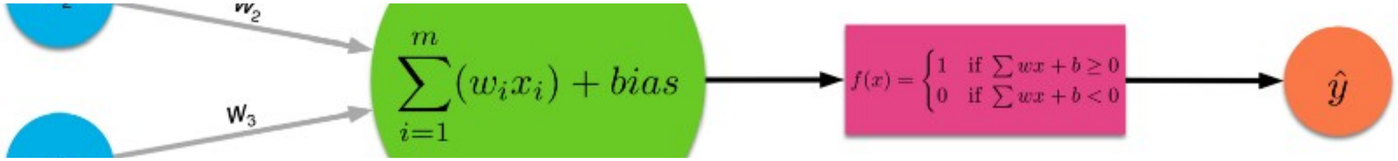
f_2



+ bias
&



0	6	0	0	1	48	0
200	156	132	237	560	689	461
259	447	283	311	392	257	241
223	423	181	0	0	68	128
23	0	0	23	30	89	99



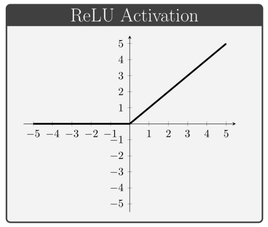
From MLP to CNNs

f ⟼

-30	0	-16	-32	5	42	-14
194	150	126	231	554	682	452
253	441	277	305	386	251	235
217	417	175	-40	-18	62	122
17	-44	-21	17	24	83	93

+ bias

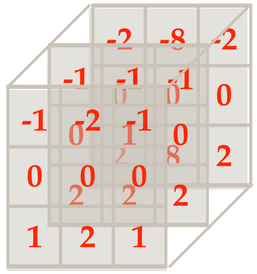
&



⟼

0	0	0	0	5	42	0
194	150	126	231	554	682	452
253	441	277	305	386	251	235
217	417	175	0	0	62	122
17	0	0	17	24	83	93

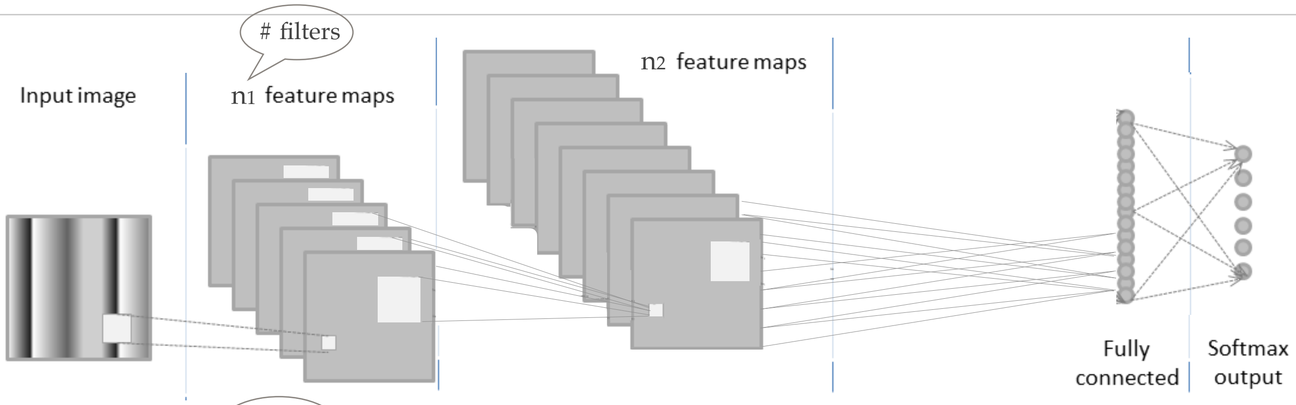
f =

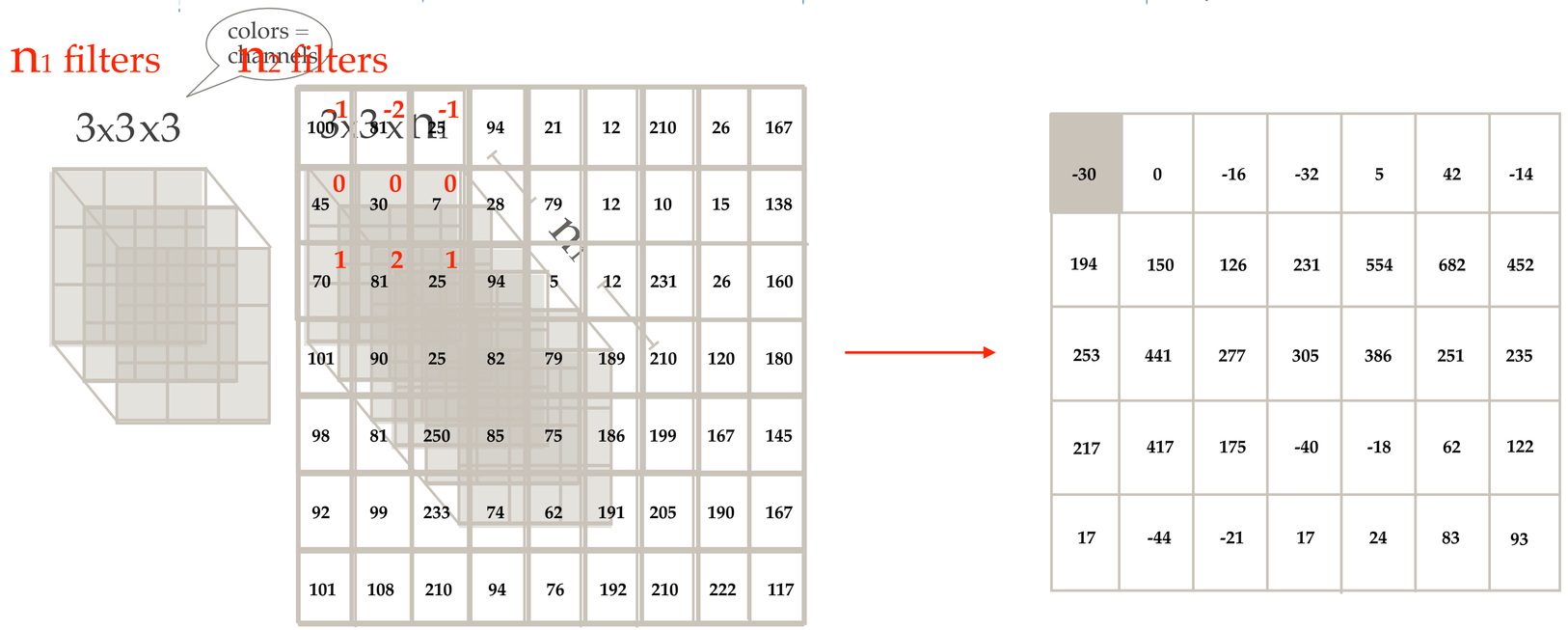


From MLP to CNNs

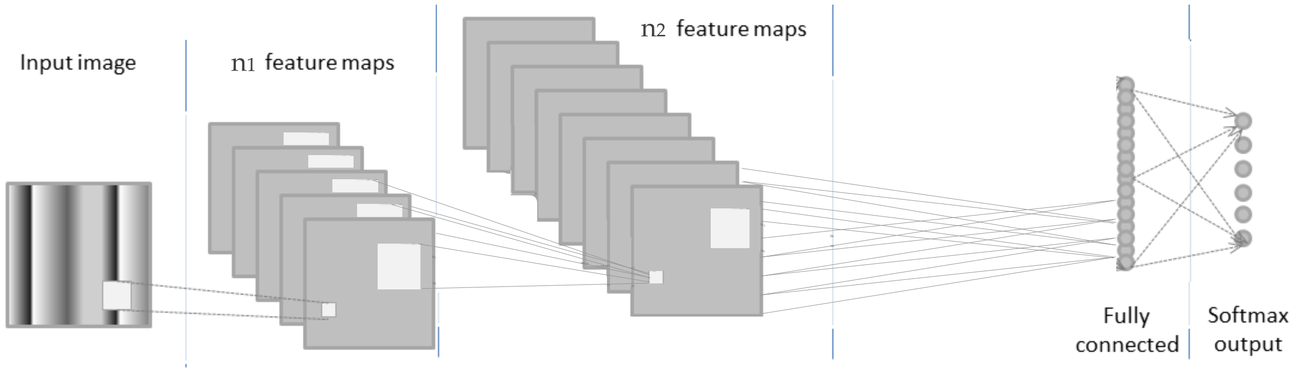
- Gran numero de parámetros ✓
- Redundante, sin atributos compartidos ✓
- Sin invariancia de traslación

CNNs basics

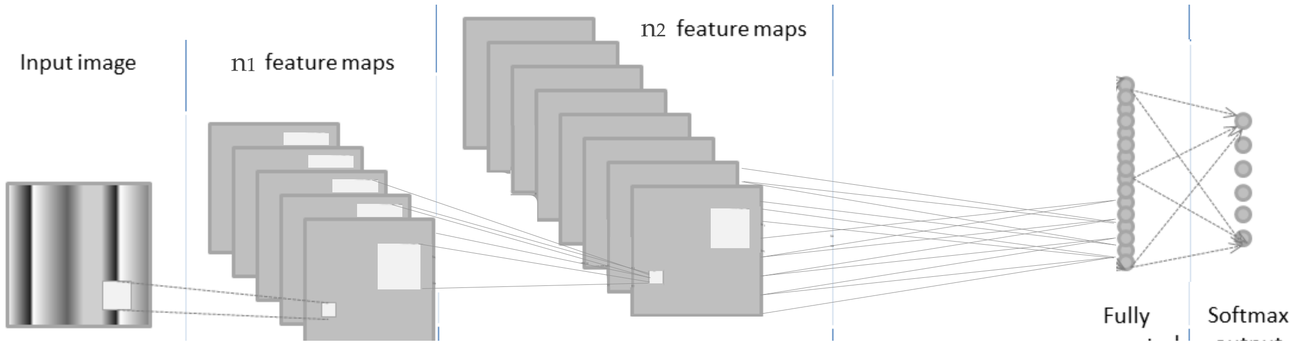

filters
Input image
n1 feature maps
n2 feature maps
Fully connected
Softmax output


n1 filters
colors = channels
n2 filters
3x3x3
3x3xn1
-1 -2 -1
0 0 0
1 2 1
n1
100 31 25 94 21 12 210 26 167
45 30 7 28 79 12 10 15 138
70 81 25 94 5 12 231 26 160
101 90 25 82 79 189 210 120 180
98 81 250 85 75 186 199 167 145
92 99 233 74 62 191 205 190 167
101 108 210 94 76 192 210 222 117
-30 0 -16 -32 5 42 -14
194 150 126 231 554 682 452
253 441 277 305 386 251 235
217 417 175 -40 -18 62 122
17 -44 -21 17 24 83 93

CNNs basics

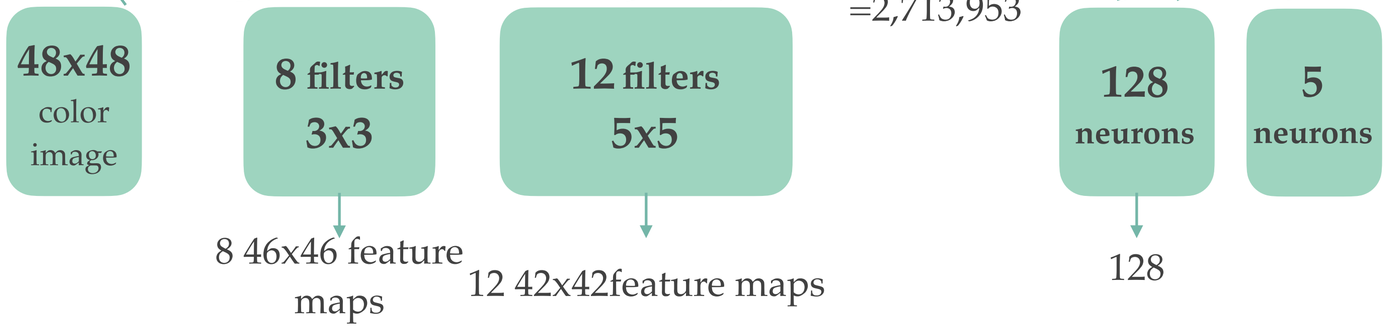

Input image
n1 feature maps
n2 feature maps
Fully connected
Softmax output

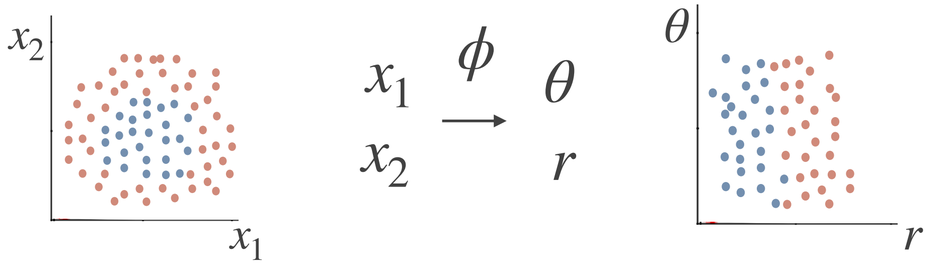
CNNs basics



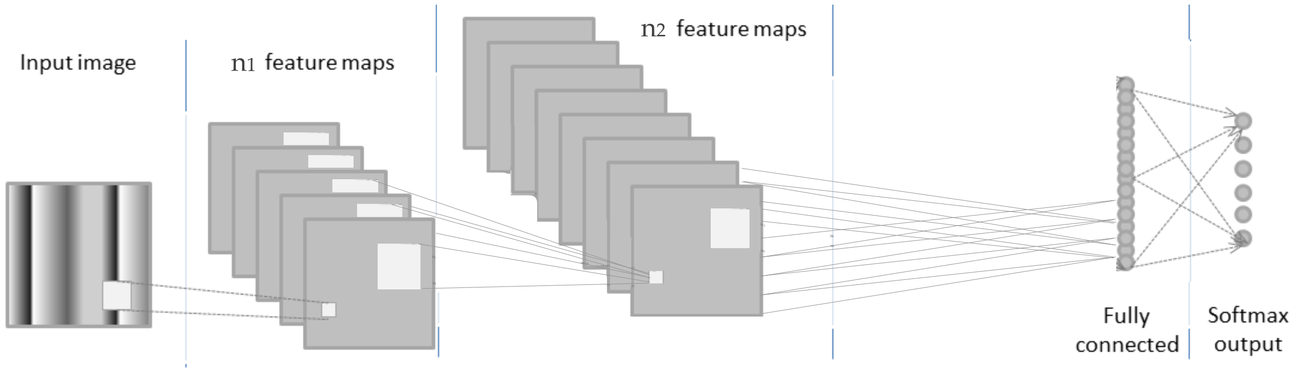
In class quiz: Cuantos parámetros tiene esta red?

Parameteros: (3x3x3+1)x8 (5x5x8+1)x12 12x42x42x128+128 =2,713,953 128x5+5 = 2,714,097

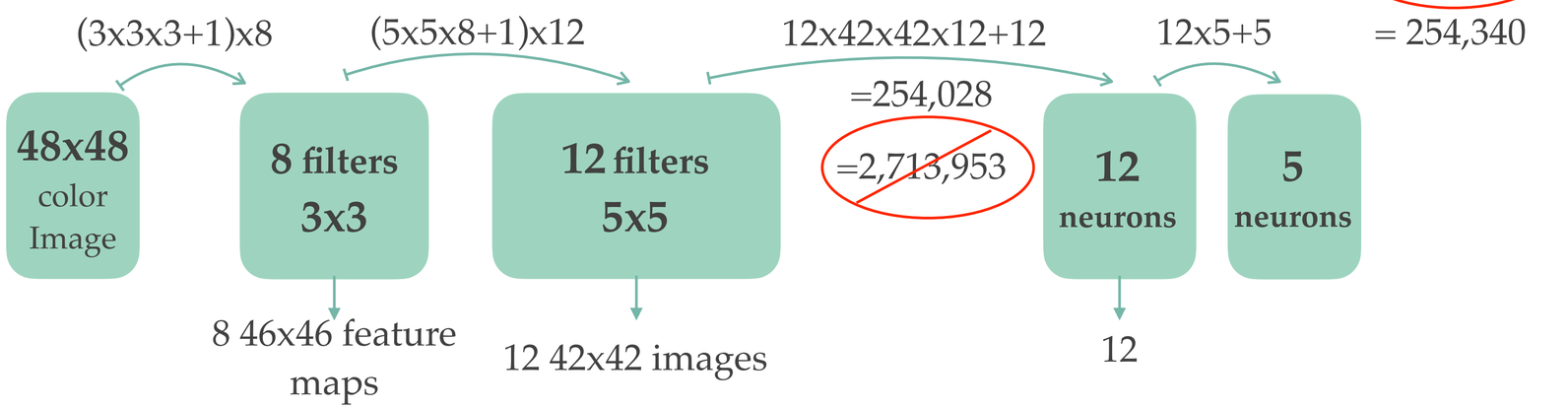




CNNs basics



Parameters:

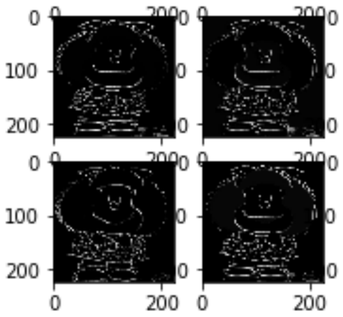


- Redundant, no shared features ✓
- No translation invariance ✓
- Large number of parameter ✓

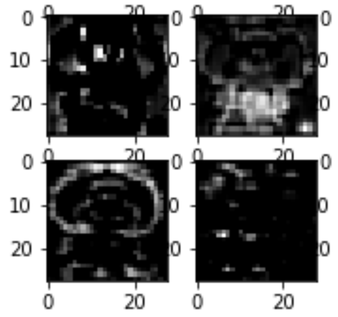
CNNs Layers

VGG

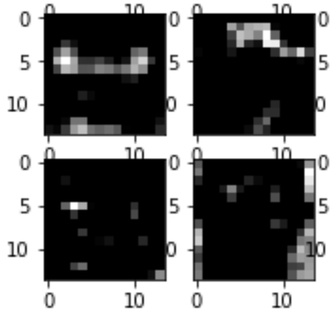
Simonyan & Zisserman
Oxford 2014



Layer 1

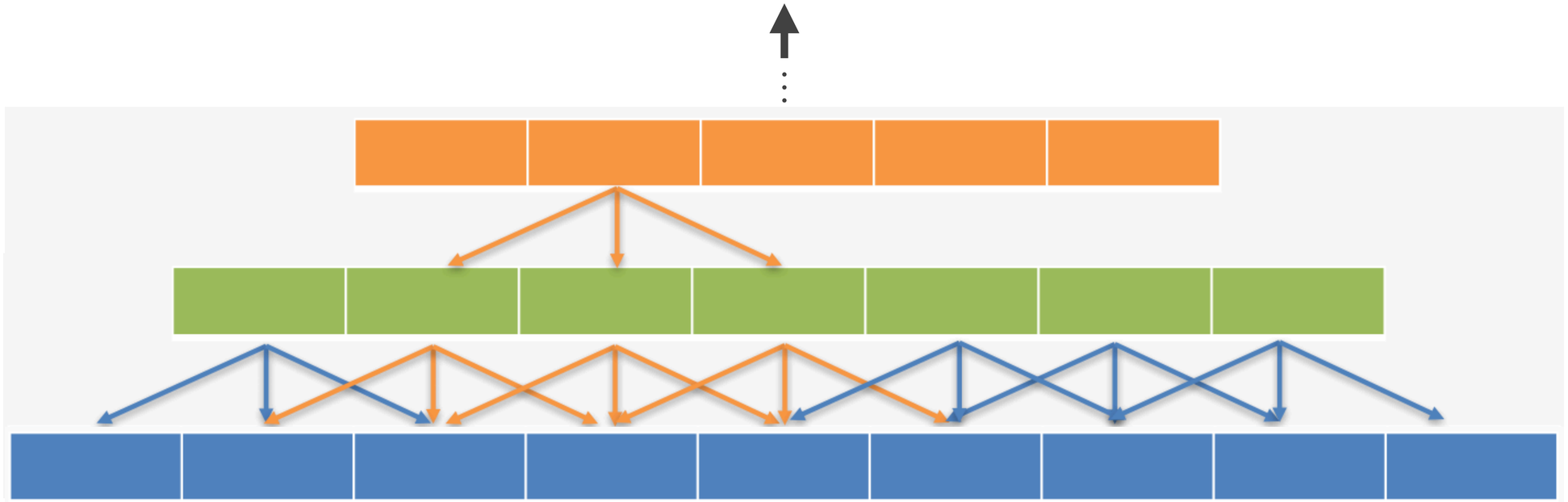


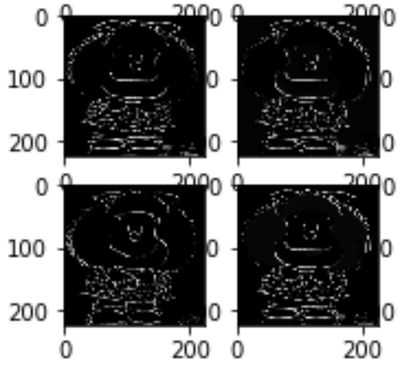
Layer 10

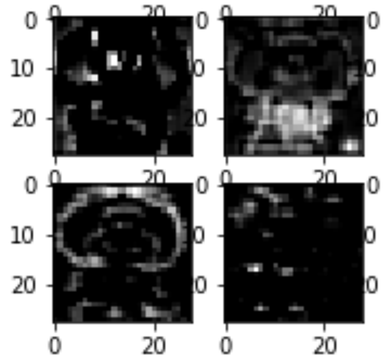


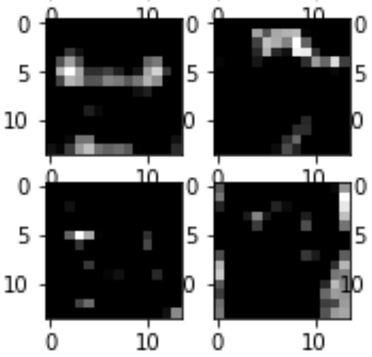
Layer 16

CNNs Layers



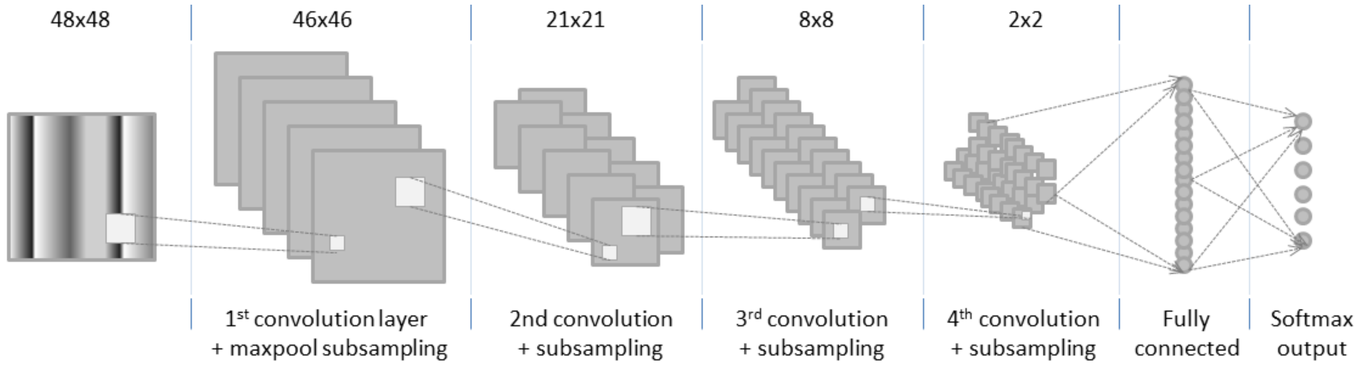




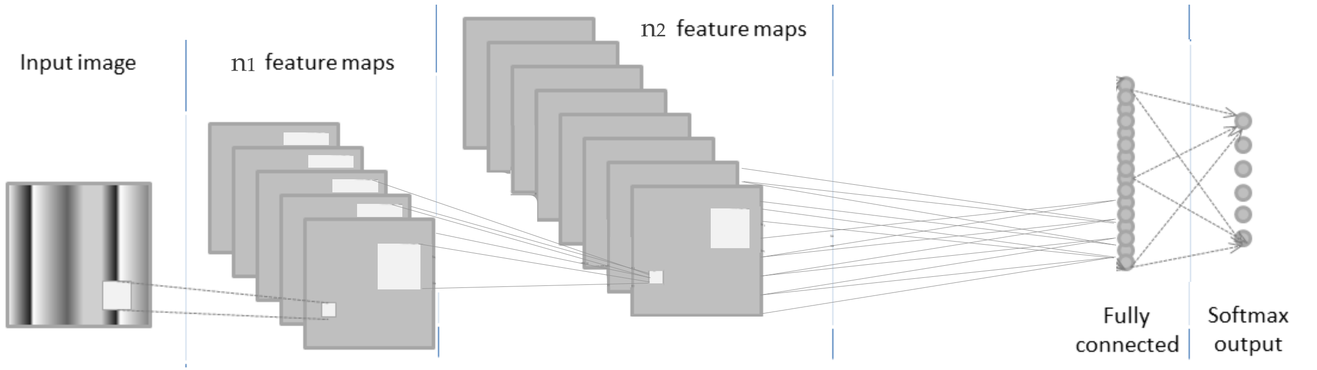


CNNs Layers

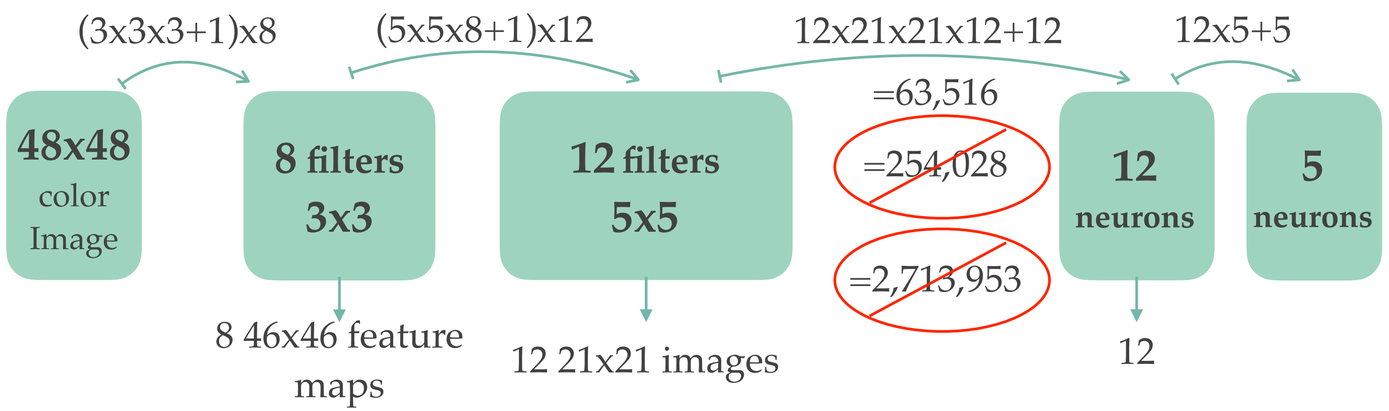
Pooling para reducción dimensional

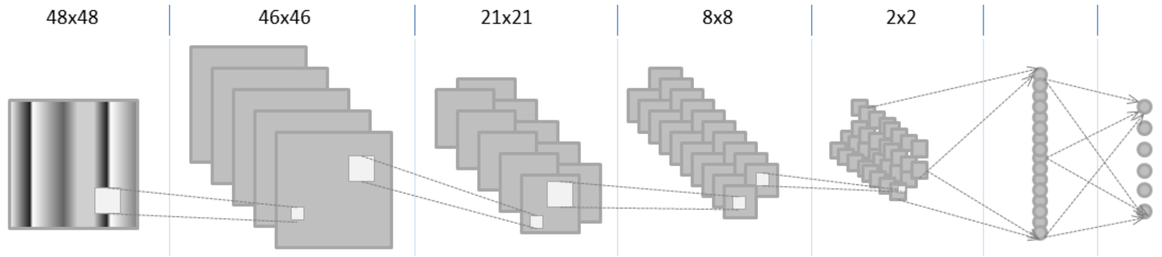


CNNs basics

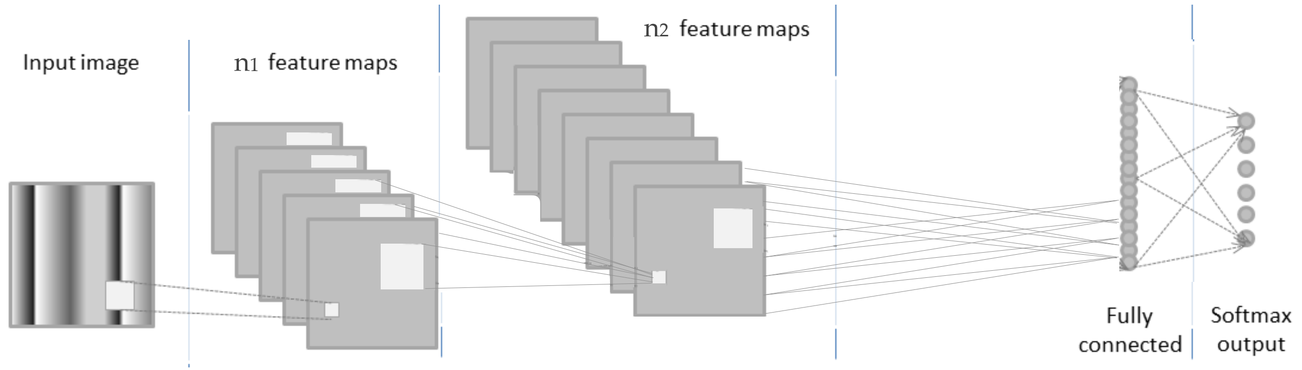


Parameters:

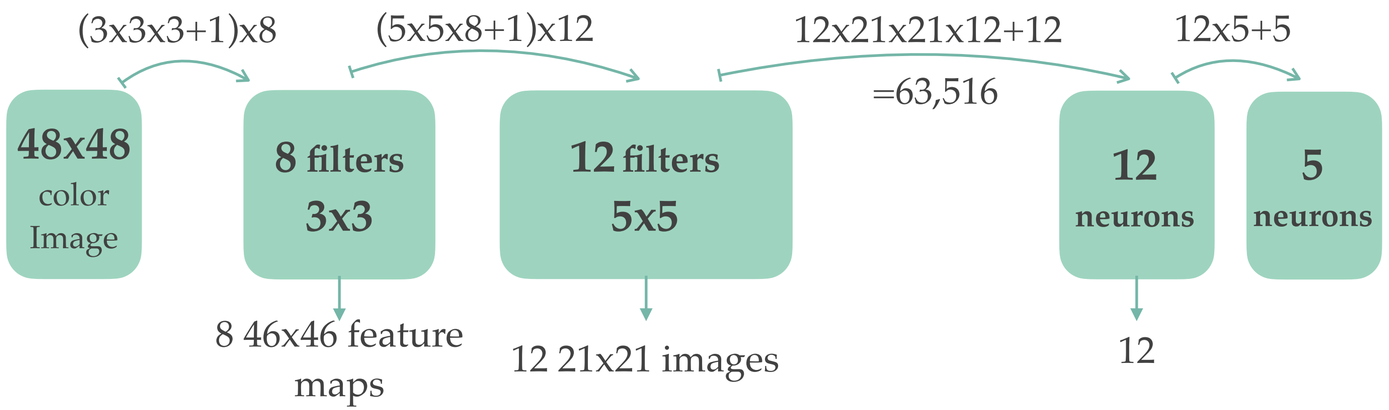




CNNs basics

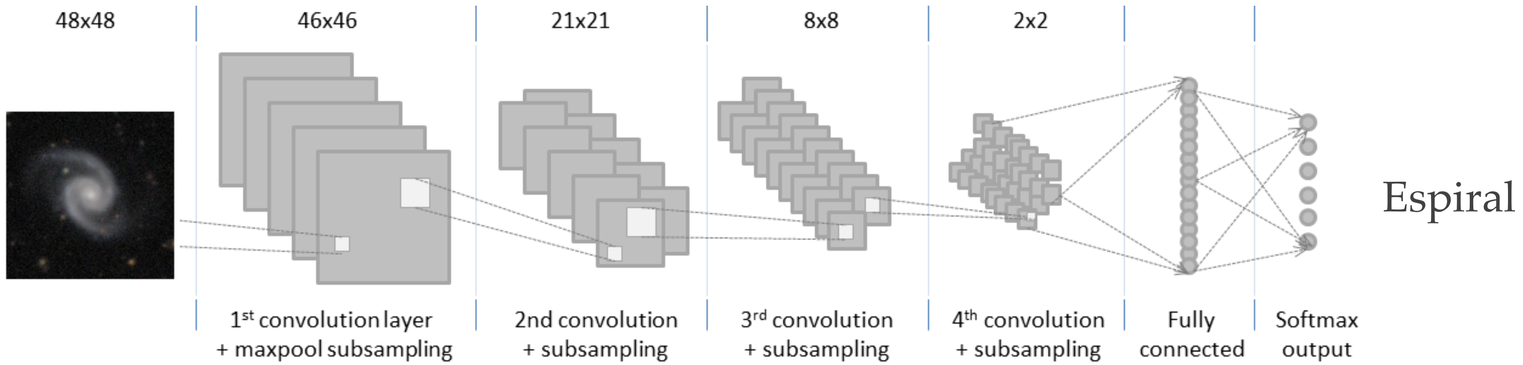


Parameters:



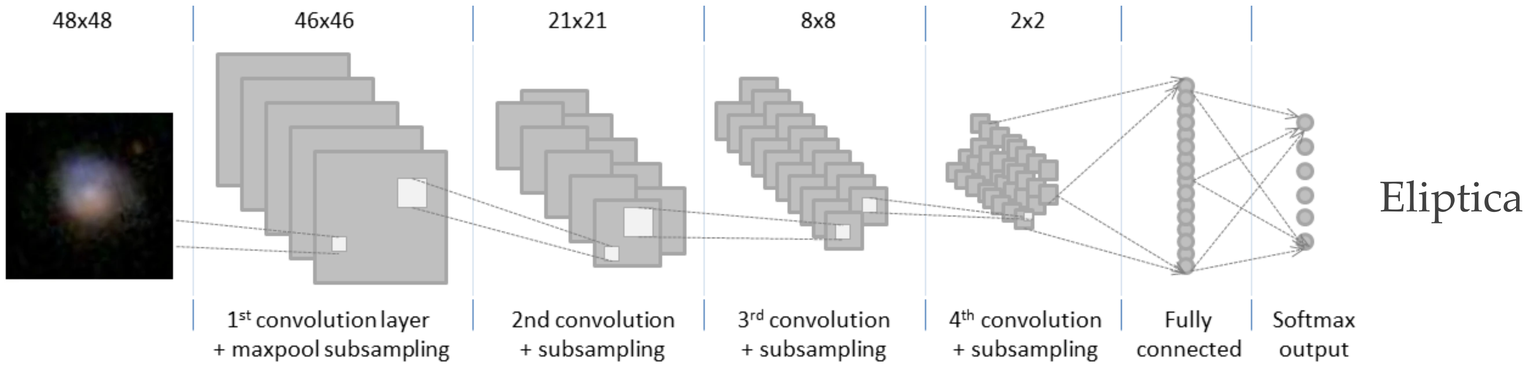
- Redundant, no shared features ✓
- No translation invariance ✓
- Large number of parameter ✓

Interpreting CNNs: Saliency Maps



Espiral

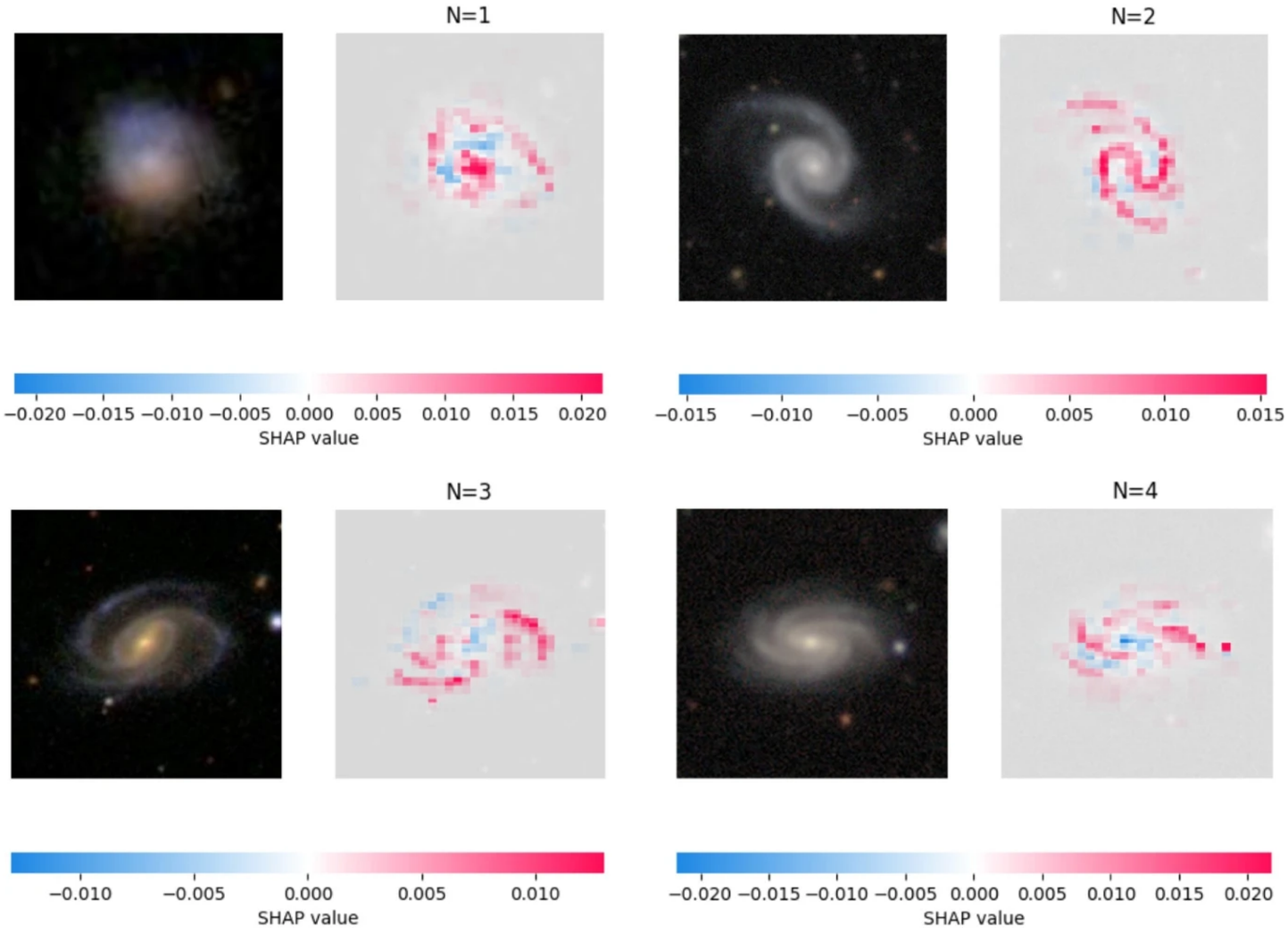
Interpreting CNNs: Saliency Maps



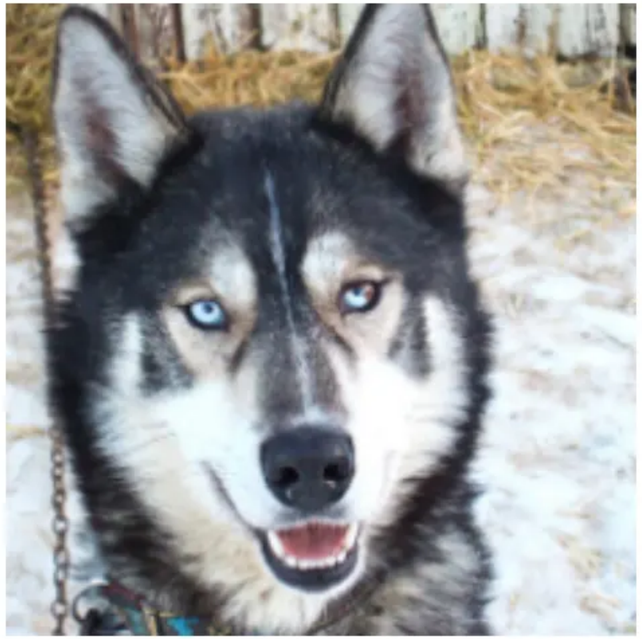
$$\frac{\partial \mathscr{L}(\theta)}{\partial x}$$

$$\mathscr{L}(y, \hat{y})$$

Interpreting CNNs: Saliency Maps



Interpreting CNNs: Saliency Maps



(a) Husky classified as wolf

(b) Explanation

Take Away

- CNNs se basan en MPL pero son robustas para análisis de imágenes: invariancia de traslación
- Características aprendidas y compartidas
- La dimensionalidad del modelo se reduce reduced gracias al aprendizaje de representación y a las características compartidas.

Mañana

- Modelos Semi-supervisados / Aplicaciones en Rayos-X
- Redes Neuronales Fisicamente Informadas (PINNs)
- Redes Neuronales Hamiltonianas (HNNs)
- Redes Neuronales Lagrangianas (LNNs)
- Redes Equivariantes y bias inductivos
- Aplicaciones en Física Teórica (Práctica)

Questions?