DOT NET MLG

30 SEPTIEMBRE 2017

sequel

la térmica
CENTRO DE CULTURA CONTEMPORÁNEA

SIROS Management Solutions

plain concepts

Microsoft

pasiona

Carmen Lupiañez

lemoncode

veum.es

DOT NET MLG

30 SEPTIEMBRE 2017

sequel

DEEP LEARNING CON CNTK

Taller de Visión Artificial

Manuel Rodrigo Cabello Malagón

Software Engineer @ PLAIN CONCEPTS

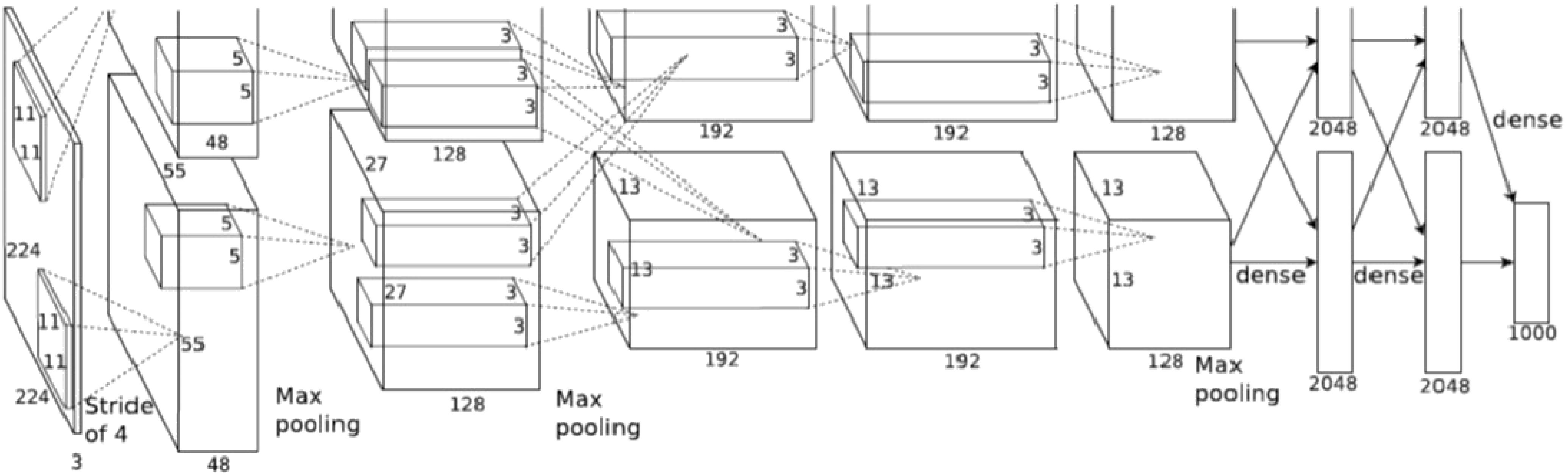
@mrcabellom
mrcabellom@Gmail.com
https://geeks.ms/mrcabellom/

Pablo Doval

DATA PONTIFEX @ PLAIN CONCEPTS

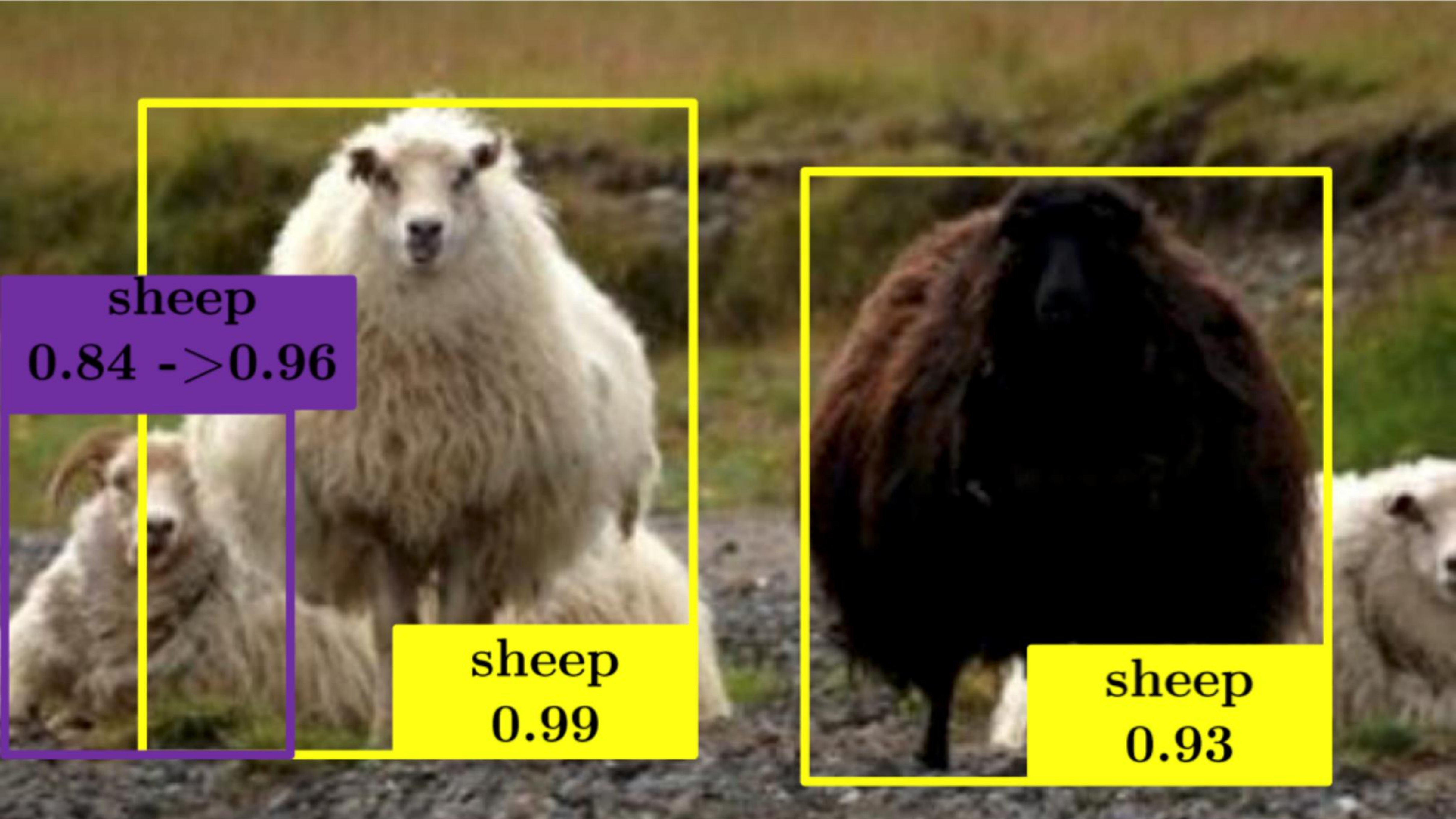
"I work with code and data, but don't tell my mom; she thinks I'm a piano player in a whorehouse."

¿QUÉ ES ESTO?



224
11
11
11
11
224
Stride
of 4
3
5
5
48
55
5
5
55
48
Max
pooling
3
3
3
27
128
3
3
27
3
3
128
Max
pooling
3
3
192
13
13
3
3
192
3
3
192
13
13
3
3
192
128
13
13
128
Max
pooling
dense
dense
2048
2048
2048
2048
dense
1000

¿PARA QUE NOS SIRVE ESTO?


sheep
0.84 ->0.96
sheep
0.99
sheep
0.93

MACHINE LEARNING EN AZURE

Azure ML

- ~~Fácil y sencillo~~
- ~~Escalabilidad limitada~~
- Version Renovada hace 4 dias ;)

R Server

- Lenguaje conocido
- Escalado a grandes datasets
- Evaluaciones en tiempo real

HDInsight

- Mahout
- SparkML
- Escalabilidad casi ilimitada

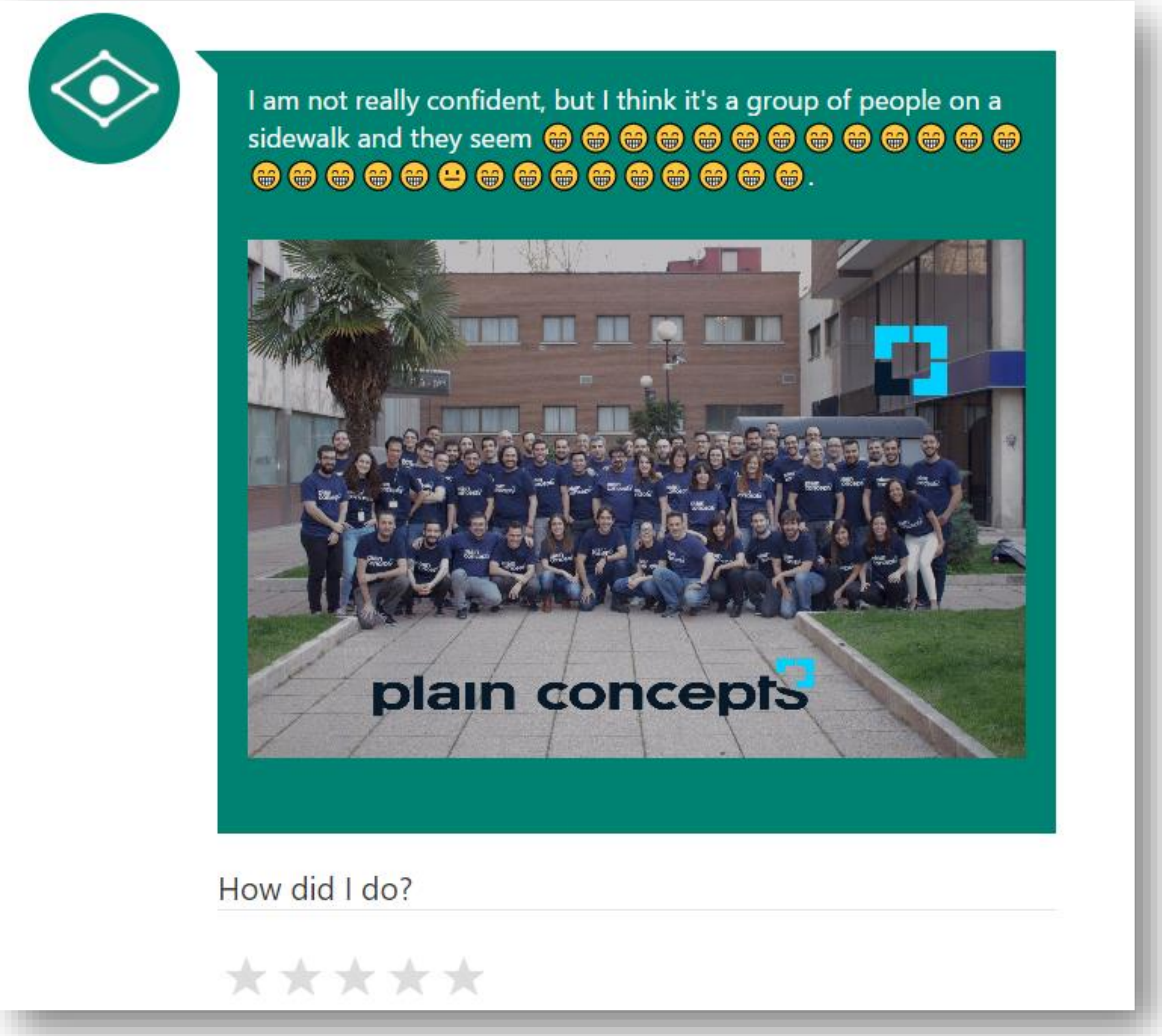
Cognitive Services

- APIs listas para usar
- Modelos pre-entrenados
- No son necesarios conocimientos de ML

CNTK

- Modelos complejos
- Entrenamiento personalizado
- Entrenamiento multi server/multi GPU

¿CUANDO NO USAR COGNITIVE SERVICES?



I am not really confident, but I think it's a group of people on a sidewalk and they seem 😁😁😁😁😁😁😁😁😁😁😁😁😁😁😁😁😁😁😐😁😁😁😁😁😁😁😁😁.

How did I do?

★★★★★

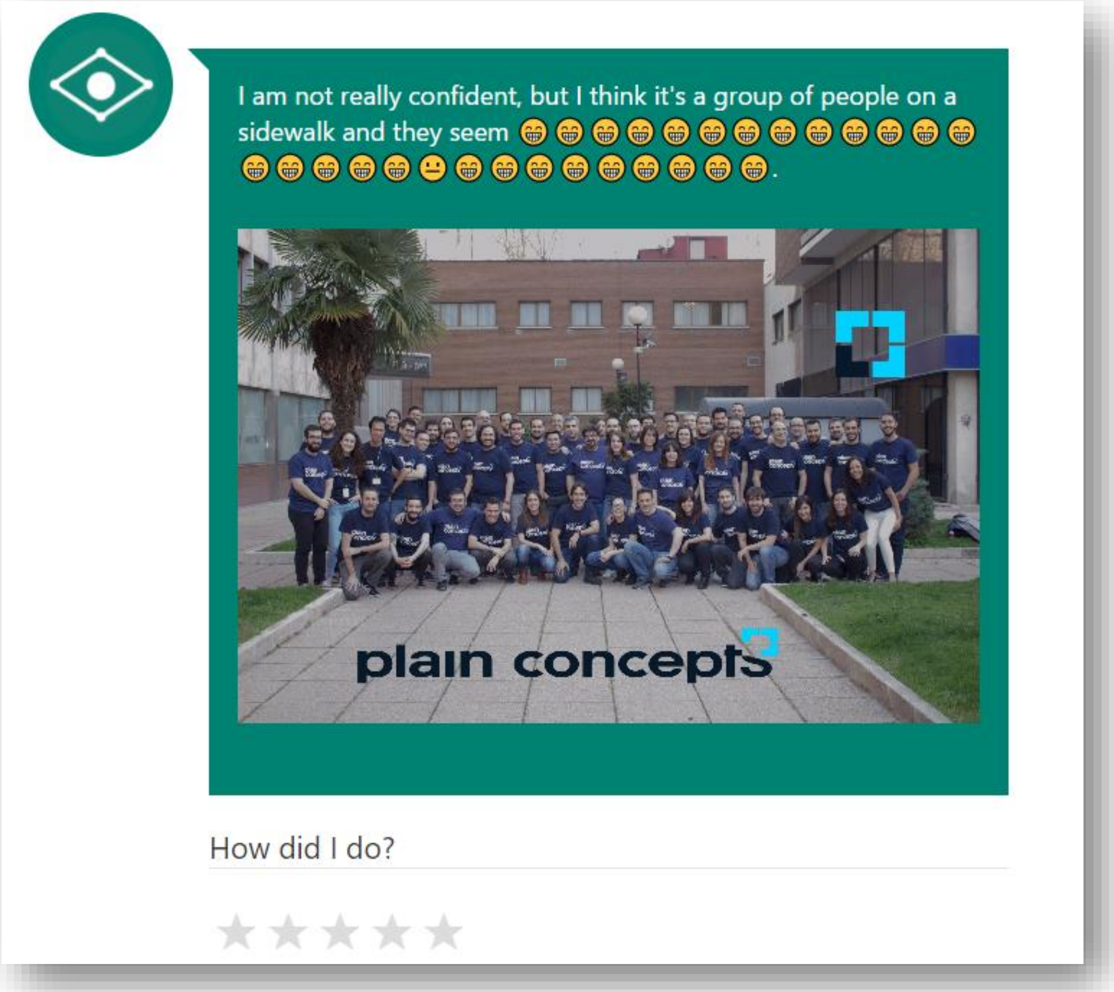
Vision

Speech

Language

Knowledge

Search



I am not really confident, but I think it's a group of people on a sidewalk and they seem 😁😁😁😁😁😁😁😁😁😁😁😁😁😁😁😁😁😁😐😁😁😁😁😁😁😁😁😁.

How did I do?

★★★★★

Vision

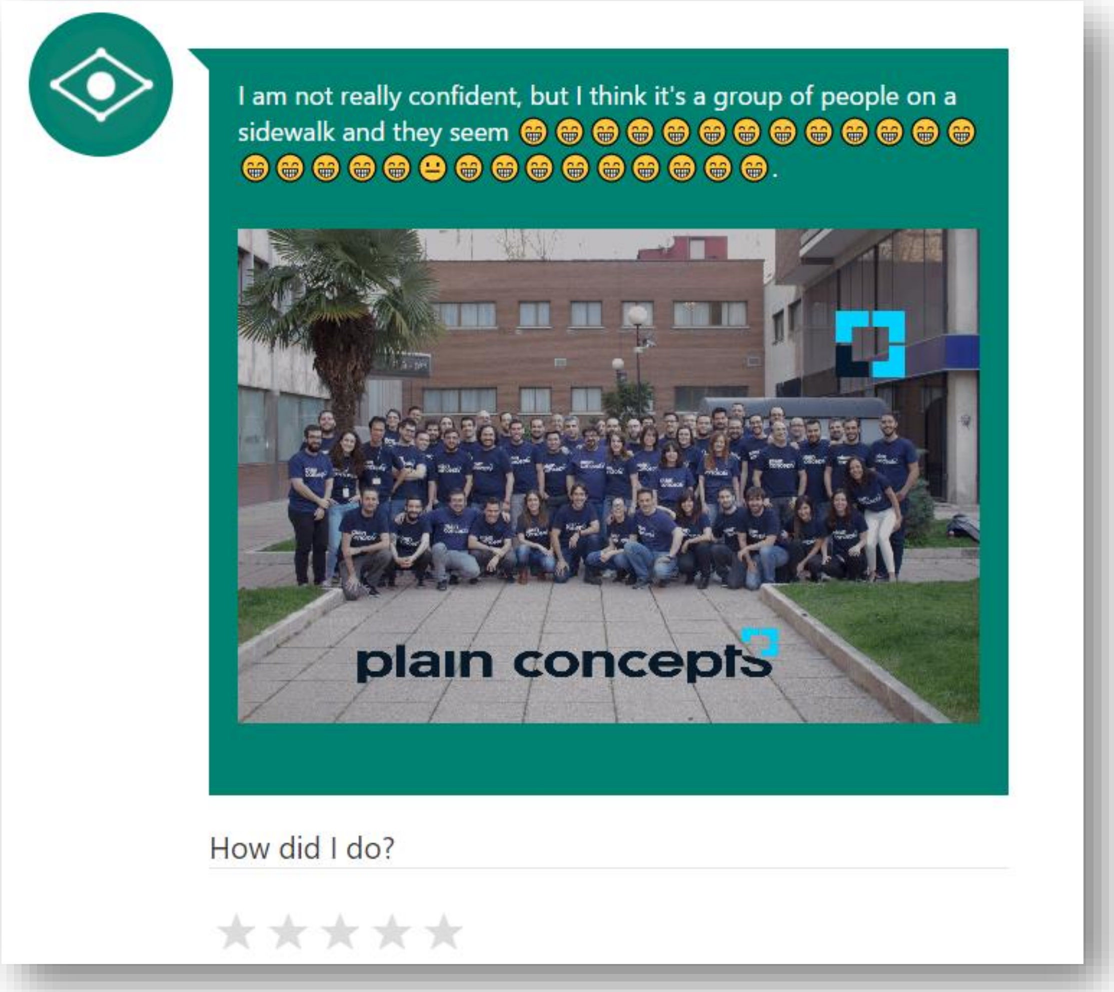
- Computer Vision
- Content Moderator
- Emotion API
- Face API
- …

Speech

Language

Knowledge

Search



I am not really confident, but I think it's a group of people on a sidewalk and they seem 😁😁😁😁😁😁😁😁😁😁😁😁😁😁😁😁😁😁😐😁😁😁😁😁😁😁😁😁.

How did I do?

★★★★★

Vision

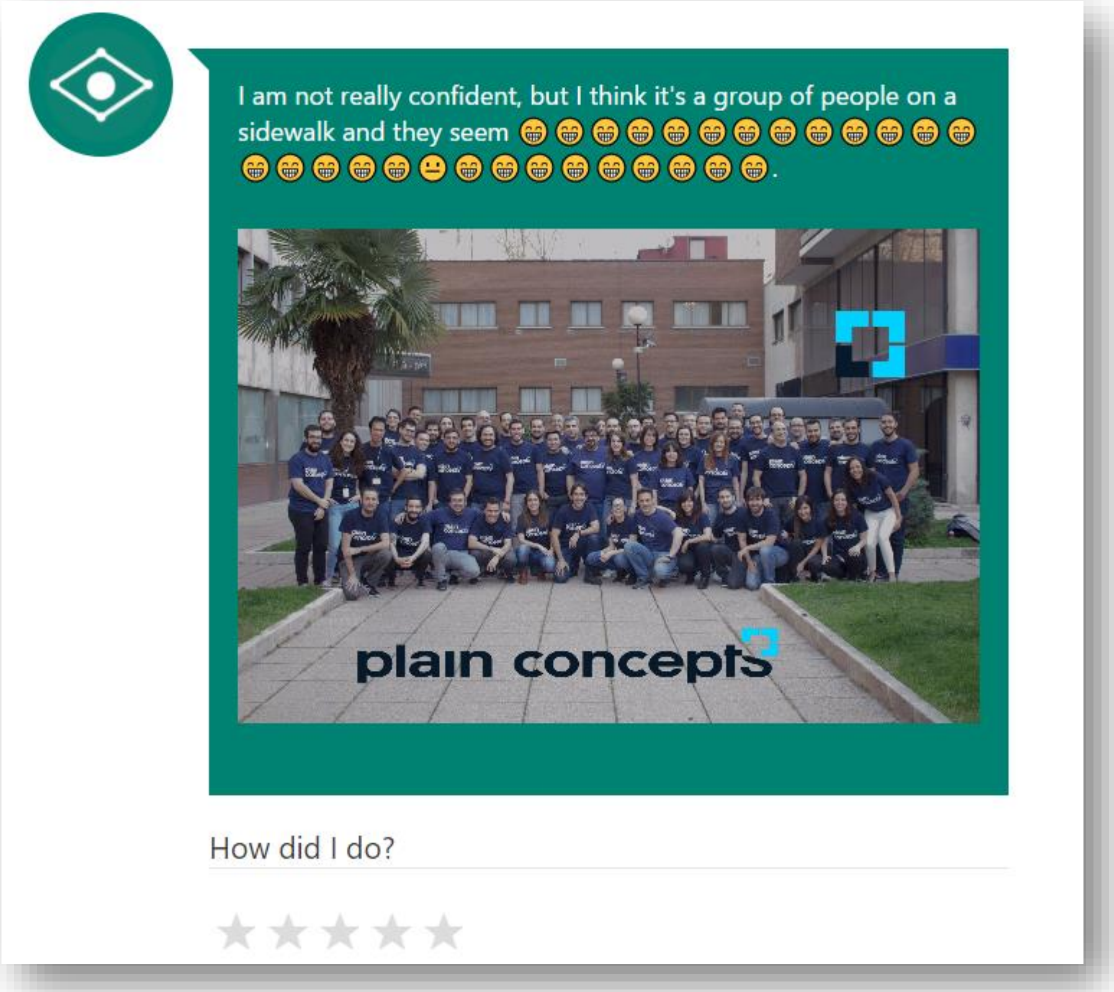
Speech

- Bing Speech API
- Custom Speech Service
- Speaker Recognition Service

Language

Knowledge

Search



I am not really confident, but I think it's a group of people on a sidewalk and they seem 😁😁😁😁😁😁😁😁😁😁😁😁😁😁😁😁😁😁😐😁😁😁😁😁😁😁😁😁.

How did I do?

★★★★★

Vision

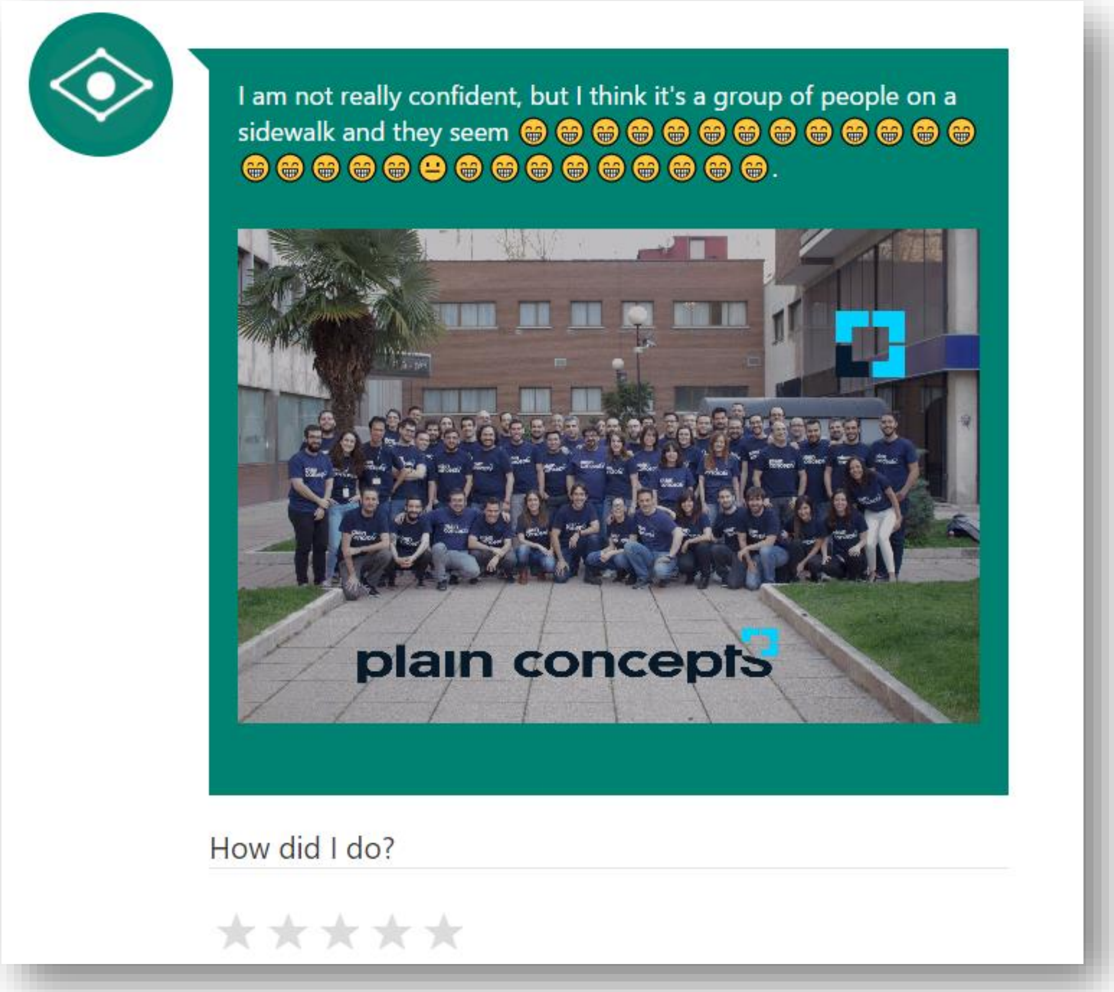
Speech

Language

- Spell Check API
- LUIS
- Linguistic Analysis
- Translator API
- …

Knowledge

Search



I am not really confident, but I think it's a group of people on a sidewalk and they seem 😁😁😁😁😁😁😁😁😁😁😁😁😁😁😁😁😁😁😐😁😁😁😁😁😁😁😁😁.

How did I do?

☆☆☆☆☆

Vision

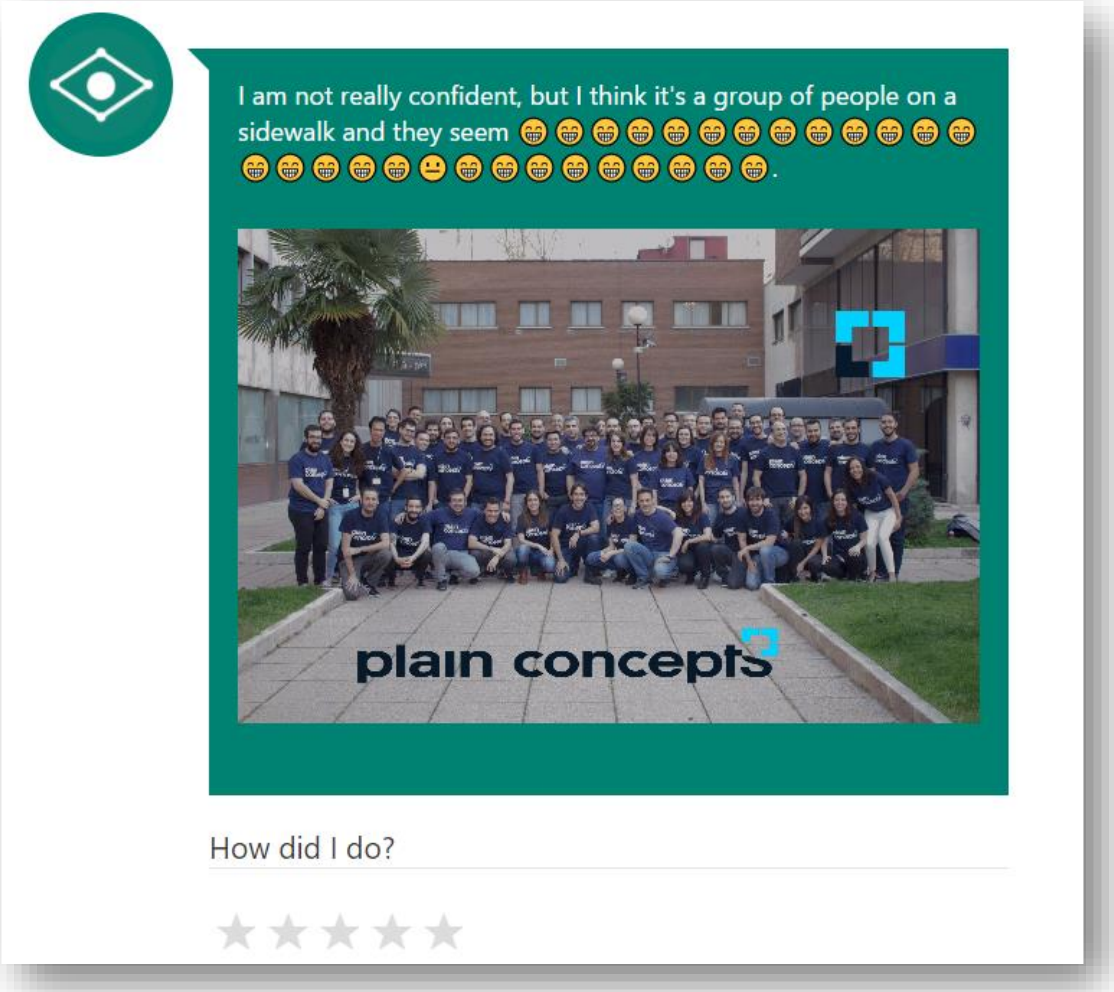
Speech

Language

Knowledge

- Entity Linking Service
- Academic Knowledge Service
- Q&A Maker
- …

Search



I am not really confident, but I think it's a group of people on a sidewalk and they seem 😁😁😁😁😁😁😁😁😁😁😁😁😁😁😁😁😁😁😐😁😁😁😁😁😁😁😁😁.

How did I do?

★★★★★

Vision

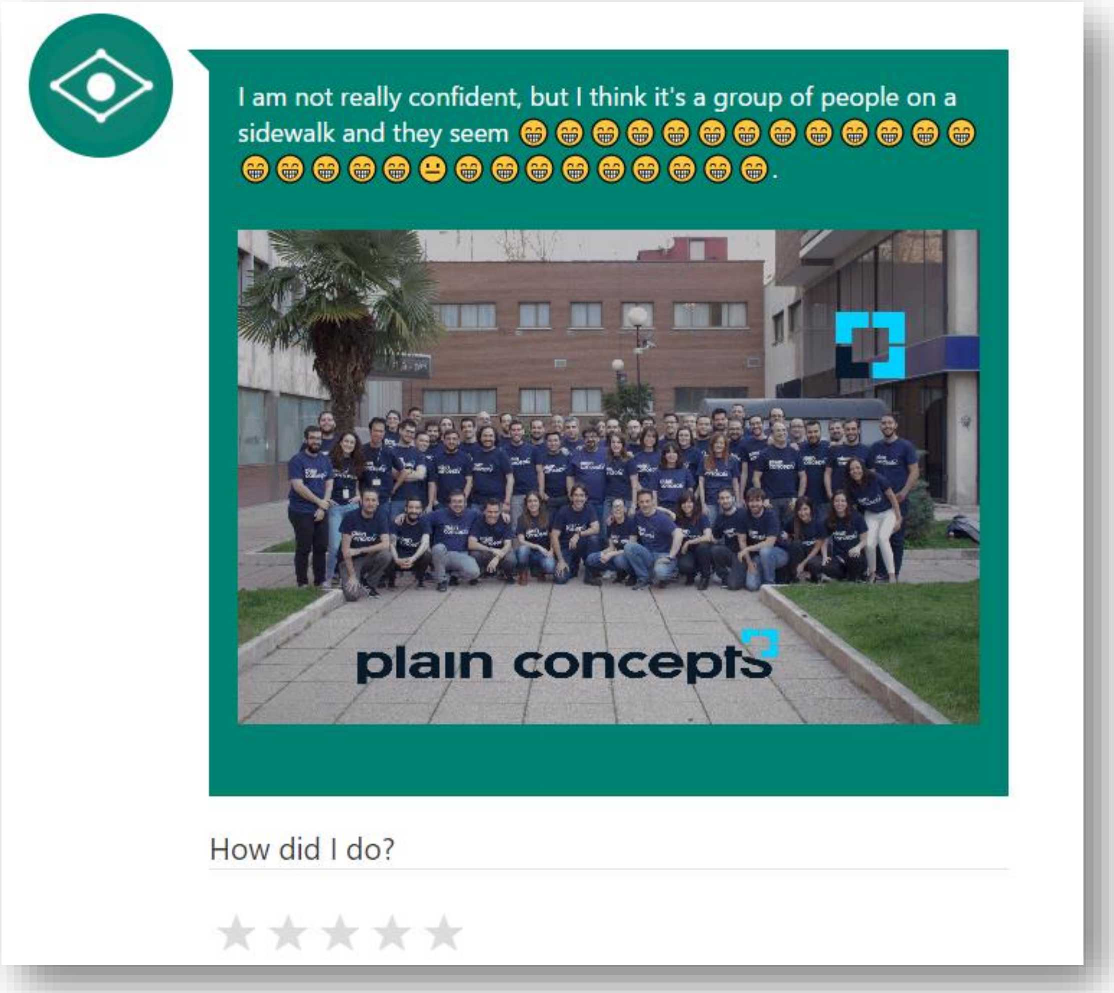
Speech

Language

Knowledge

Search

- Autosuggest API
- Image Search API
- News Search API
- …



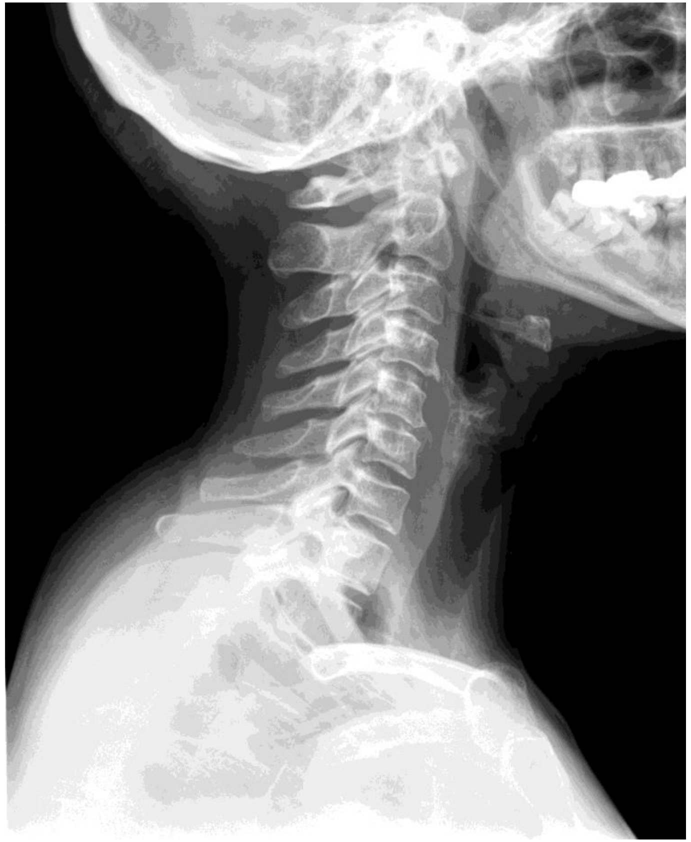
I am not really confident, but I think it's a group of people on a sidewalk and they seem 😁😁😁😁😁😁😁😁😁😁😁😁😁
😁😁😁😁😁😐😁😁😁😁😁😁😁😁😁.

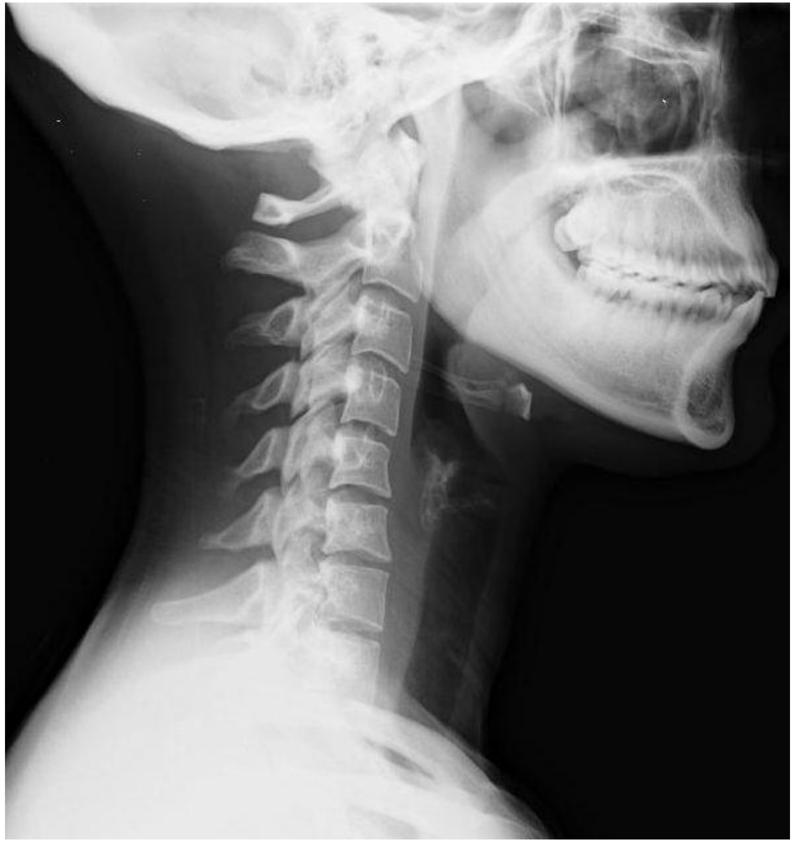
How did I do?

★★★★★

¿CUANDO **NO** USAR COGNITIVE SERVICES?

¿CUANDO **NO** USAR COGNITIVE SERVICES?





¿CUANDO **NO** USAR COGNITIVE SERVICES?

¿CUANDO **NO** USAR COGNITIVE SERVICES?

DEEP LEARNING CON CNTK

DEEP LEARNING CON CNTK

DEEP LEARNING

Machine Learning

Basados en Redes Neuronales

Más de una capa oculta (*Deep Models*)

DEEP LEARNING CON CNTK

CNTK

¿Por qué?

- Desarrollo más rápido de los modelos
- Uso de múltiples CPUs y GPUs
- Multi-plataforma
- Es el producto que Microsoft usa internamente

CNTK

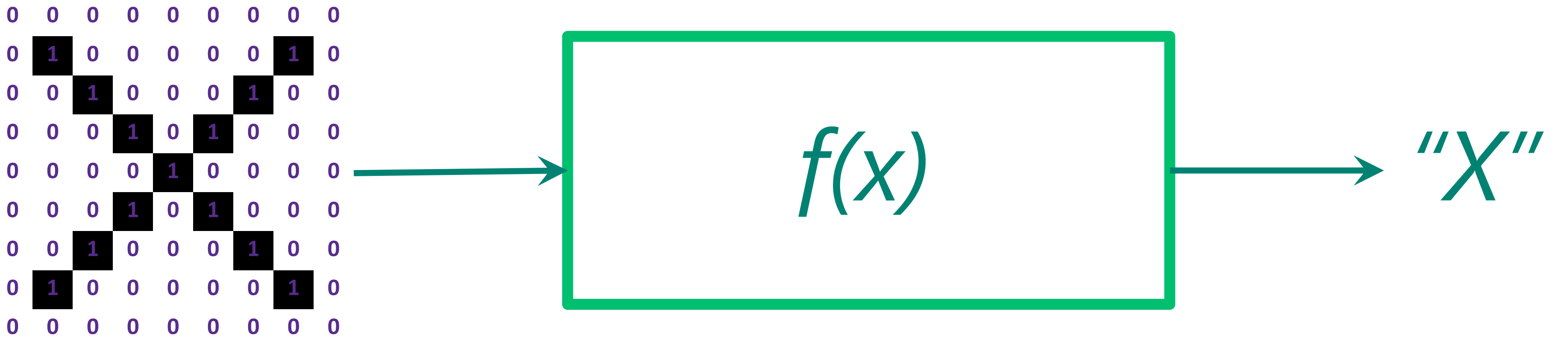
- Es el acrónimo del algo.
- Es un framework de Deep Learning, y es open-source, con contribuciones externas relevantes del MIT y Stanford, entre otros.
- Permite expresar redes neuronales complejas de un modo relativamente sencillo, y se encarga de todos los pasos de su ciclo de vida: desde el entrenamiento hasta la evaluación.
- Desarrollo en BrainScript, Python (.NET en camino)

LAB 0: INSTALACIÓN

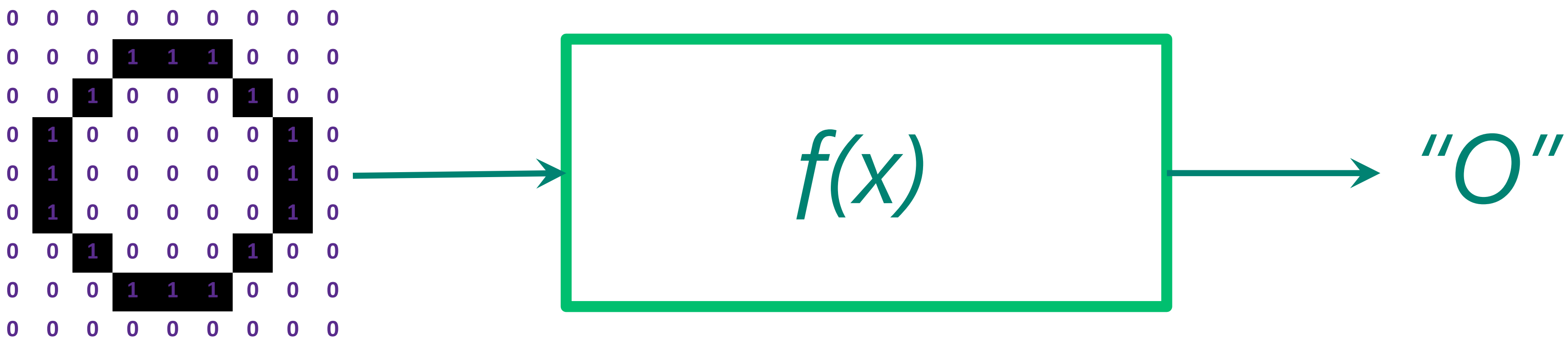
TÉCNICAS DE VISIÓN ARTIFICIAL

Regresión Logística

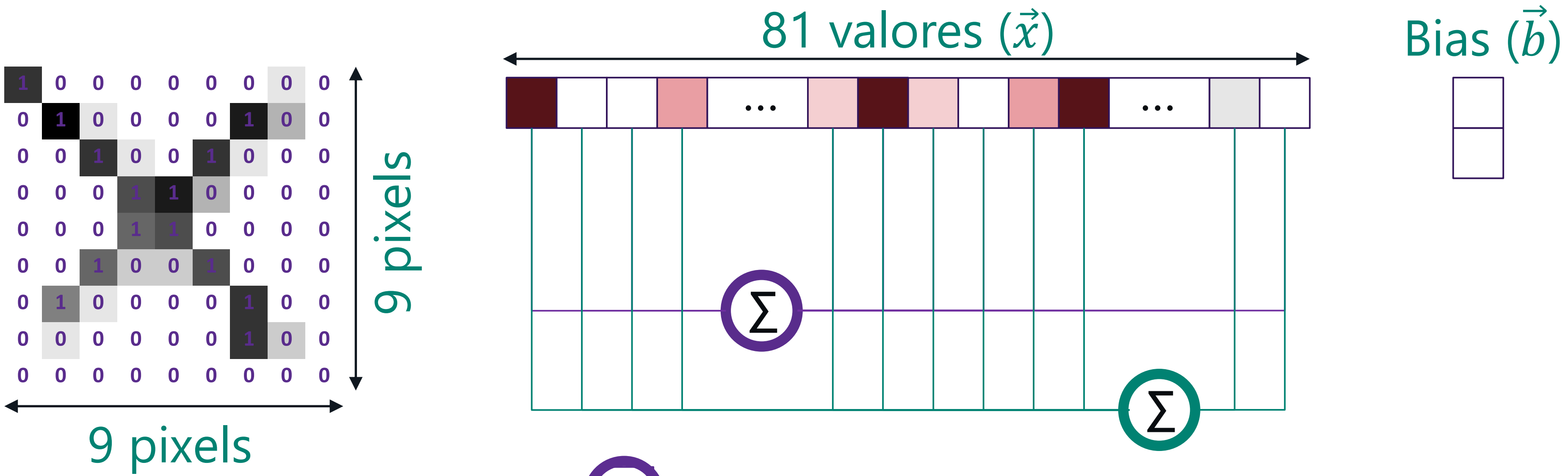
UN EJEMPLO



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



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



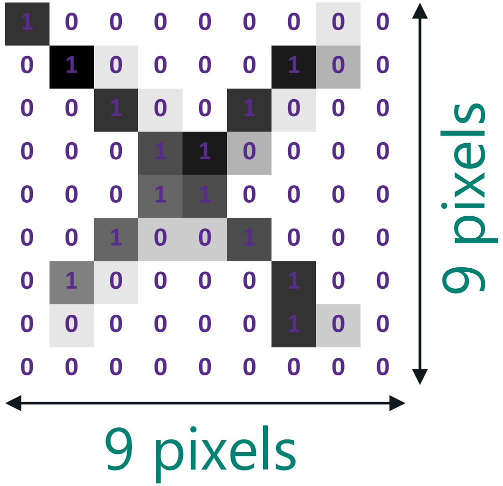
1	0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	0	0
0	0	1	0	0	1	0	0	0
0	0	0	1	1	0	0	0	0
0	0	0	1	1	0	0	0	0
0	0	1	0	0	1	0	0	0
0	1	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0

Σ *Sum (weights x pixels)* = $\vec{w_x} \cdot \vec{x}$

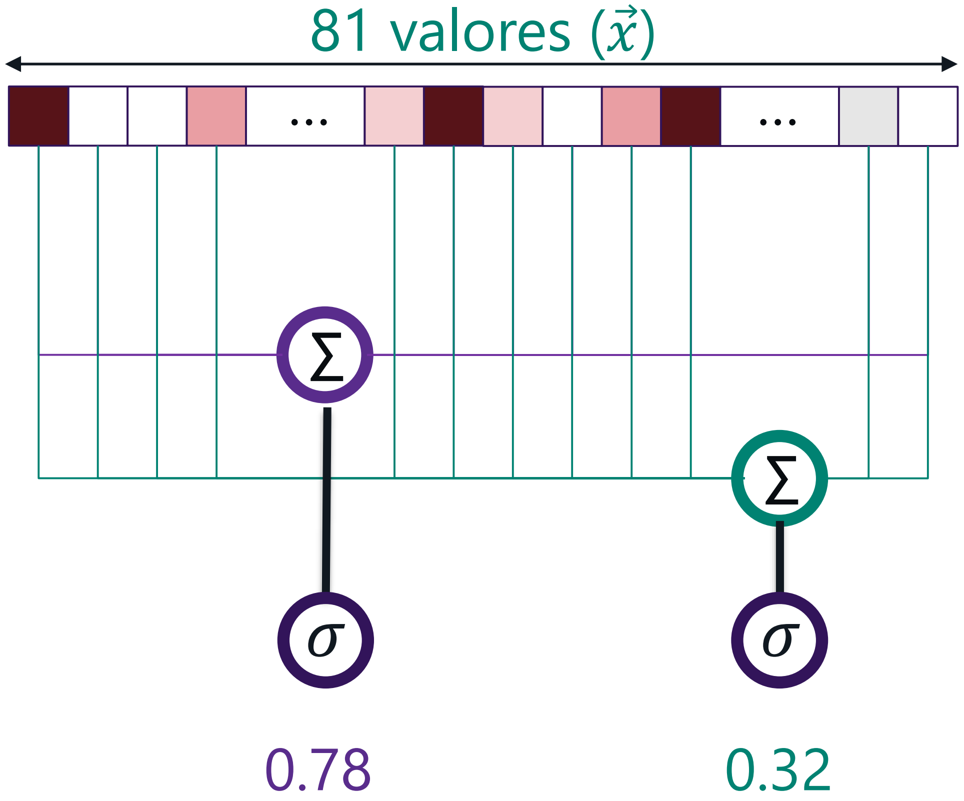
Σ *Sum (weights x pixels)* = $\vec{w_O} \cdot \vec{x}$

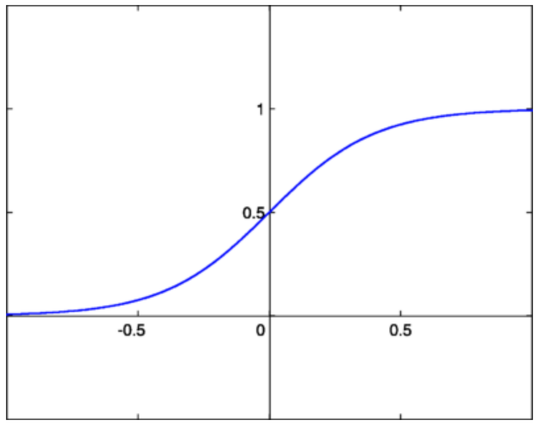
$$z = \vec{w} \cdot \vec{x} + b$$

SIGMOIDE

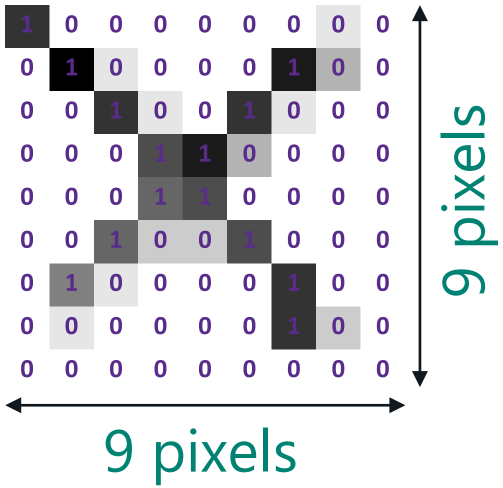


1	0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	0	0
0	0	1	0	0	1	0	0	0
0	0	0	1	1	0	0	0	0
0	0	0	1	1	0	0	0	0
0	0	1	0	0	1	0	0	0
0	1	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0

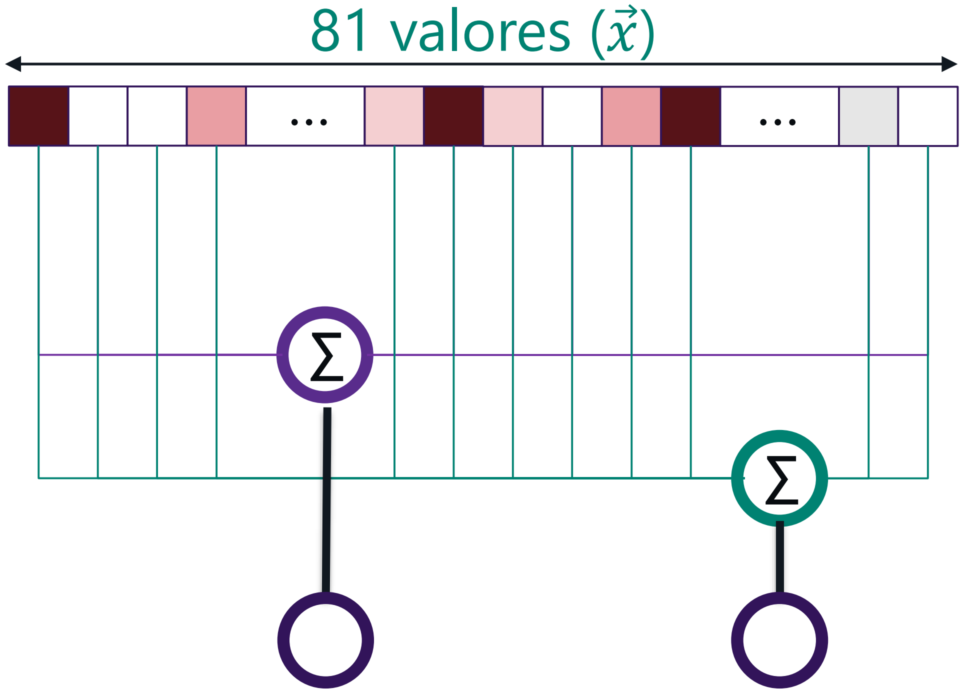




SOFTMAX



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

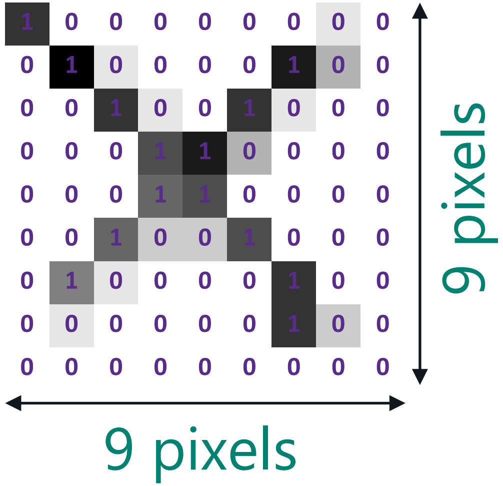


$$s_{(y_i)} = \frac{e^{y_i}}{\sum_j e^{y_i}}$$

0.75 0.25

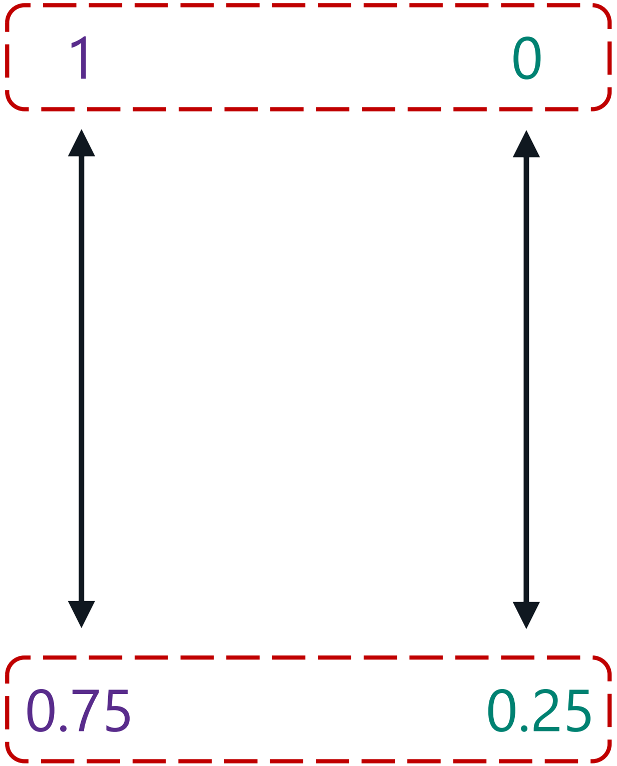
Esto ya son probabilidades

FUNCIÓN DE PERDIDA

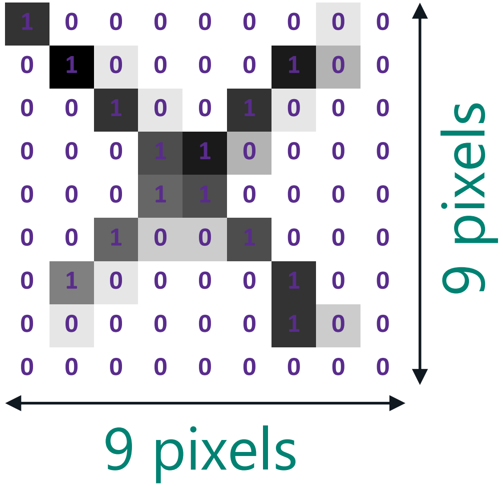


1	0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	0	0
0	0	1	0	0	1	0	0	0
0	0	0	1	1	0	0	0	0
0	0	0	1	1	0	0	0	0
0	0	1	0	0	1	0	0	0
0	1	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0

One-hot encoding



FUNCIÓN DE PERDIDA



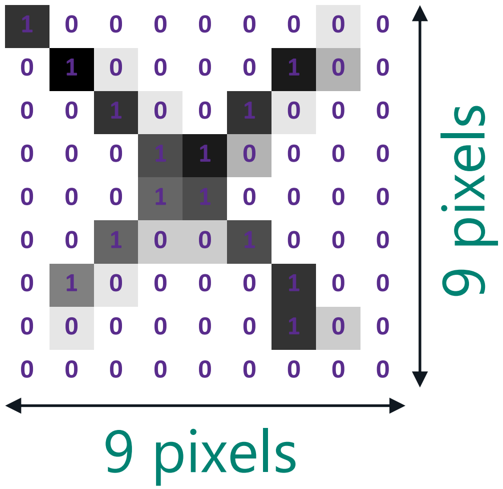
1	0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	0	0
0	0	1	0	0	1	0	0	0
0	0	0	1	1	0	0	0	0
0	0	0	1	1	0	0	0	0
0	0	1	0	0	1	0	0	0
0	1	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0

1 0

$$\text{error} = \frac{1}{n}\sum_{i=1}^{n}\left(y_j - p_j\right)^2$$

0.75 0.25

CROSS ENTROPY



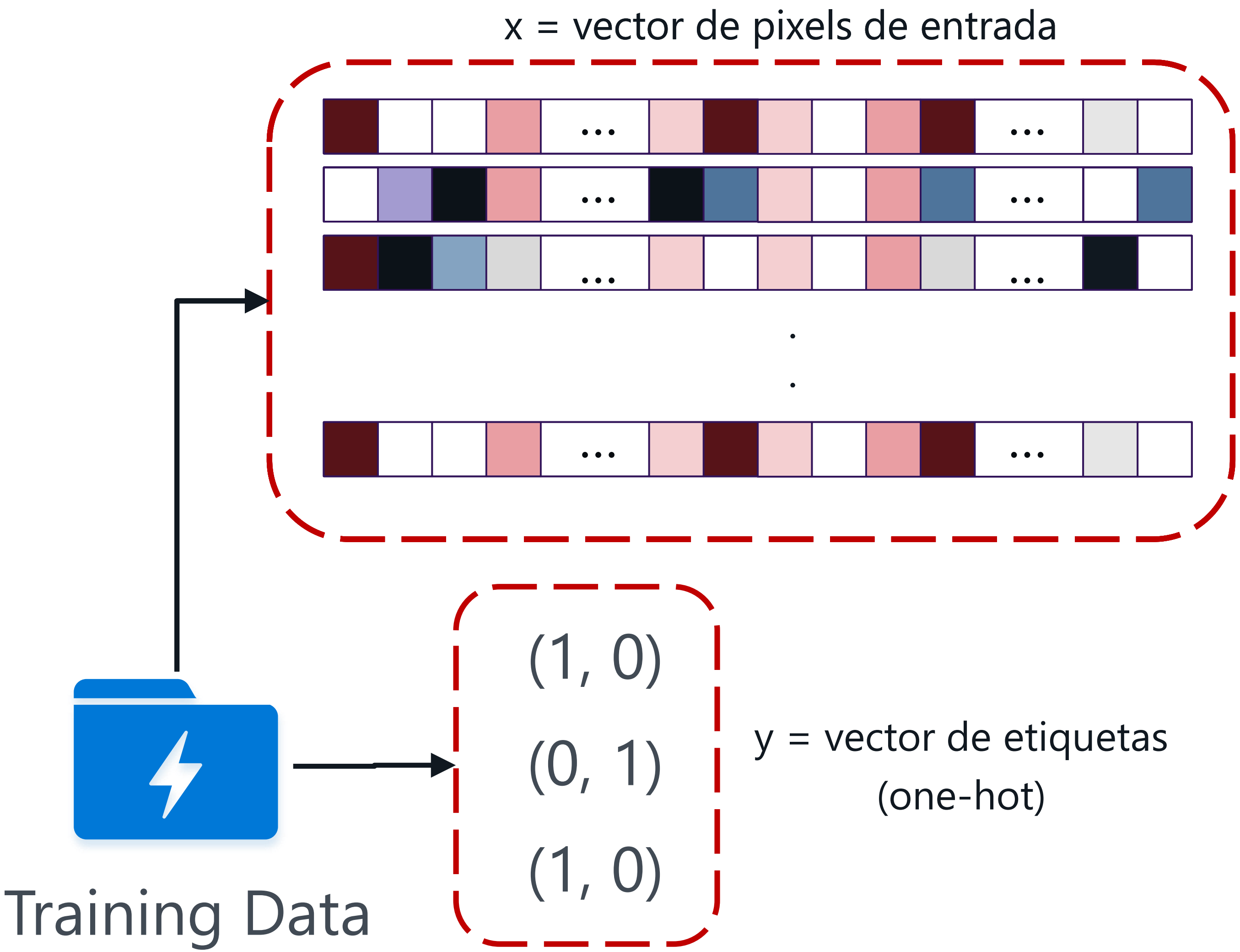
1	0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	0	0
0	0	1	0	0	1	0	0	0
0	0	0	1	1	0	0	0	0
0	0	0	1	1	0	0	0	0
0	0	1	0	0	1	0	0	0
0	1	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0

1 0

$$\text{error} = {y_j}^T \log\ P_j$$

0.75 0.25

ENTRENAMIENTO



x = vector de pixels de entrada

(1, 0)

(0, 1)

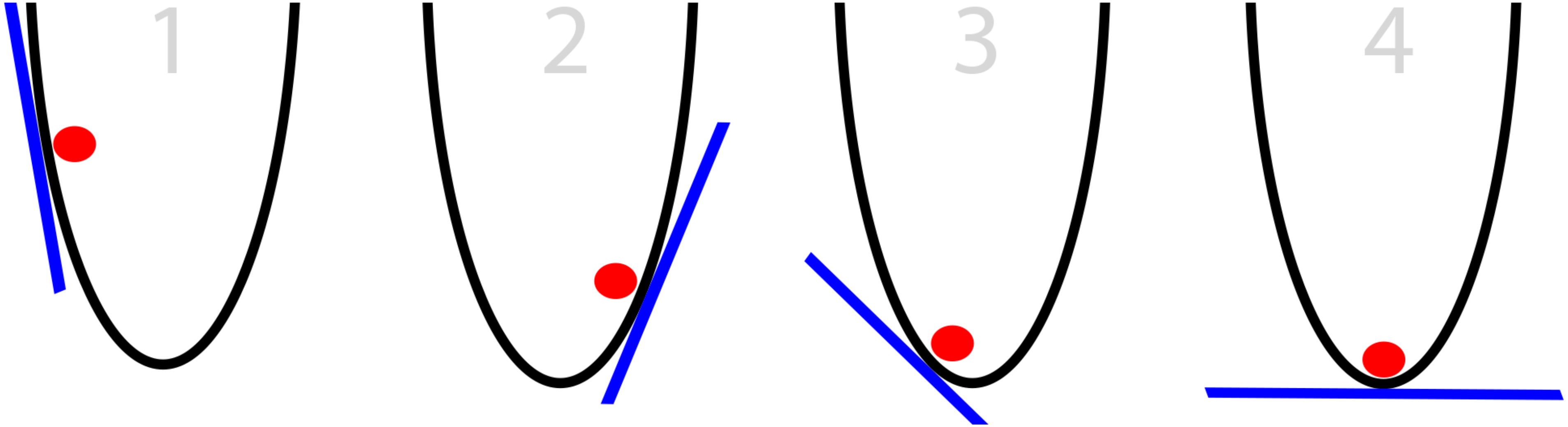
(1, 0)

y = vector de etiquetas (one-hot)

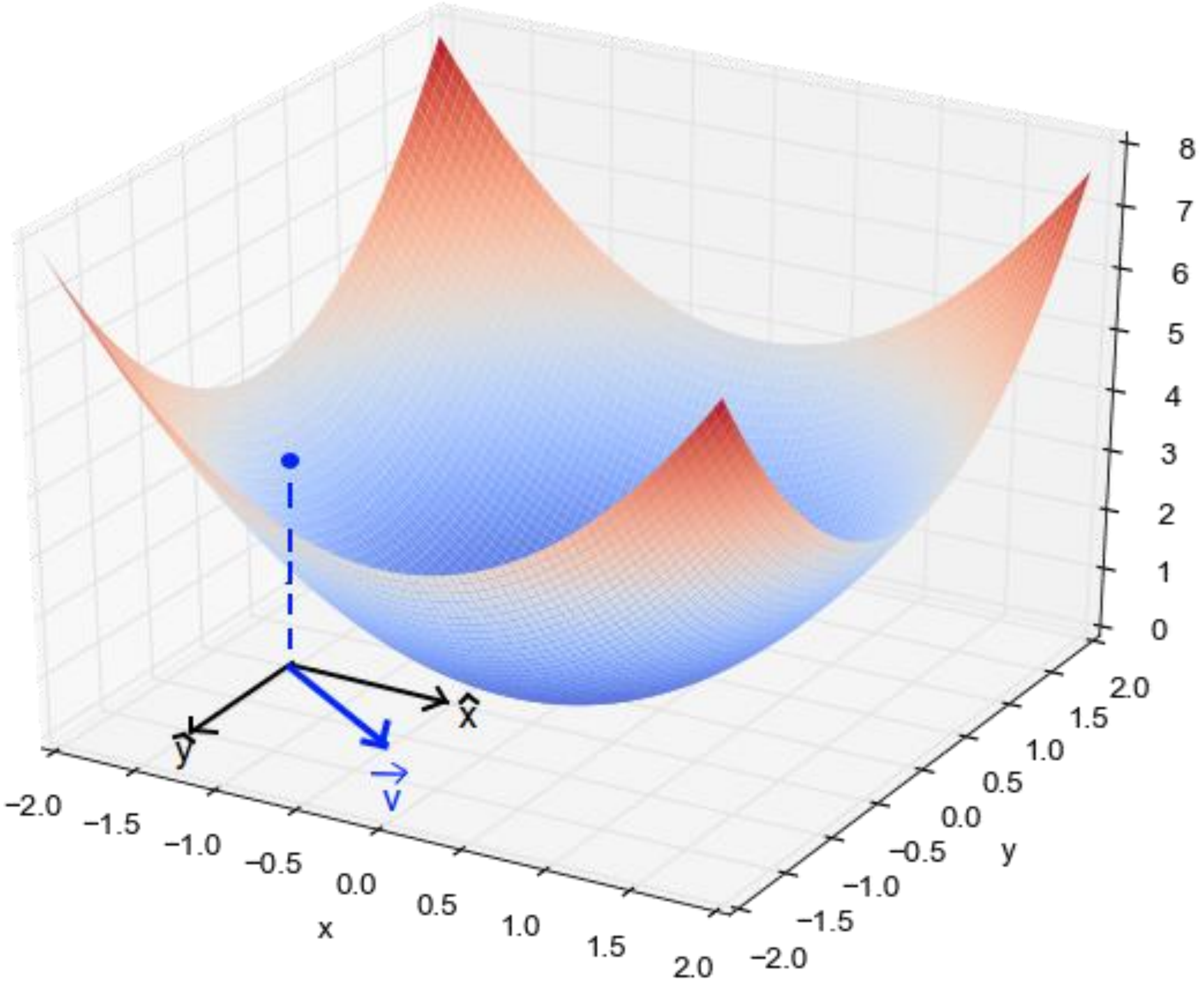
Training Data

```
z = times(x, W) + b
cross_entropy_with_softmax(z, y)
classification_error(z, y)
```

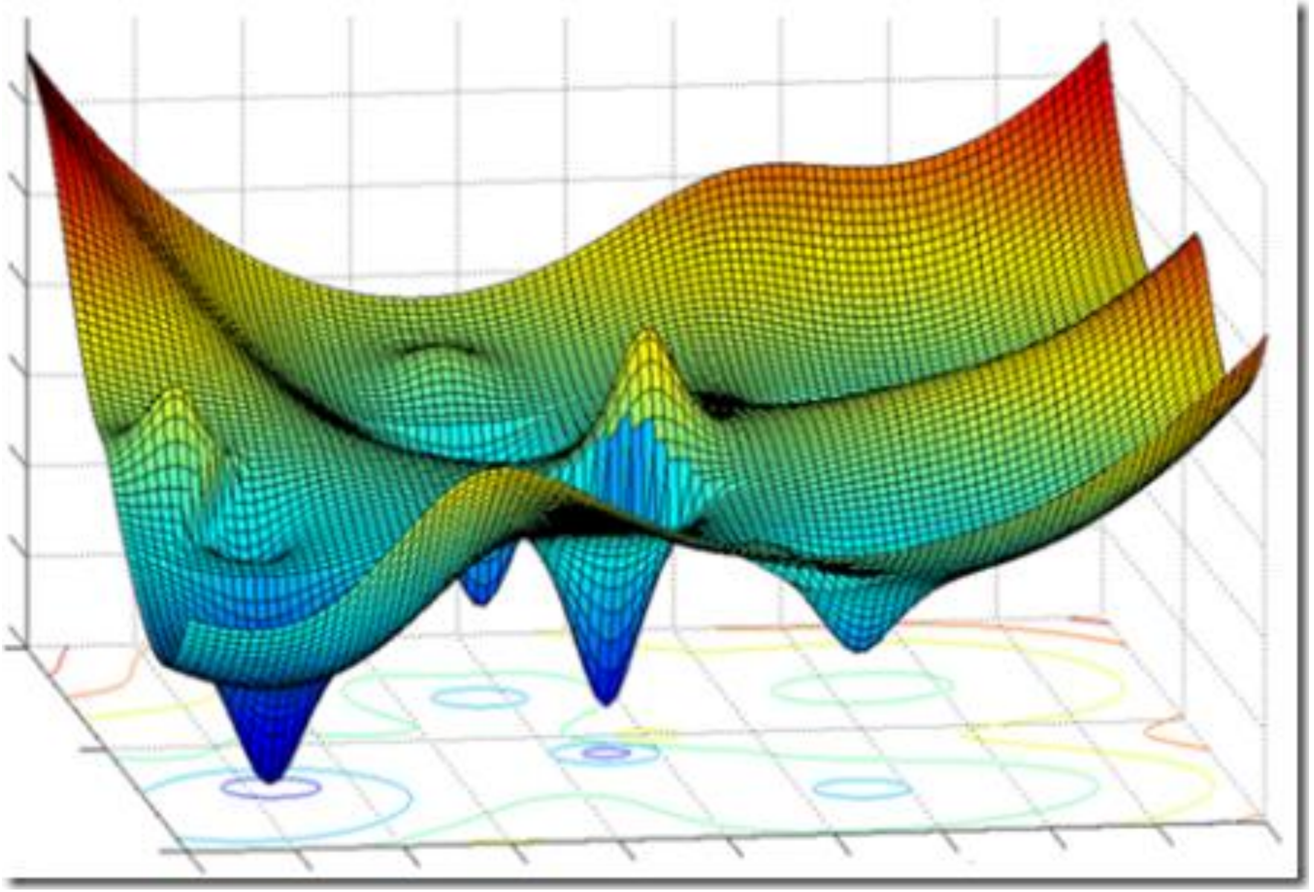
DESCENSO DE GRADIENTE



DESCENSO DE GRADIENTE



DESCENSO DE GRADIENTE

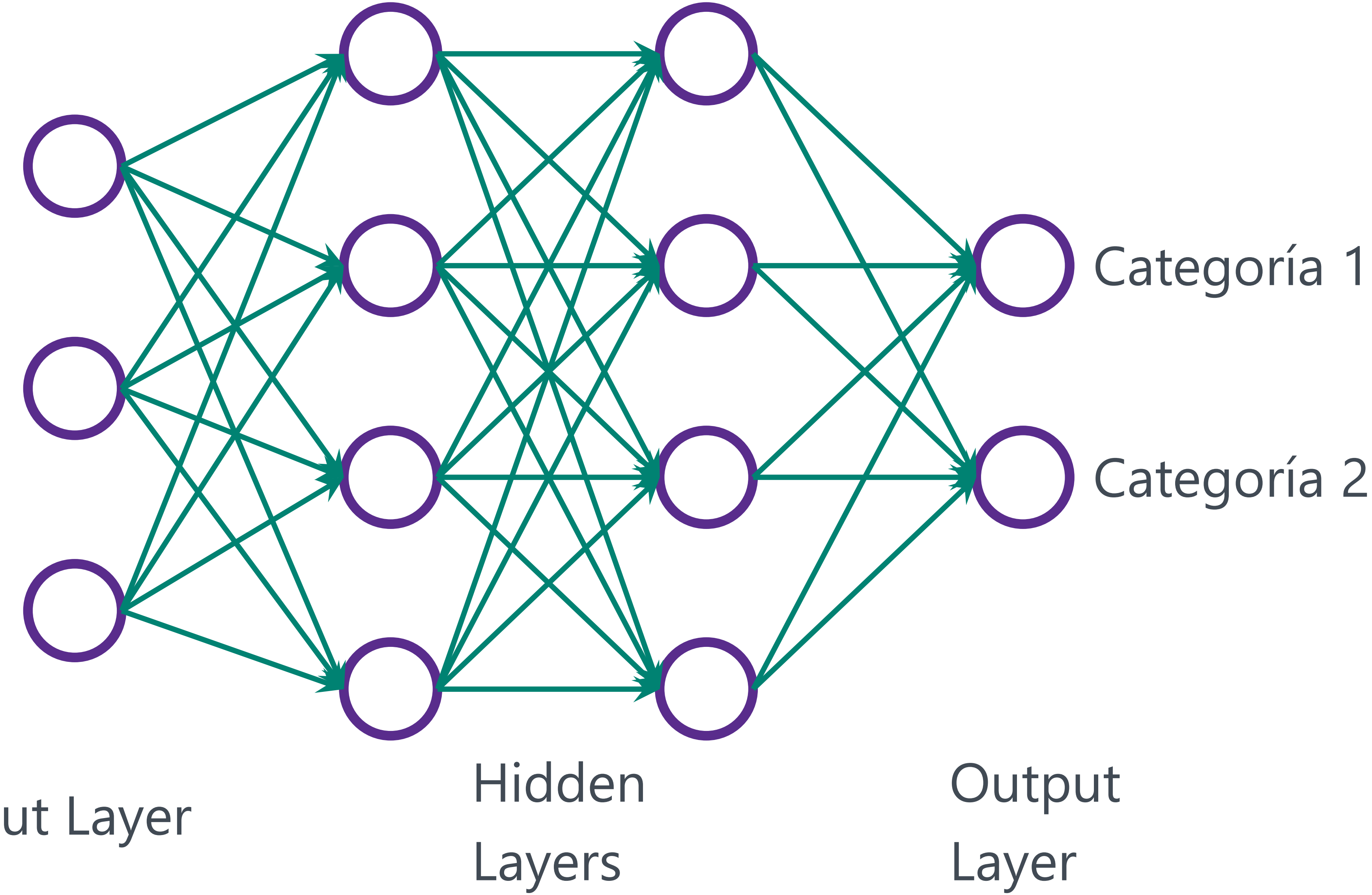


LAB 1: REGRESIÓN LOGÍSTICA

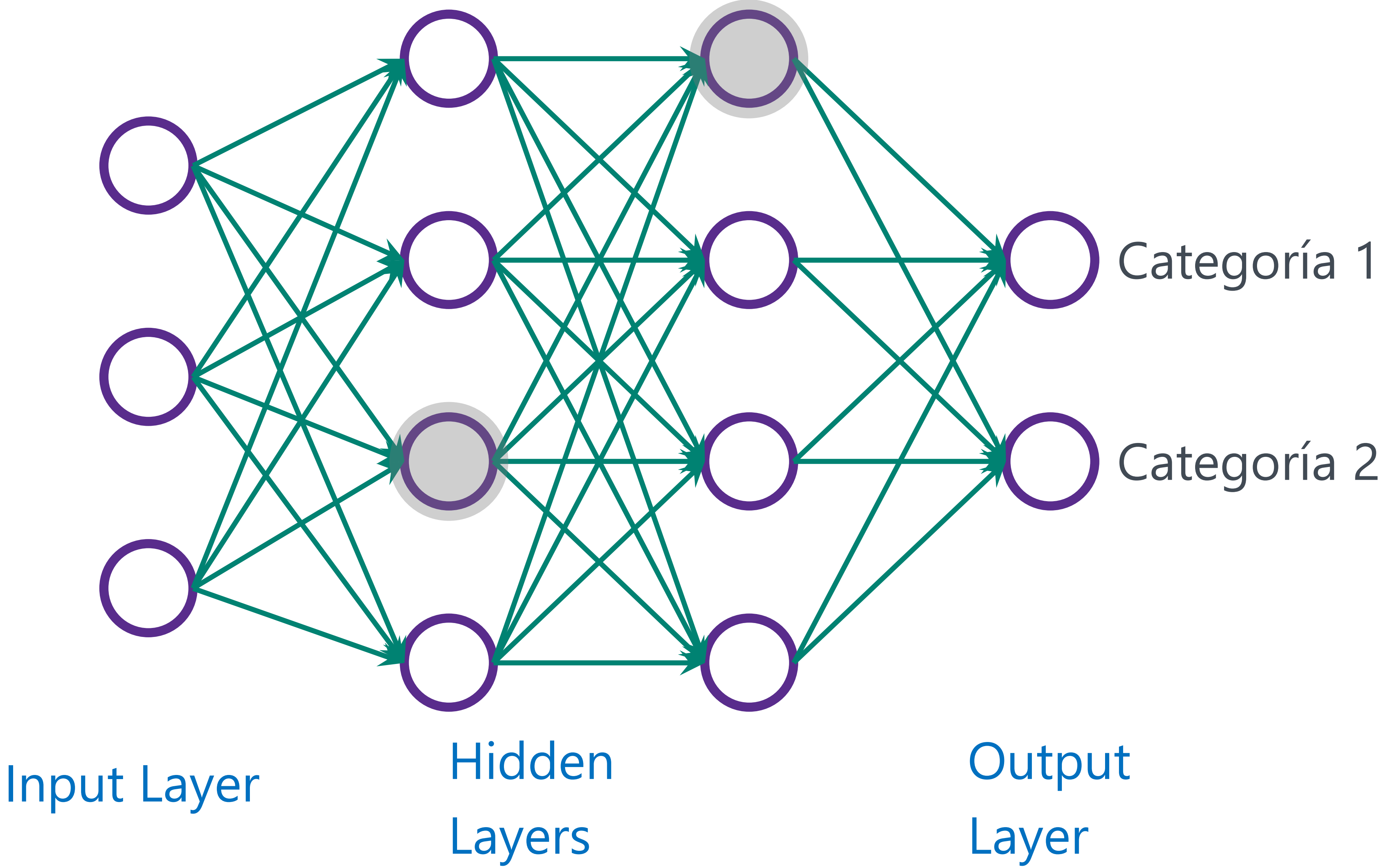
TÉCNICAS DE VISIÓN ARTIFICIAL

Perceptrón Multicapa

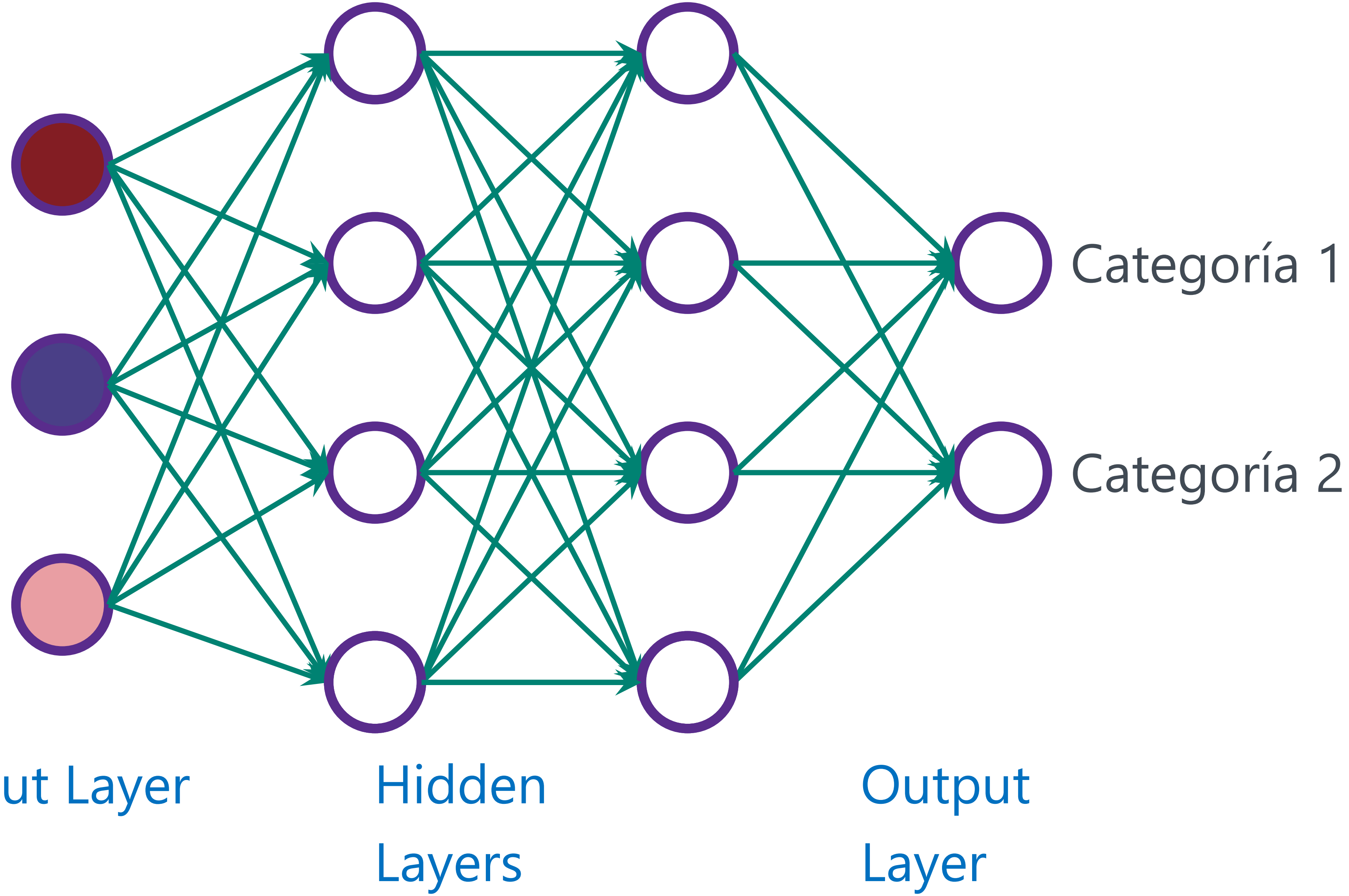
REDES NEURONALES - CLASIFICADOR



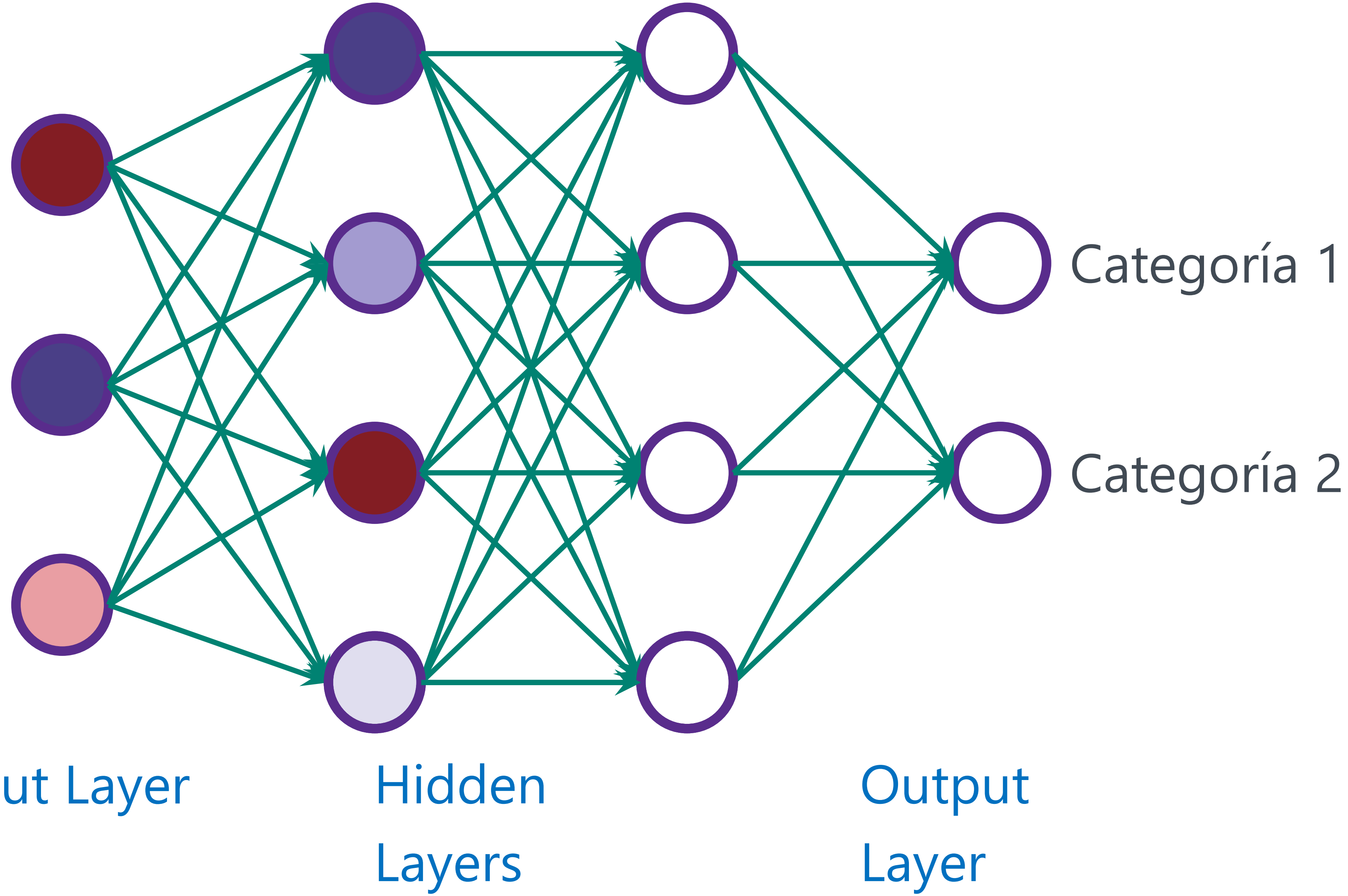
REDES NEURONALES - CLASIFICADOR



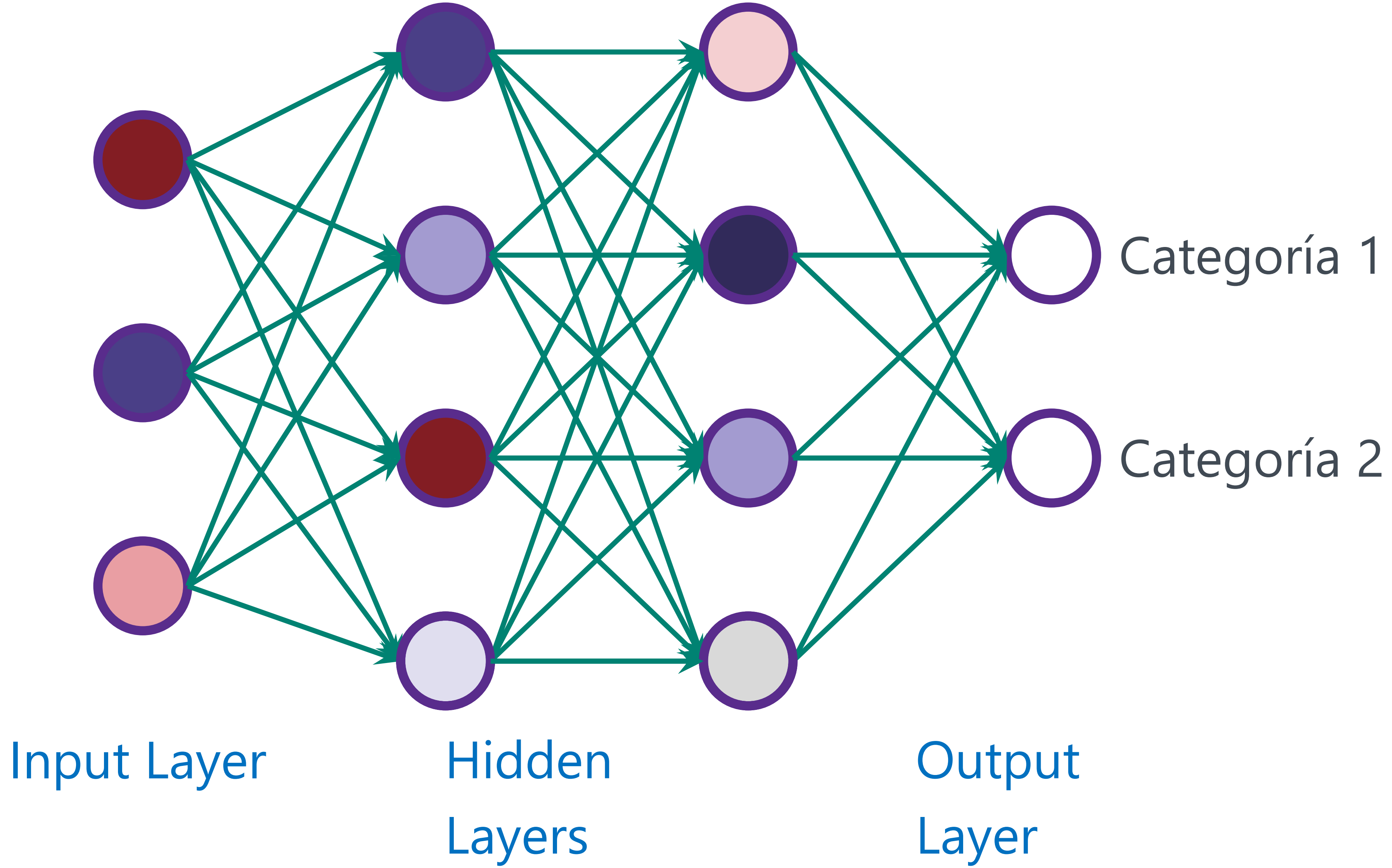
FORWARD PROPAGATION



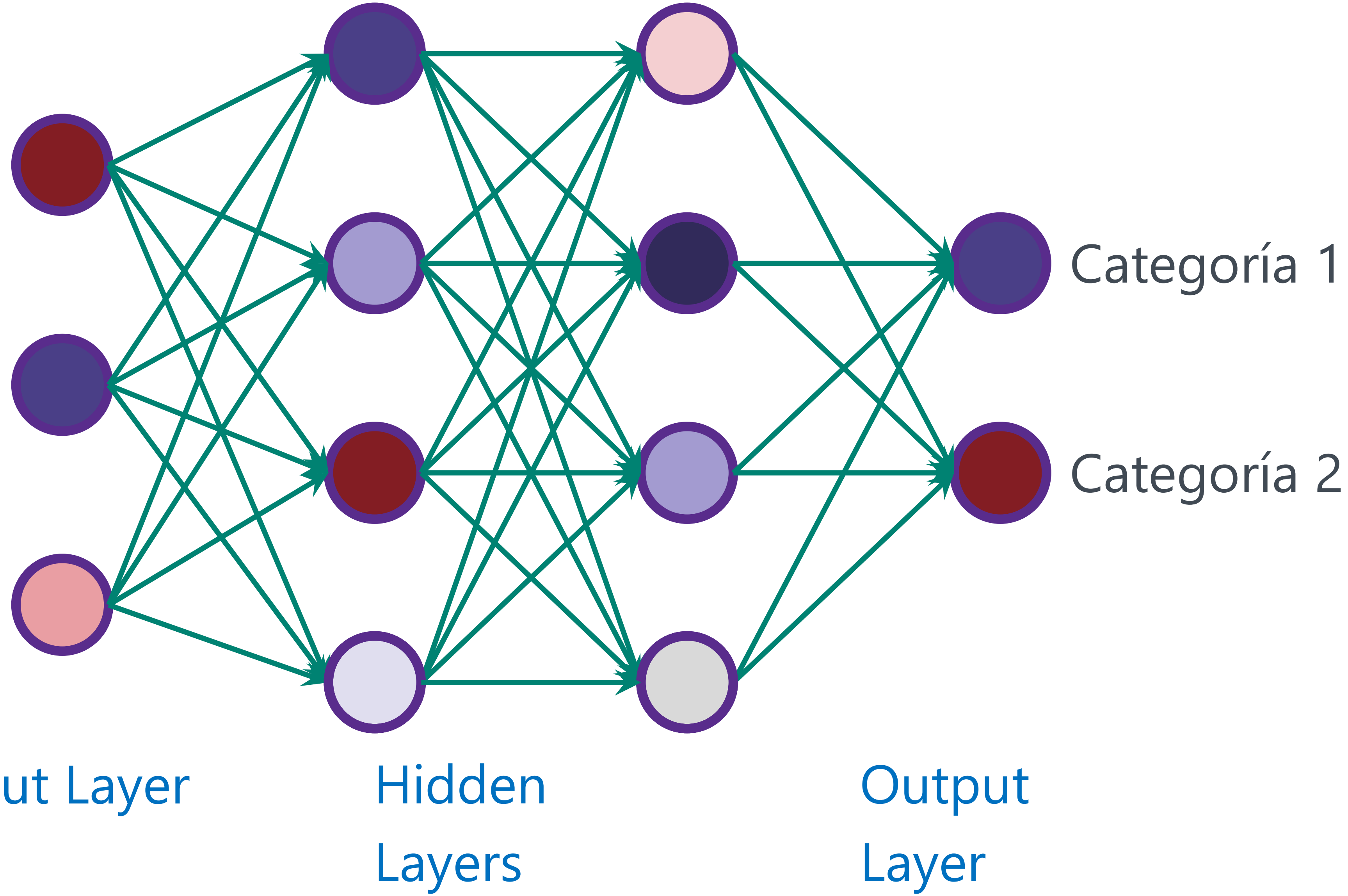
FORWARD PROPAGATION



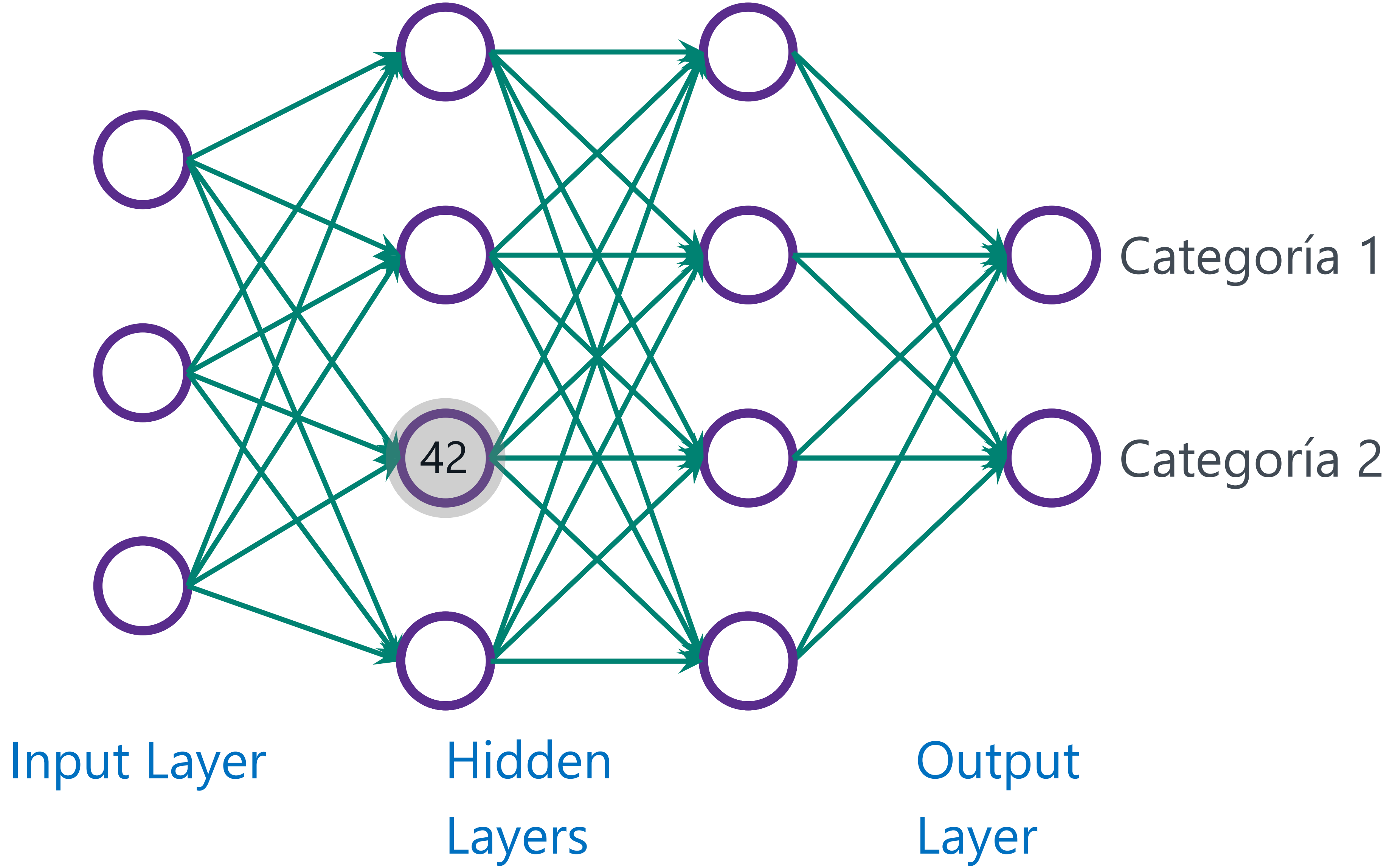
FORWARD PROPAGATION



FORWARD PROPAGATION



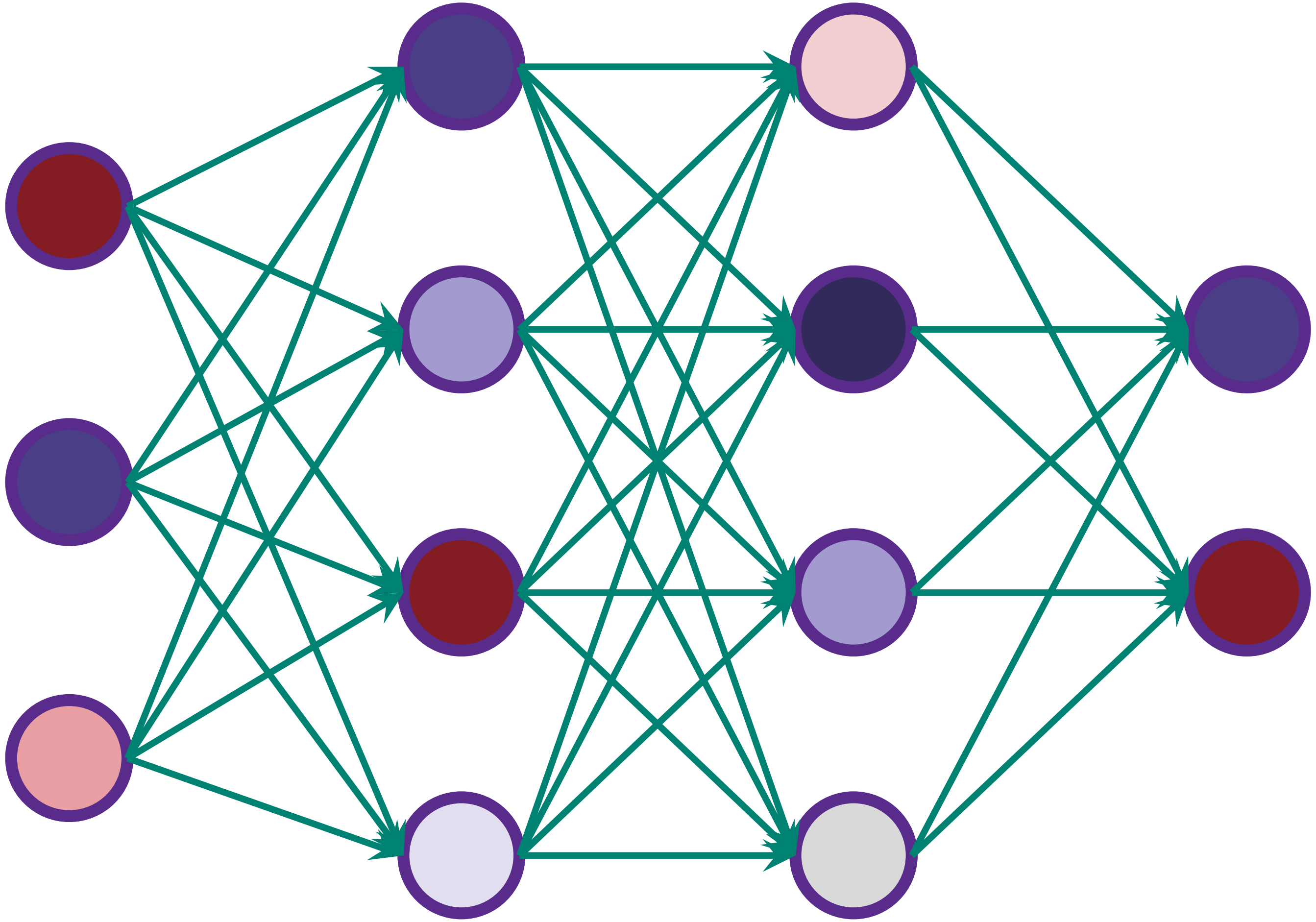
PESOS Y BIAS



42
Categoría 1
Categoría 2
Input Layer
Hidden
Layers
Output
Layer

ENTRENAMIENTO

Coste = Valor Generado – Valor Real



Categoría 1

Categoría 2

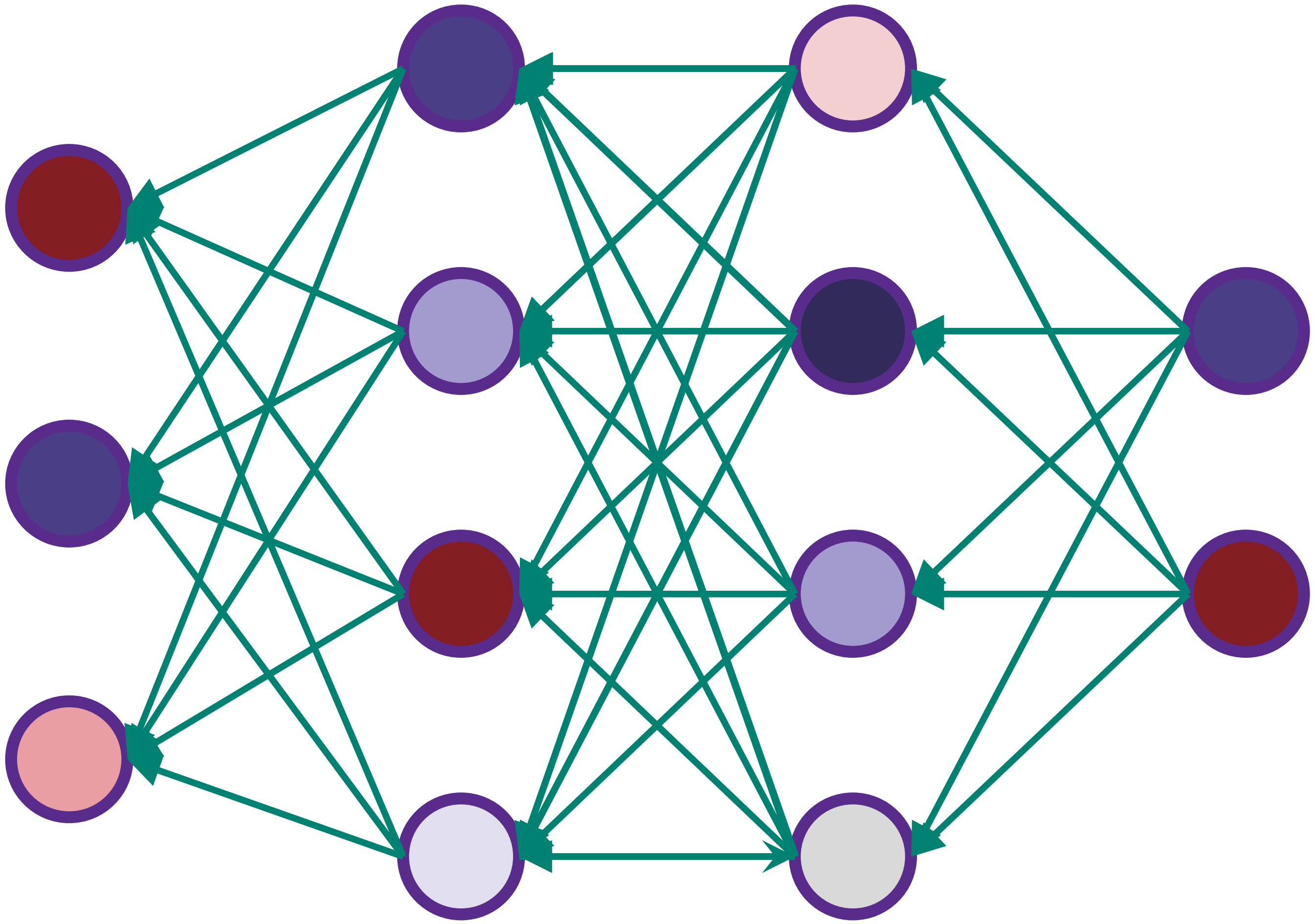
Input Layer

Hidden Layers

Output Layer

BACK PROPAGATION

Coste = Valor Generado – Valor Real



Categoría 1

Error = X

Categoría 2

Input Layer

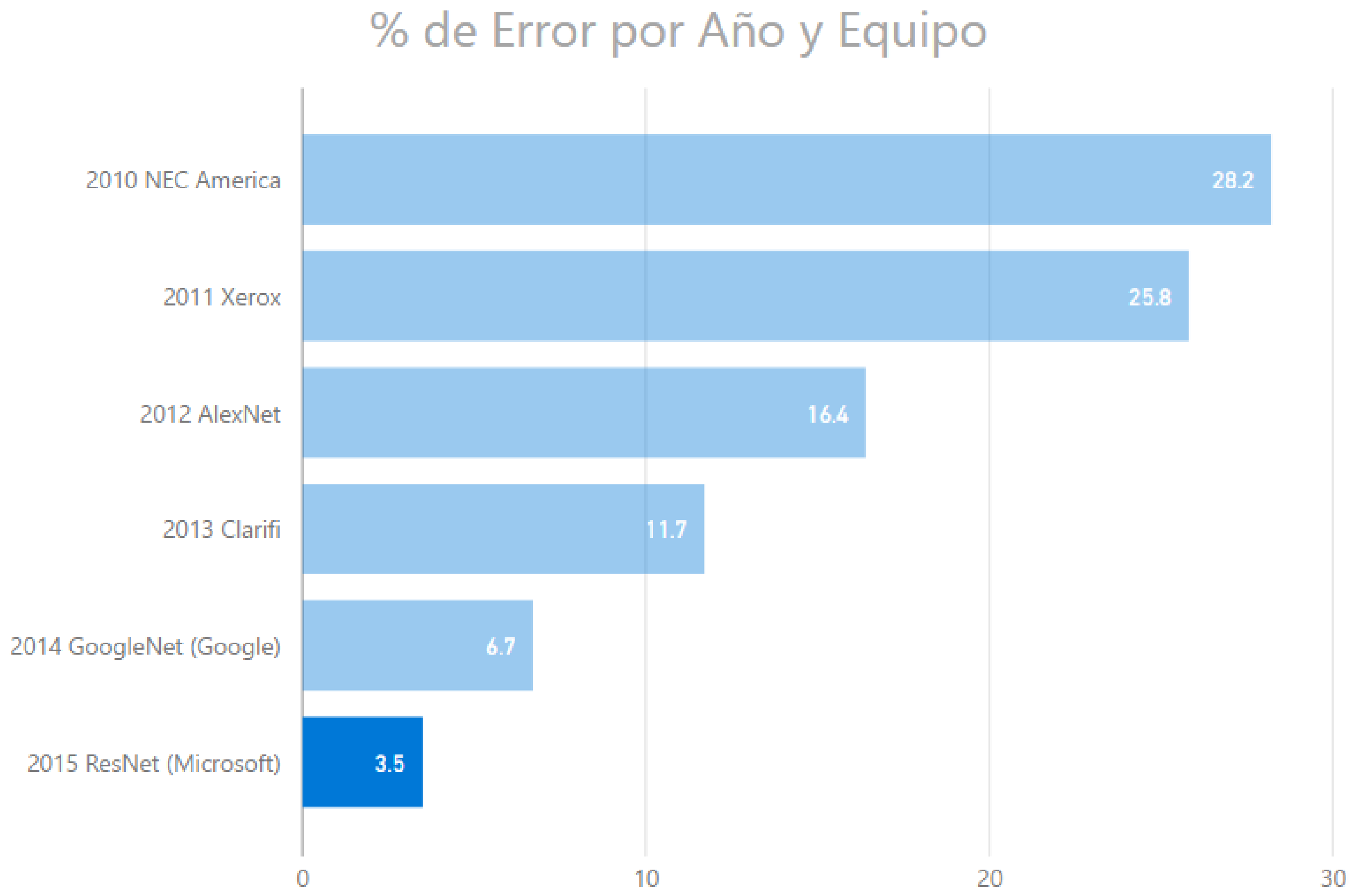
Hidden Layers

Output Layer

LAB 2: PERCEPTRÓN MULTICAPA

DEEP LEARNING

IMAGENET CHALLENGE



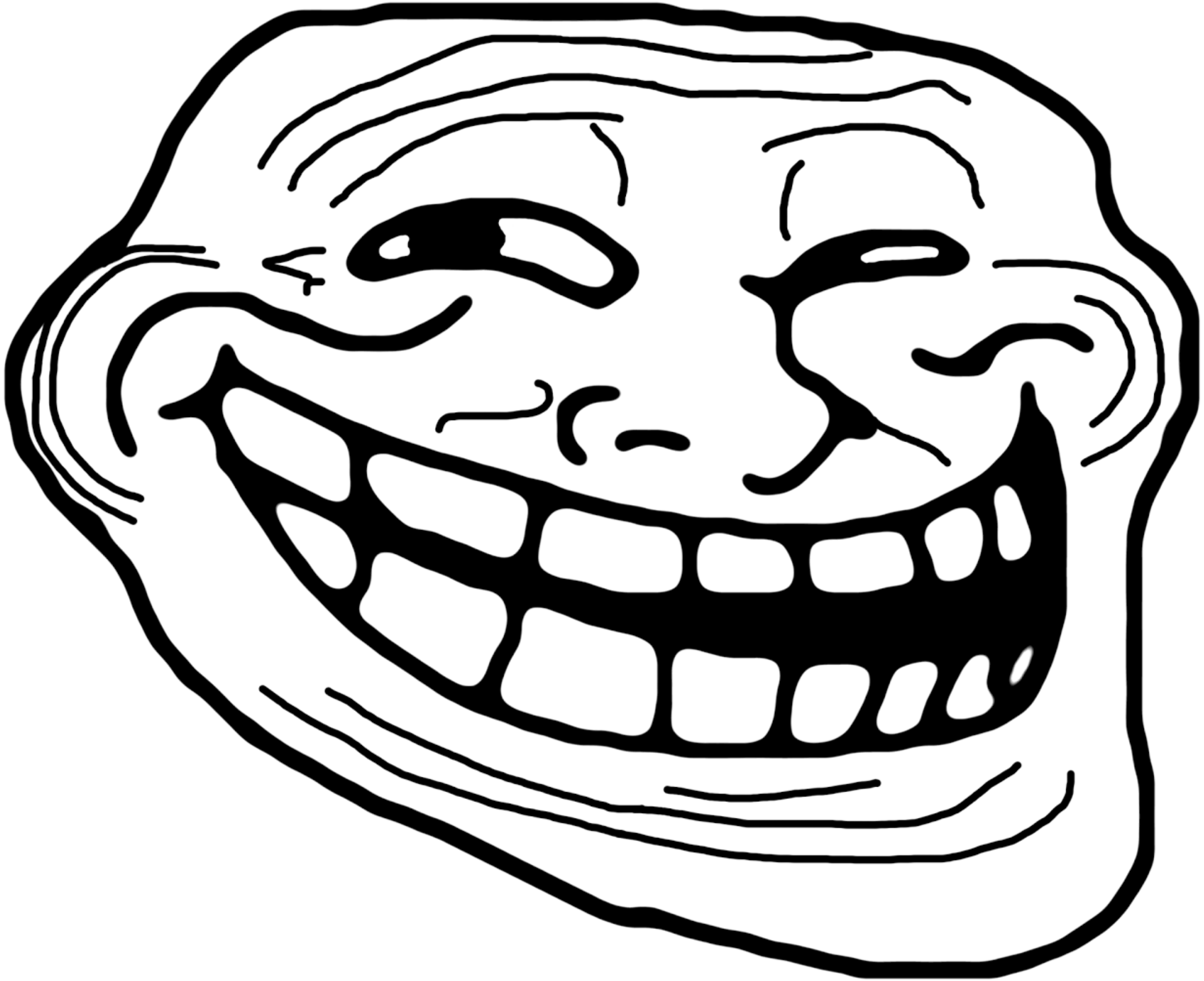
TÉCNICAS DE VISIÓN ARTIFICIAL

Redes Neuronales Convolucionales

TÉCNICAS DE VISIÓN ARTIFICIAL

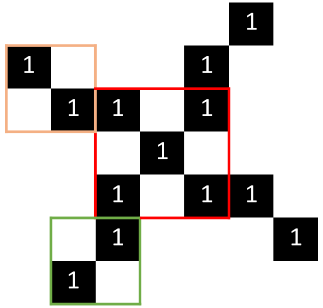
Redes Neuronales Convolucionales

Con Excel



FILTRADO

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



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

FILTRADO

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

Filtro 1

1	-1	-1
-1	1	-1
-1	-1	1

Filtro 2

-1	-1	1
-1	1	-1
1	-1	-1

Filtro 3

1	-1	1
-1	1	-1
1	-1	1

1	1	1
1	1	1
1	1	1

1.0

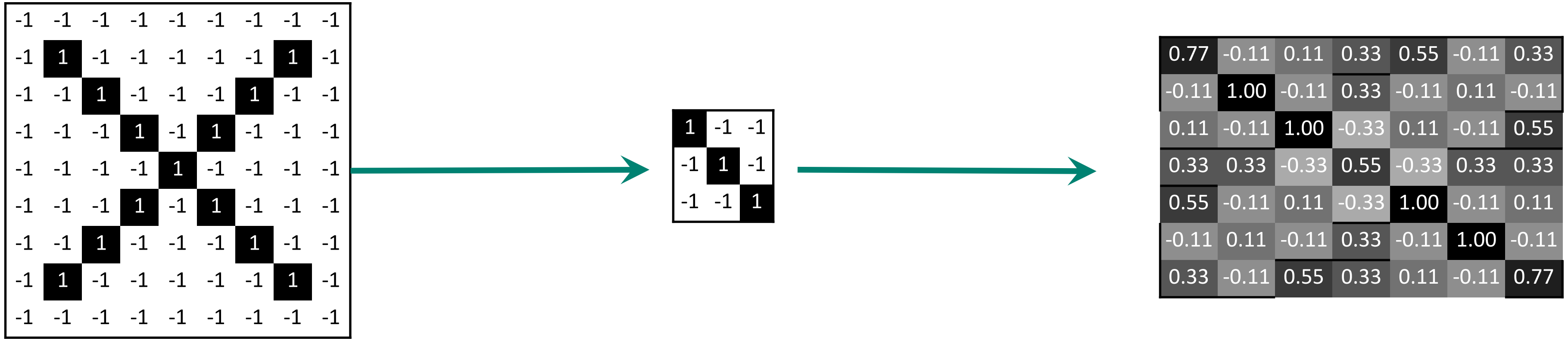
-1	1	-1
1	1	1
-1	1	-1

0.11

1	1	-1
1	1	1
-1	1	1

0.55

FILTRADO

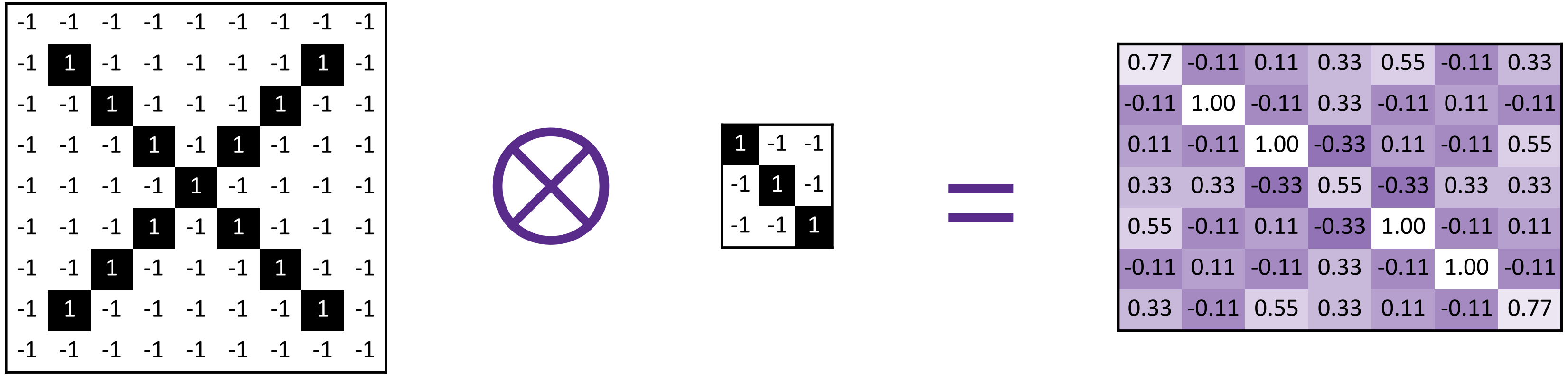


-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

1	-1	-1
-1	1	-1
-1	-1	1

0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

CONVOLUCIÓN



-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

⊗

1	-1	-1
-1	1	-1
-1	-1	1

=

0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

CONVOLUCIÓN

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

⊗

1	-1	-1
-1	1	-1
-1	-1	1

=

0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

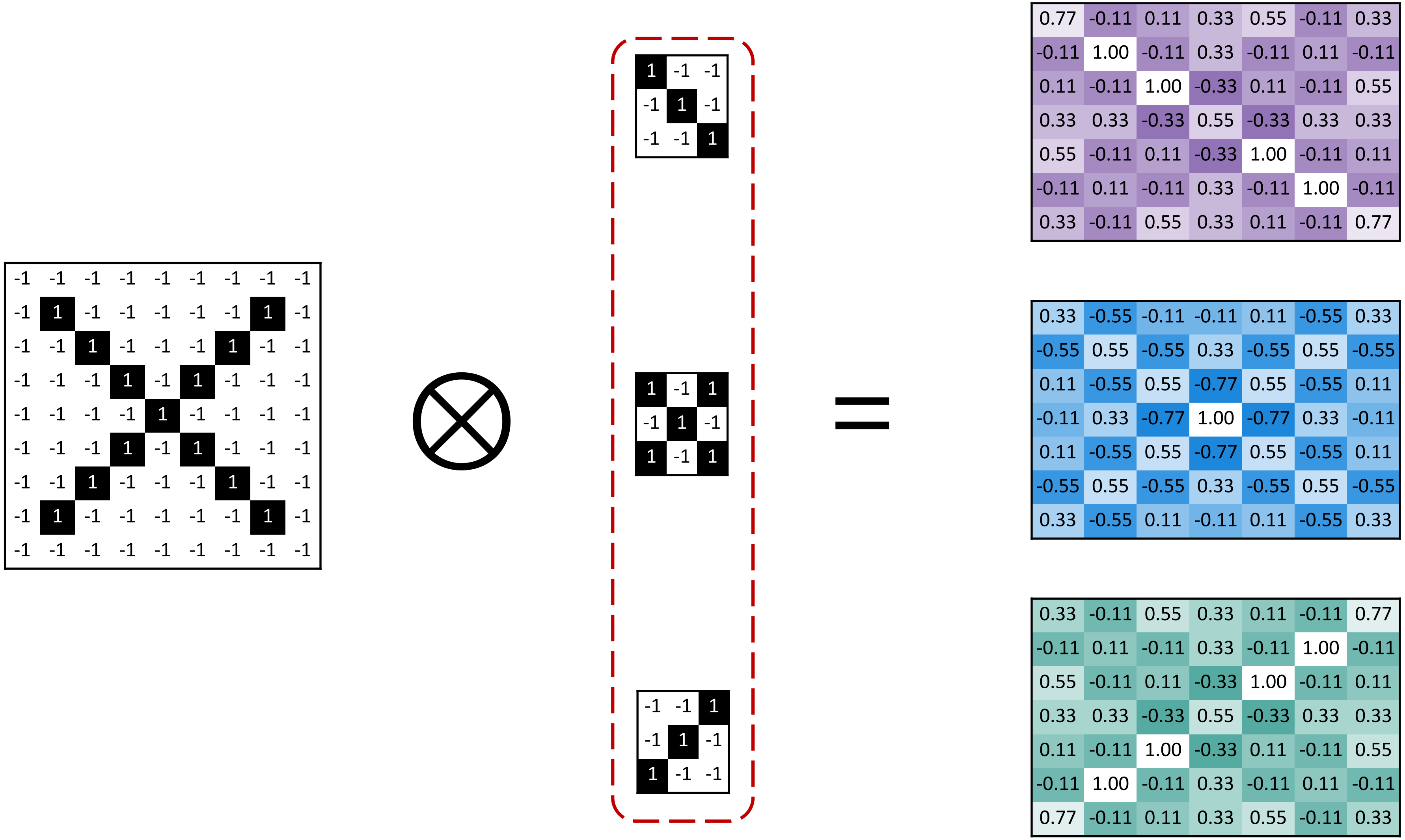
⊗

-1	-1	1
-1	1	-1
1	-1	-1

=

0.33	-0.11	0.55	0.33	0.11	-0.11	0.77
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.77	-0.11	0.11	0.33	0.55	-0.11	0.33

CONVOLUCIÓN



-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

⊗

1	-1	-1
-1	1	-1
-1	-1	1

1	-1	1
-1	1	-1
1	-1	1

-1	-1	1
-1	1	-1
1	-1	-1

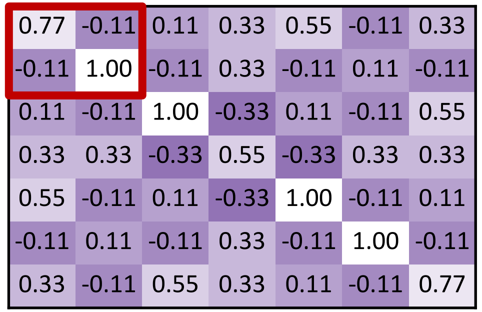
=

0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

0.33	-0.55	-0.11	-0.11	0.11	-0.55	0.33
-0.55	0.55	-0.55	0.33	-0.55	0.55	-0.55
0.11	-0.55	0.55	-0.77	0.55	-0.55	0.11
-0.11	0.33	-0.77	1.00	-0.77	0.33	-0.11
0.11	-0.55	0.55	-0.77	0.55	-0.55	0.11
-0.55	0.55	-0.55	0.33	-0.55	0.55	-0.55
0.33	-0.55	0.11	-0.11	0.11	-0.55	0.33

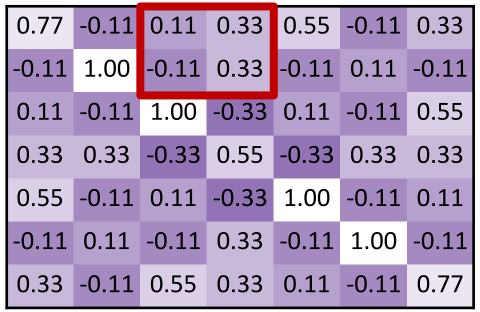
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.77	-0.11	0.11	0.33	0.55	-0.11	0.33

POOLING



0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

POOLING



0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

AVERAGE POOLING – STRIDE 1

0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

0.39	0.22	0.17	0.28	0.11	0.06
0.22	0.45	0.22	0.00	0.00	0.11
0.17	0.22	0.22	0.00	0.00	0.28
0.28	0.00	0.00	0.22	0.22	0.17
0.11	0.00	0.00	0.22	0.45	0.22
0.06	0.11	0.28	0.17	0.22	0.39

MAX POOLING – STRIDE 1

0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

1.00	1.00	0.33	0.55	0.55	0.33
1.00	1.00	1.00	0.33	0.11	0.55
0.33	1.00	1.00	0.55	0.33	0.55
0.55	0.33	0.55	1.00	1.00	0.33
0.55	0.11	0.33	1.00	1.00	1.00
0.33	0.55	0.55	0.33	1.00	1.00

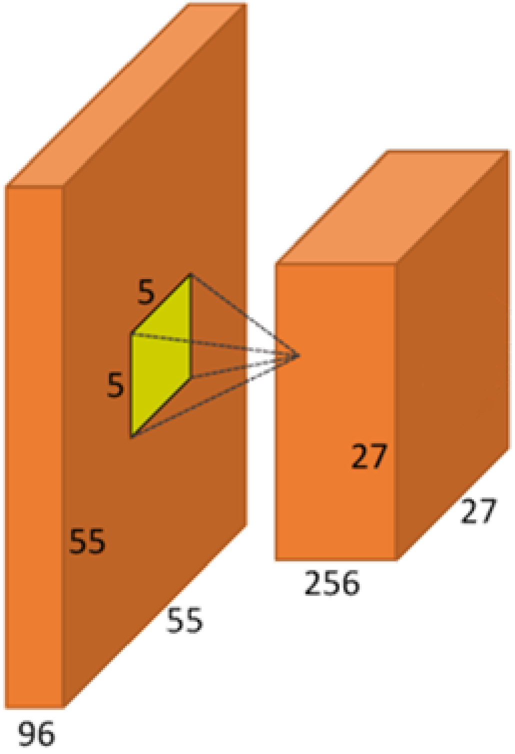
MAX POOLING – STRIDE 2

0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

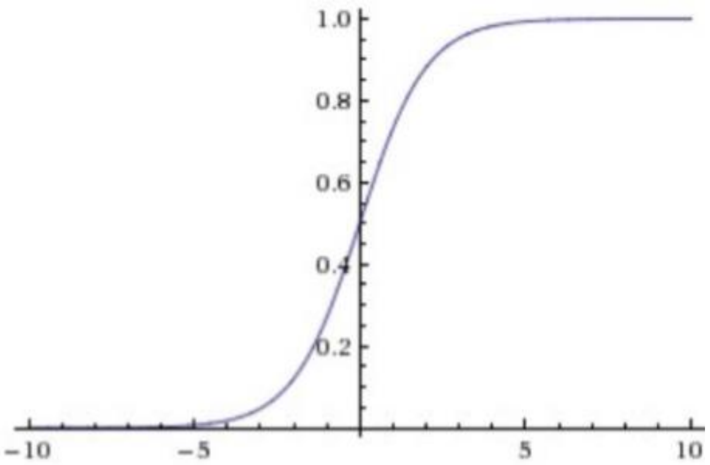
1.00	0.33	0.55	0.33
0.33	1.00	0.33	0.55
0.55	0.33	1.00	0.11
0.33	0.55	0.11	0.77

POOLING - ¿PARA QUE?

FUNCIONES DE ACTIVACIÓN



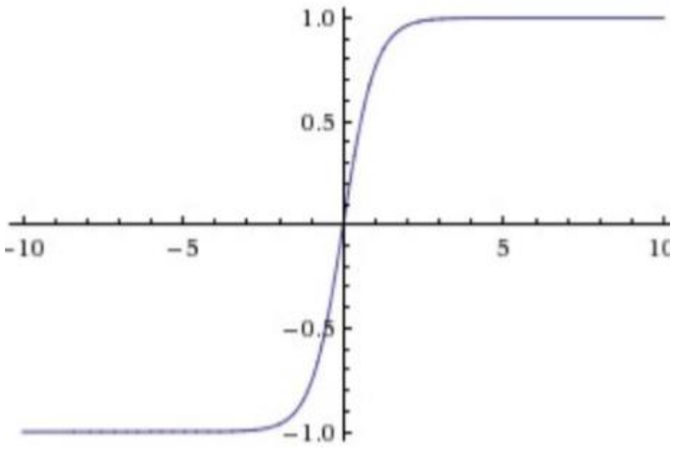
FUNCIONES DE ACTIVACIÓN



FUNCIÓN LOGÍSTICA (“*sigmoide*”):

- Simple.
- Pegas:
  - Saturación
  - No centrada en cero
- No la usamos, pero le agradecemos los servicios prestados :)

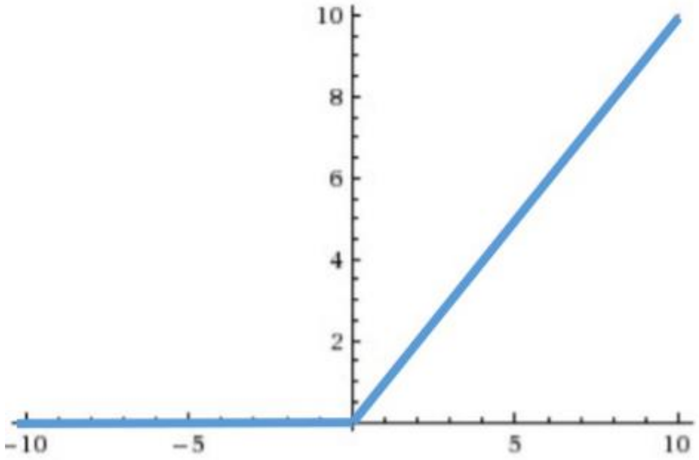
FUNCIONES DE ACTIVACIÓN



TANGENTE HIPERBÓLICA (*tanh*):

- Resuelve el problema del no-centrado en cero
- Sigue ‘matando gradientes’ por saturación
- Siempre es preferible a la función logística

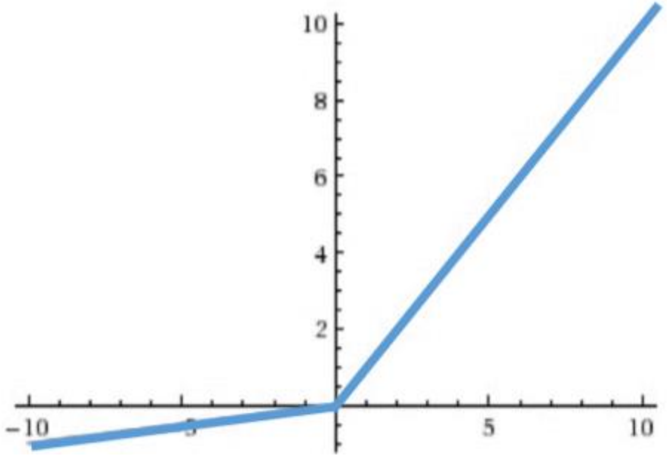
FUNCIONES DE ACTIVACIÓN



RECTIFIED LINEAR UNITS (ReLU):

- Optimización para la convergencia del SGD
- Computacionalmente más sencilla (no hay que exponenciar, etc..)
- Fragiles (pueden 'morir' dejando parte de la red inaccesible)

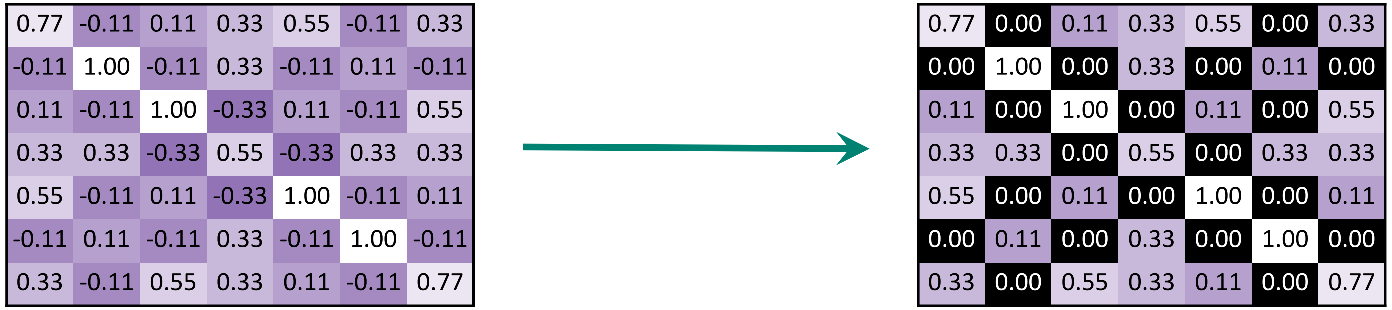
FUNCIONES DE ACTIVACIÓN



LEAKY ReLU:

- Más estables que ReLU tradicional
- La pendiente de la región negativa puede parametrizarse
- Caso particular del *MaxOut*

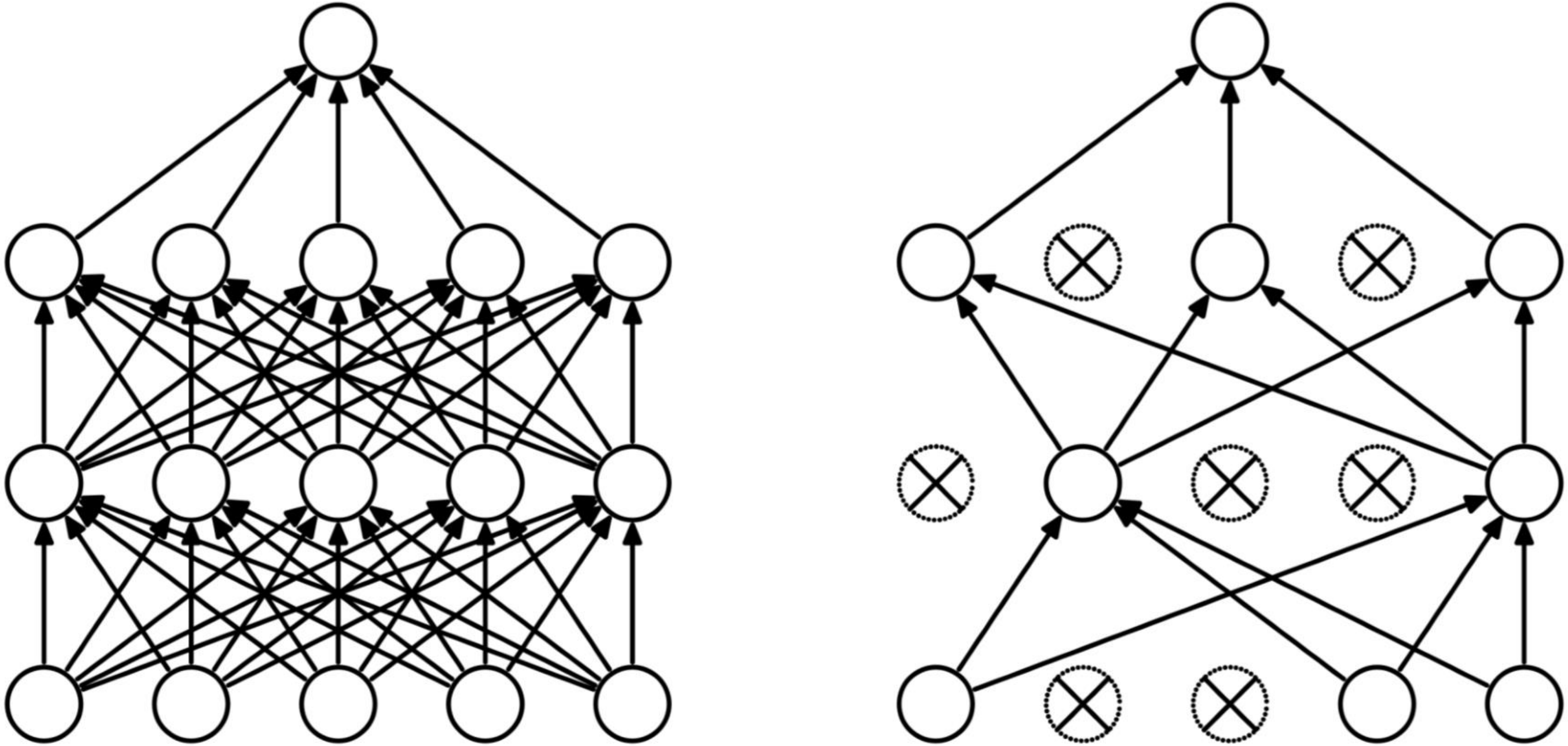
ReLU



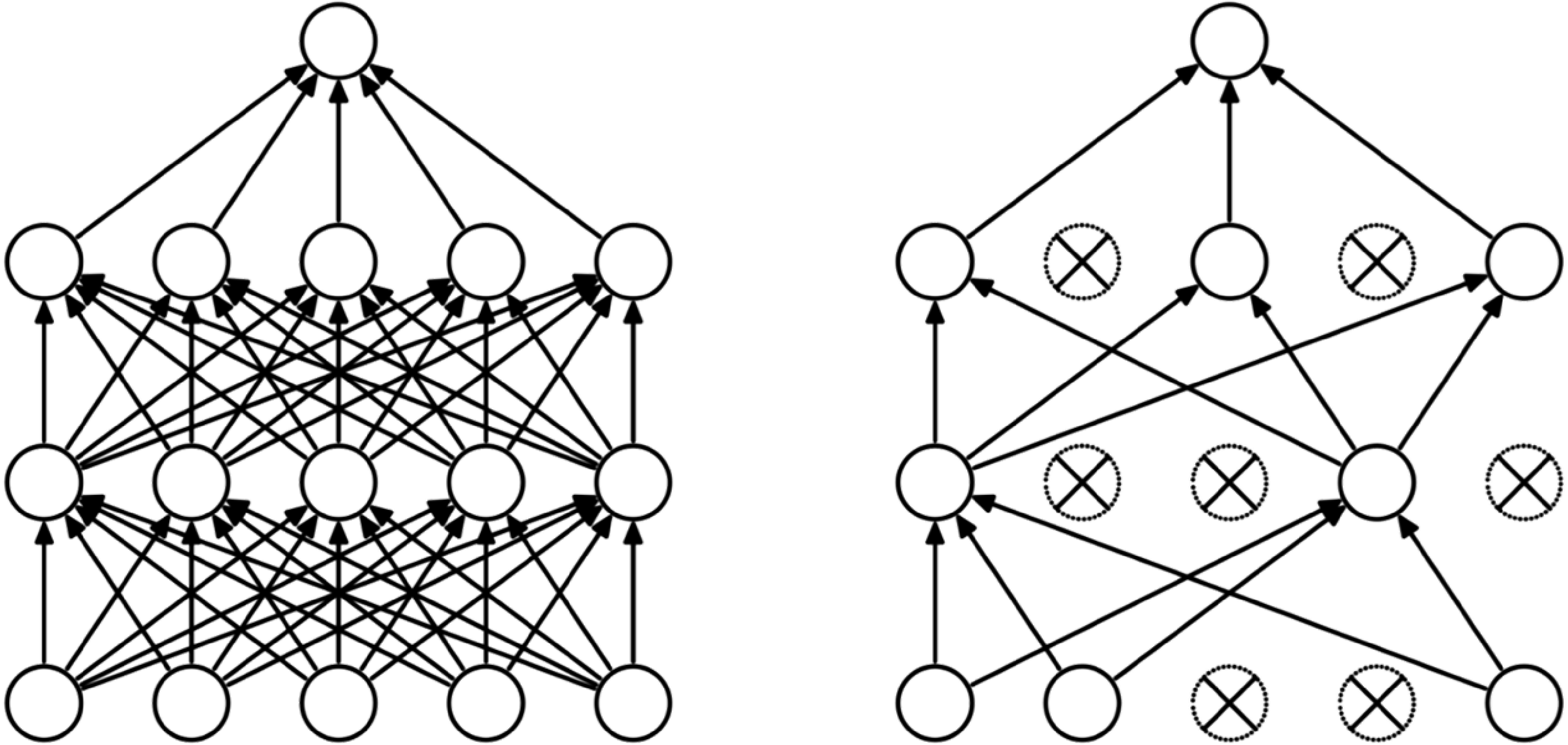
0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

0.77	0.00	0.11	0.33	0.55	0.00	0.33
0.00	1.00	0.00	0.33	0.00	0.11	0.00
0.11	0.00	1.00	0.00	0.11	0.00	0.55
0.33	0.33	0.00	0.55	0.00	0.33	0.33
0.55	0.00	0.11	0.00	1.00	0.00	0.11
0.00	0.11	0.00	0.33	0.00	1.00	0.00
0.33	0.00	0.55	0.33	0.11	0.00	0.77

DROPOUT



DROPOUT



¿QUÉ FALTA AQUÍ?

Filtro 1

1	-1	-1
-1	1	-1
-1	-1	1

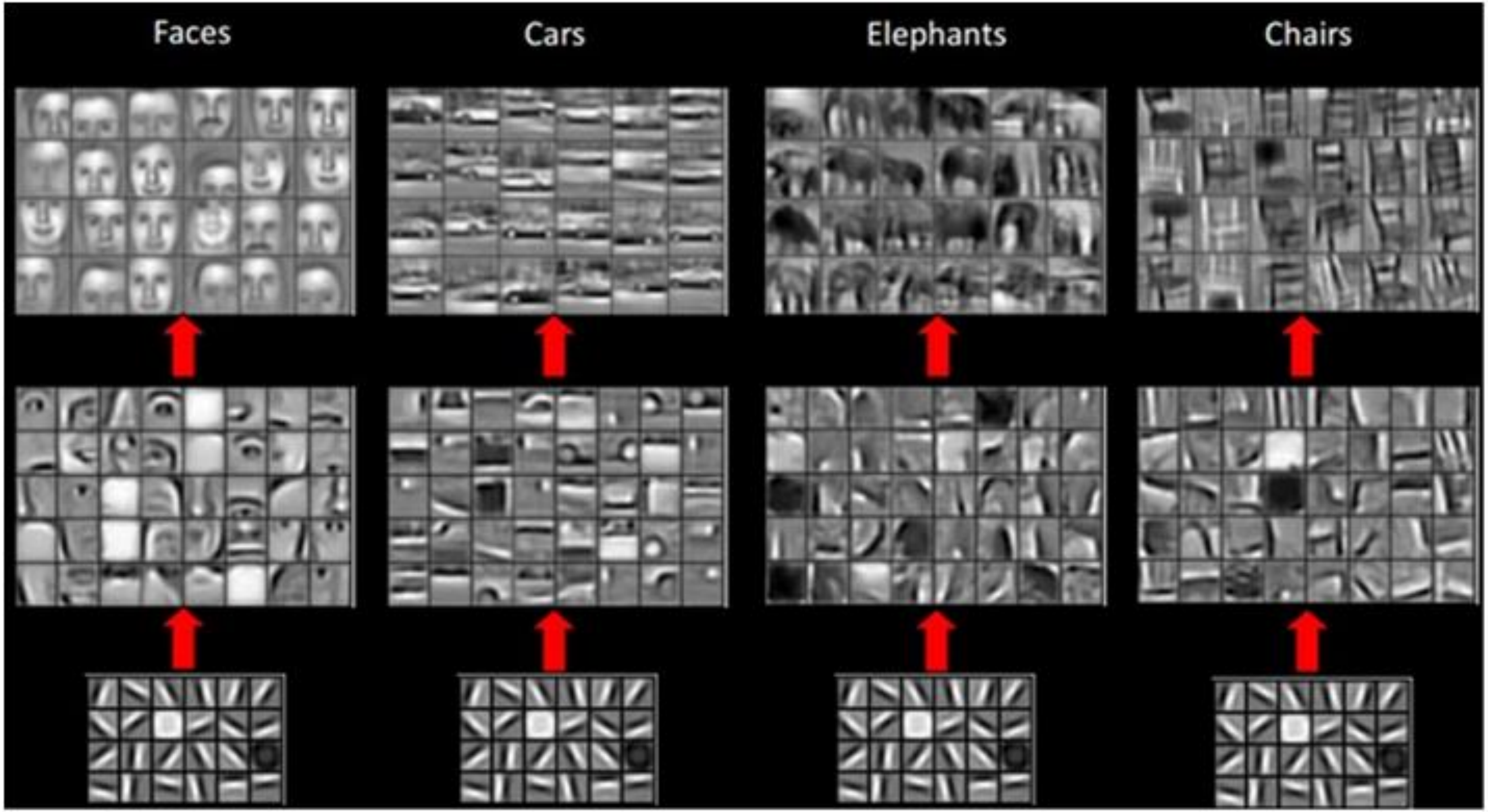
Filtro 2

-1	-1	1
-1	1	-1
1	-1	-1

Filtro 3

1	-1	1
-1	1	-1
1	-1	1

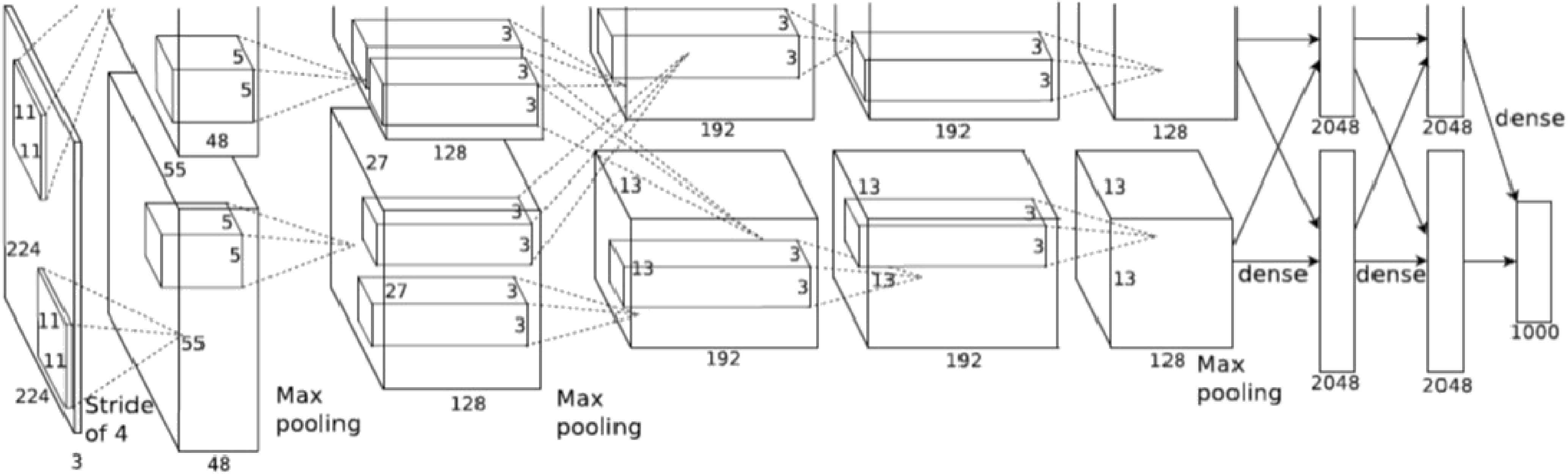
¿De dónde han salido estos?


Faces
Cars
Elephants
Chairs

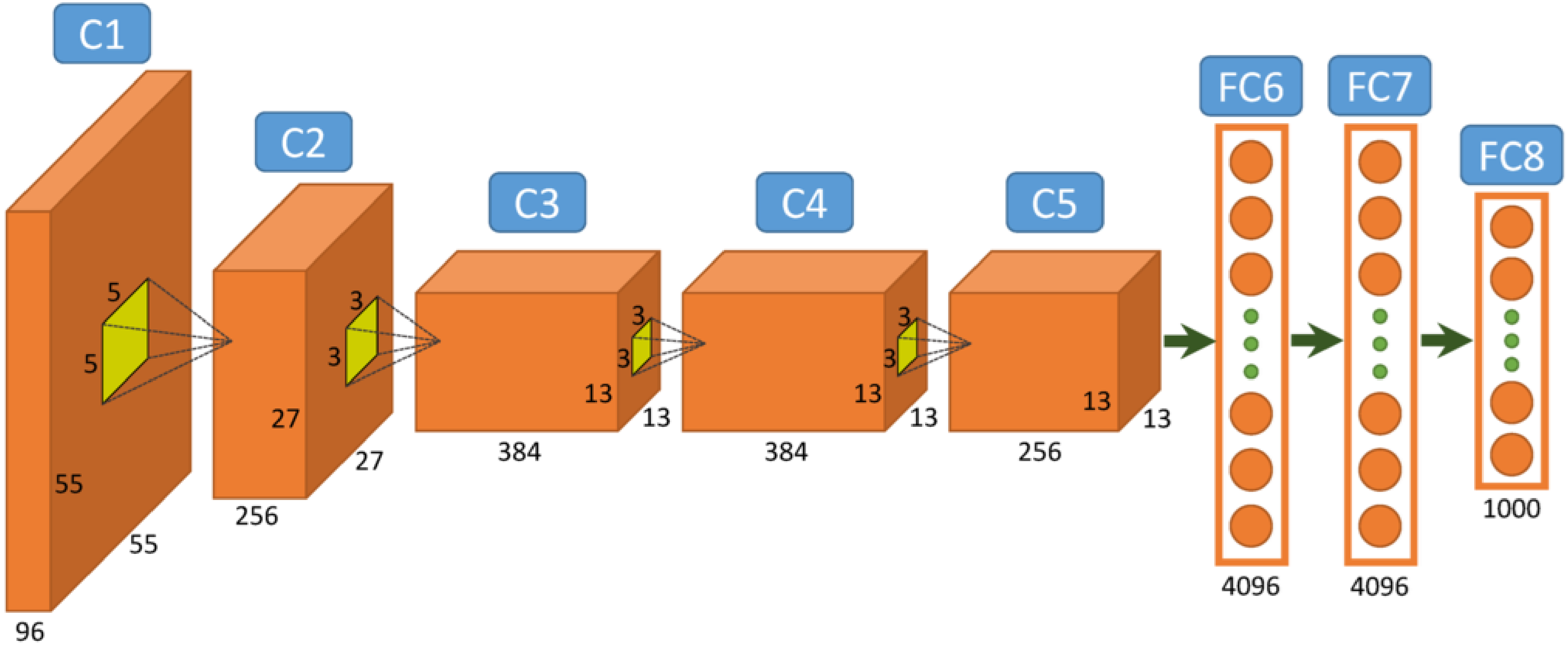
LAB 2: REDES NEURONALES CONVOLUCIONALES

ALEXNET EN CNTK (PYTHON)

EJEMPLO: AlexNet



EJEMPLO: AlexNet



PRODUCCIONALIZACIÓN

- Nuevo Servicio de AzureML
- Maquinas Virtuales de Azure de tipo NC
  - Coordinación con Azure Batch / Azure AI Training Service
- Clústers de HDInsight mediante PySpark
- Checkpoints

ALGUNAS IDEAS

- AutoEncoders
- Recurrent Neural Networks (RNN)
- Long Short Term Memory (LSTM)

ALGUNOS PAPERS INTERESANTES

“ImageNet Classification with Deep Convolutional Neural Netowrks”

[Krizhevsky, Sutskever, Hinton]

arXiv:

“Visualizing and Understanding Convolutional Networks”:

[Zeiler, Fergus]

arXiv: 1311.2901v3

“Optimization Methods for Large-Scale Machine Learning”:

[Bottou, Curtis, Nocedal]

arXiv: 1606.04838v2

CALL TO ACTION

¡Cacharread!

Cognitive toolkit @ https://cntk.ai

Tutoriales @ https://cntk.ai/pythondocs/tutorials.html

Notebooks @ https://notebooks.azure.com/cntk/libraries/tutorials

“Road to Deep Learning” en Plain<TV>

Entended el backpropagation, ¡en serio! ☺