

Lunes 6 de noviembre

Clase didáctica

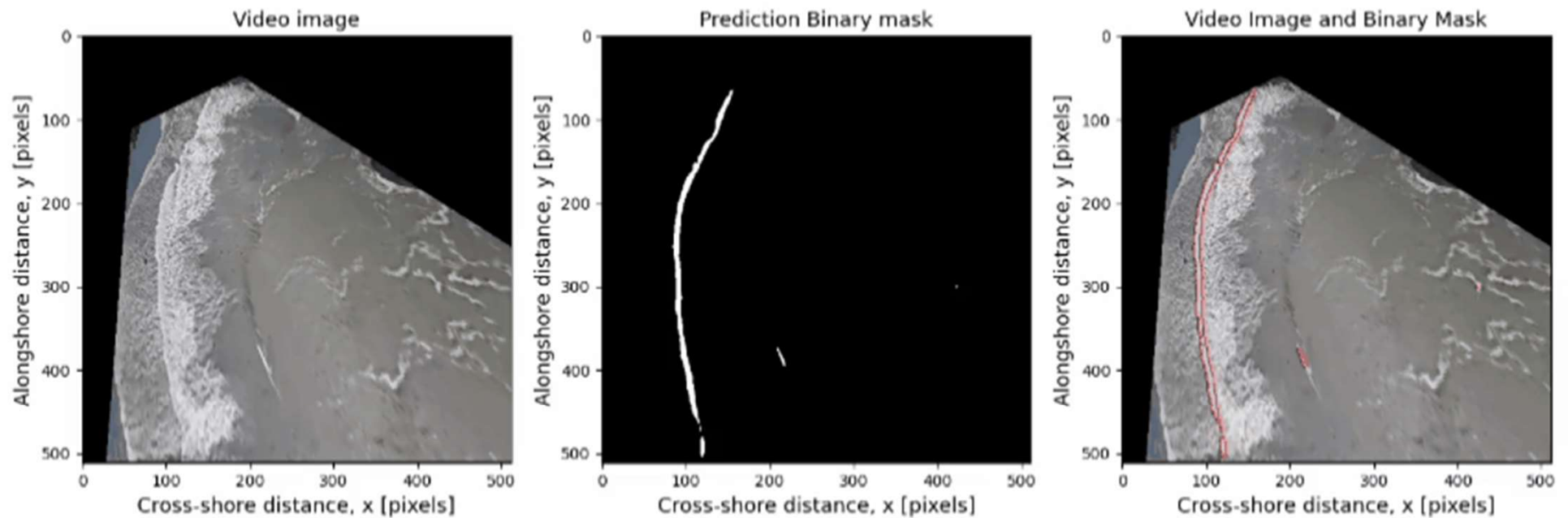
Redes Neuronales Artificiales

Profesor Carlos Valle Vidal

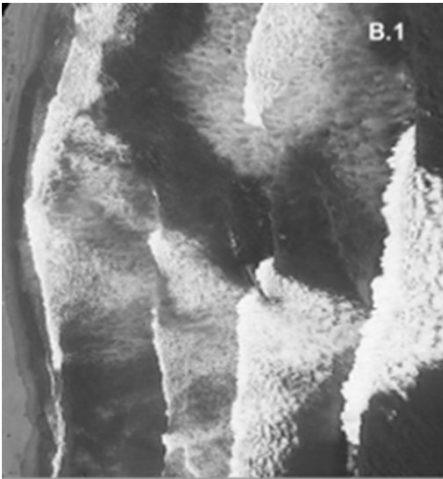
¿Por qué estudiar redes neuronales ?

Las redes neuronales nos permiten resolver muchos
problemas reales.

Ejemplo: Detección de ruptura de ola

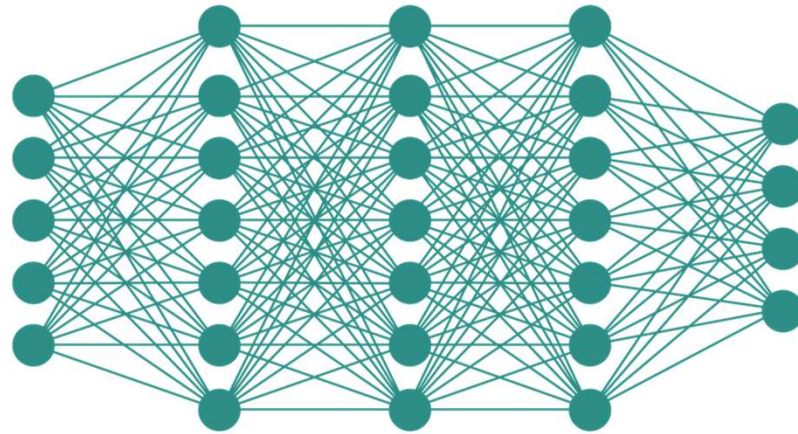


Ejemplo: Detección de ruptura de ola



1

Ejemplo: Detección de ruptura de ola



1

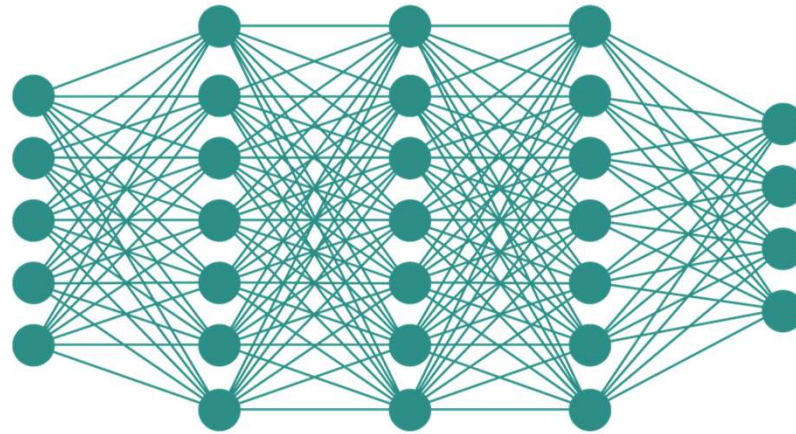
2

Red Neuronal

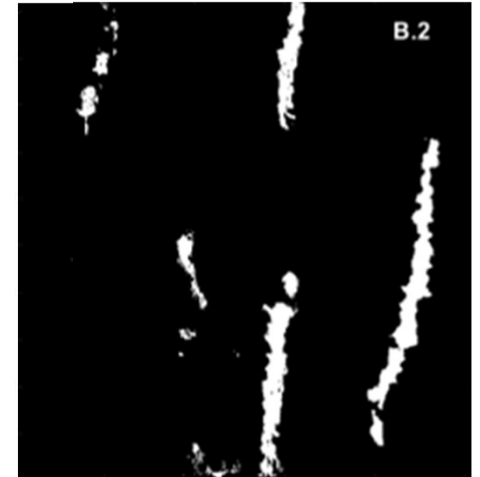
Ejemplo: Detección de ruptura de ola



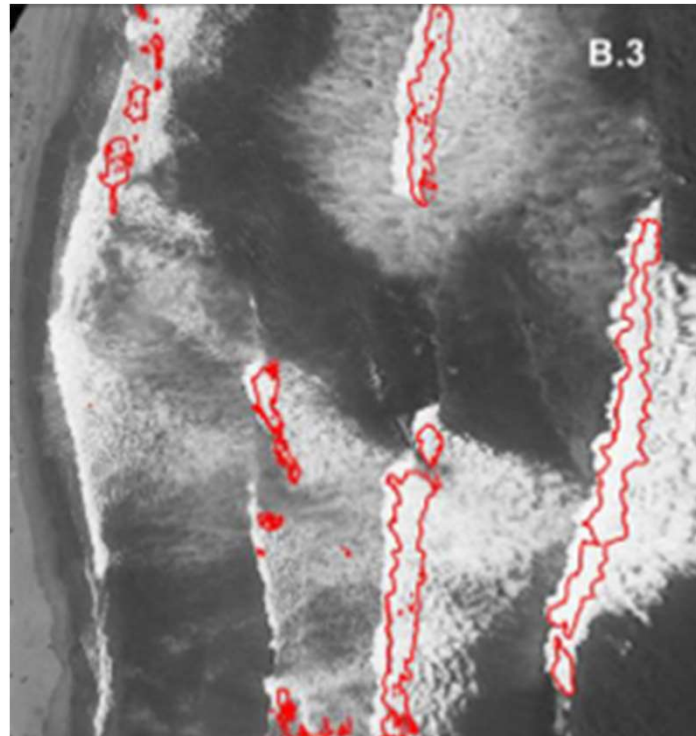
1



2



3

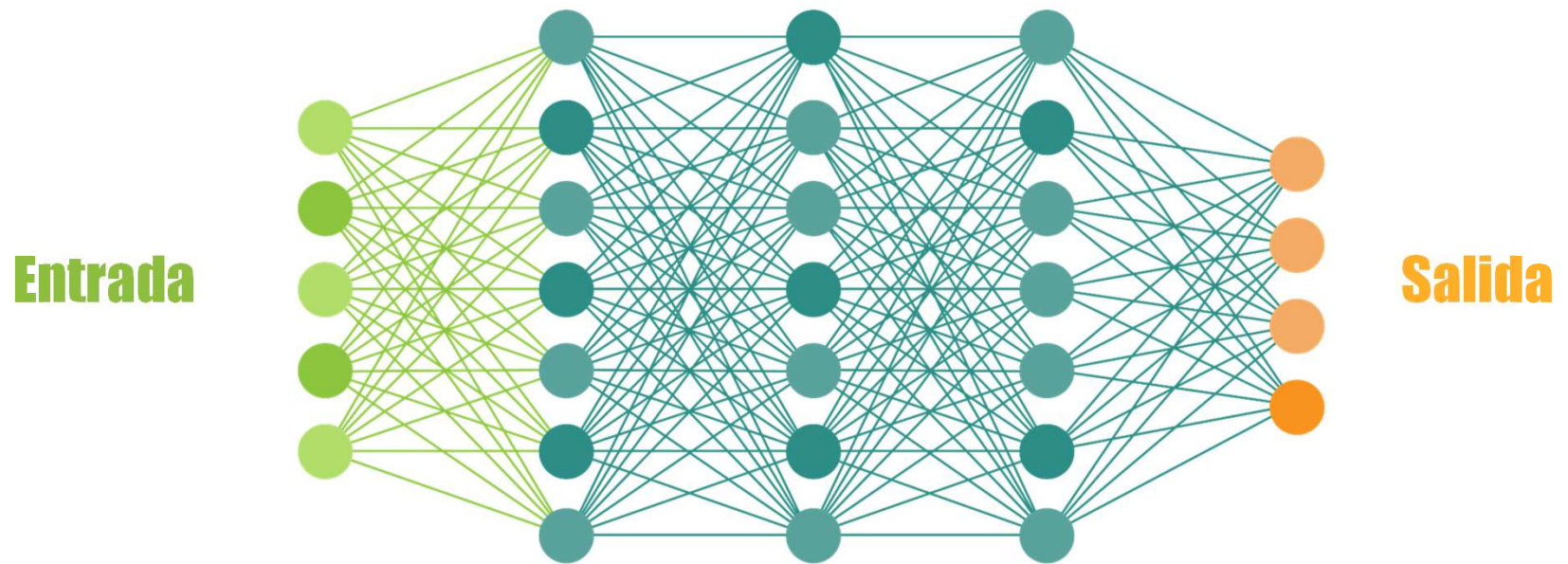


Permite la detección del punto de ruptura de las olas.

¿Qué son las redes neuronales ?

Las redes neuronales son modelos **bioinspirados**, y los entenderemos como una **red de elementos de cómputo no lineales interconectados**.

Red Neuronal





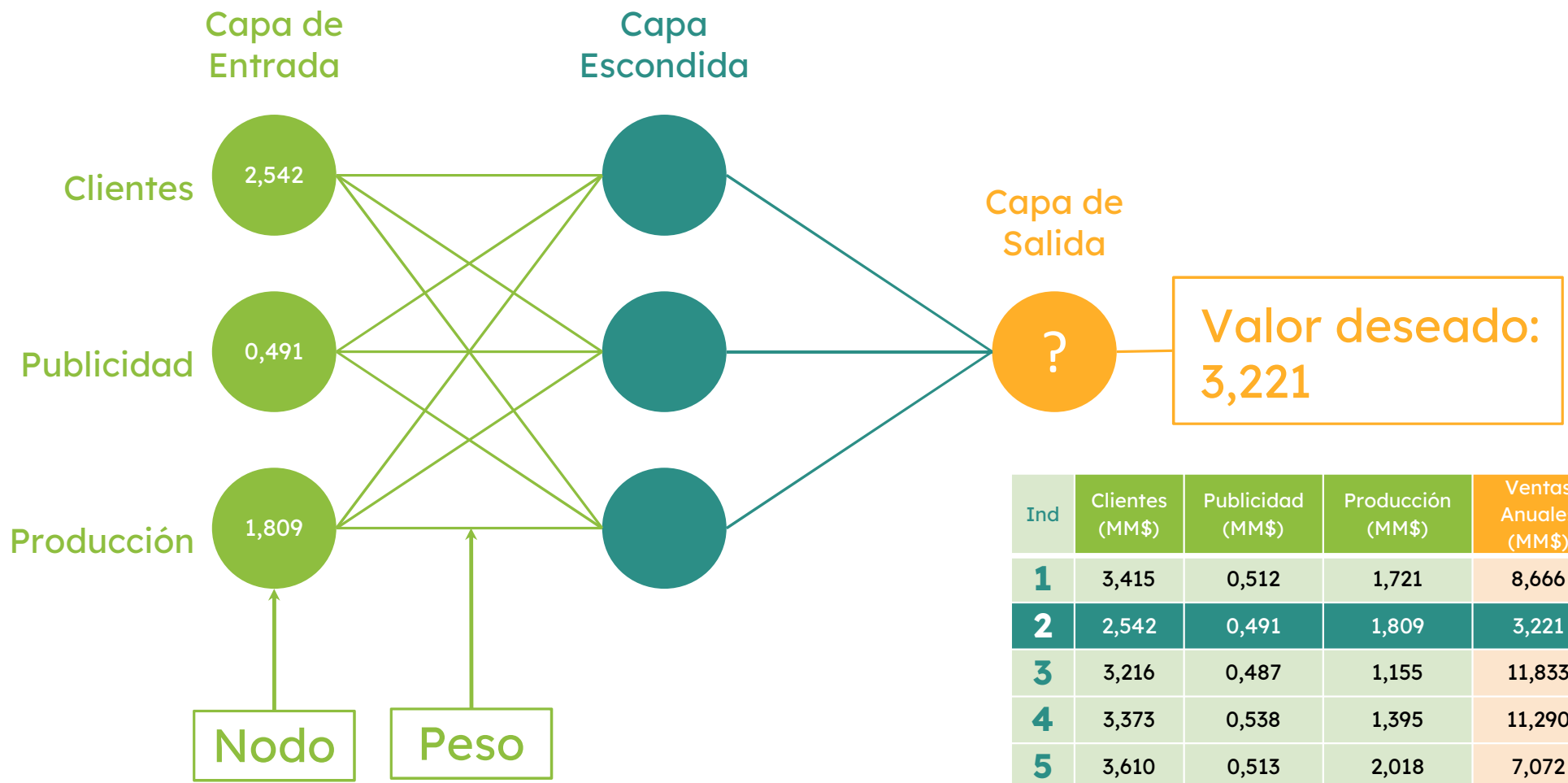
Ejemplo

Predecir las ventas anuales de una empresa (MM\$)
teniendo en cuenta:

- **Ingreso total de posibles clientes (MM\$)**
- **Gastos de publicidad de la industria (MM\$)**
- **Costos de producción (MM\$)**

Conjunto de Entrenamiento

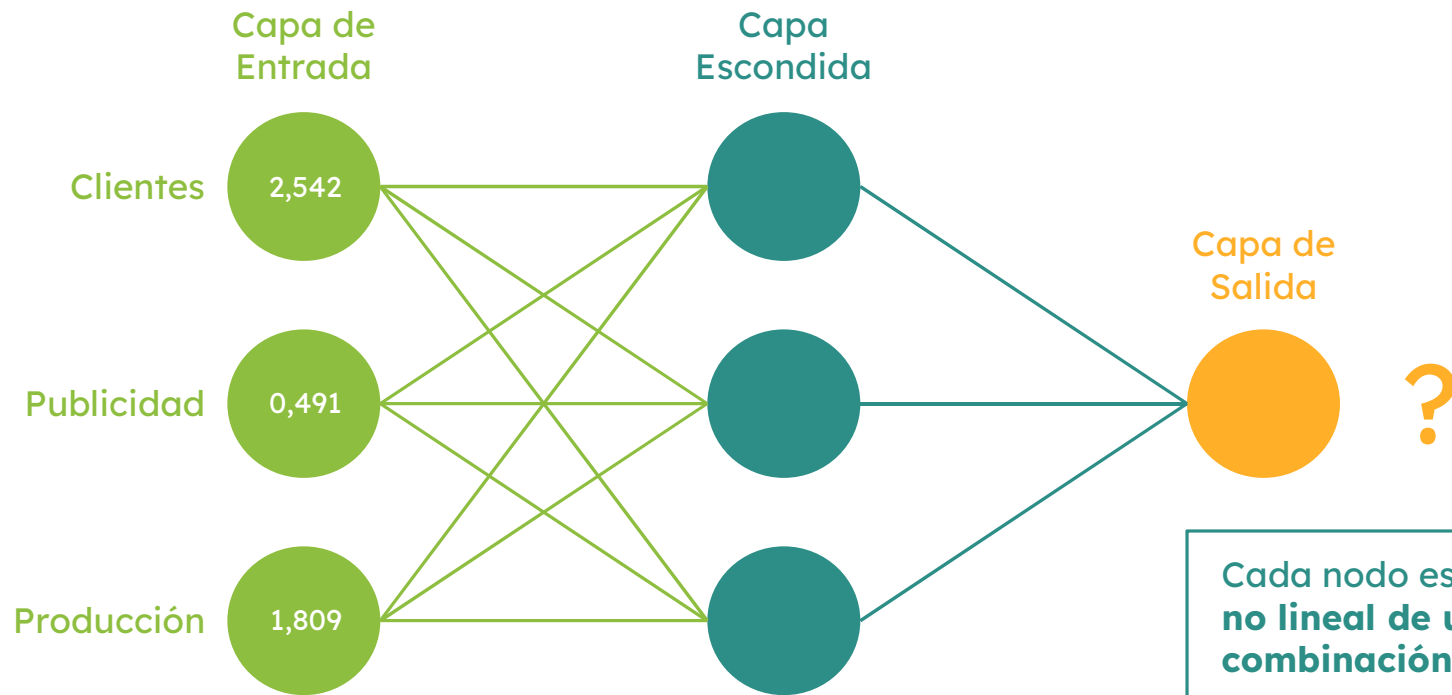
Ind	Clientes (MM\$)	Publicidad (MM\$)	Producción (MM\$)	Ventas Anuales (MM\$)
1	3,415	0,512	1,721	8,666
2	2,542	0,491	1,809	3,221
3	3,216	0,487	1,155	11,833
4	3,373	0,538	1,395	11,290
5	3,610	0,513	2,018	7,072





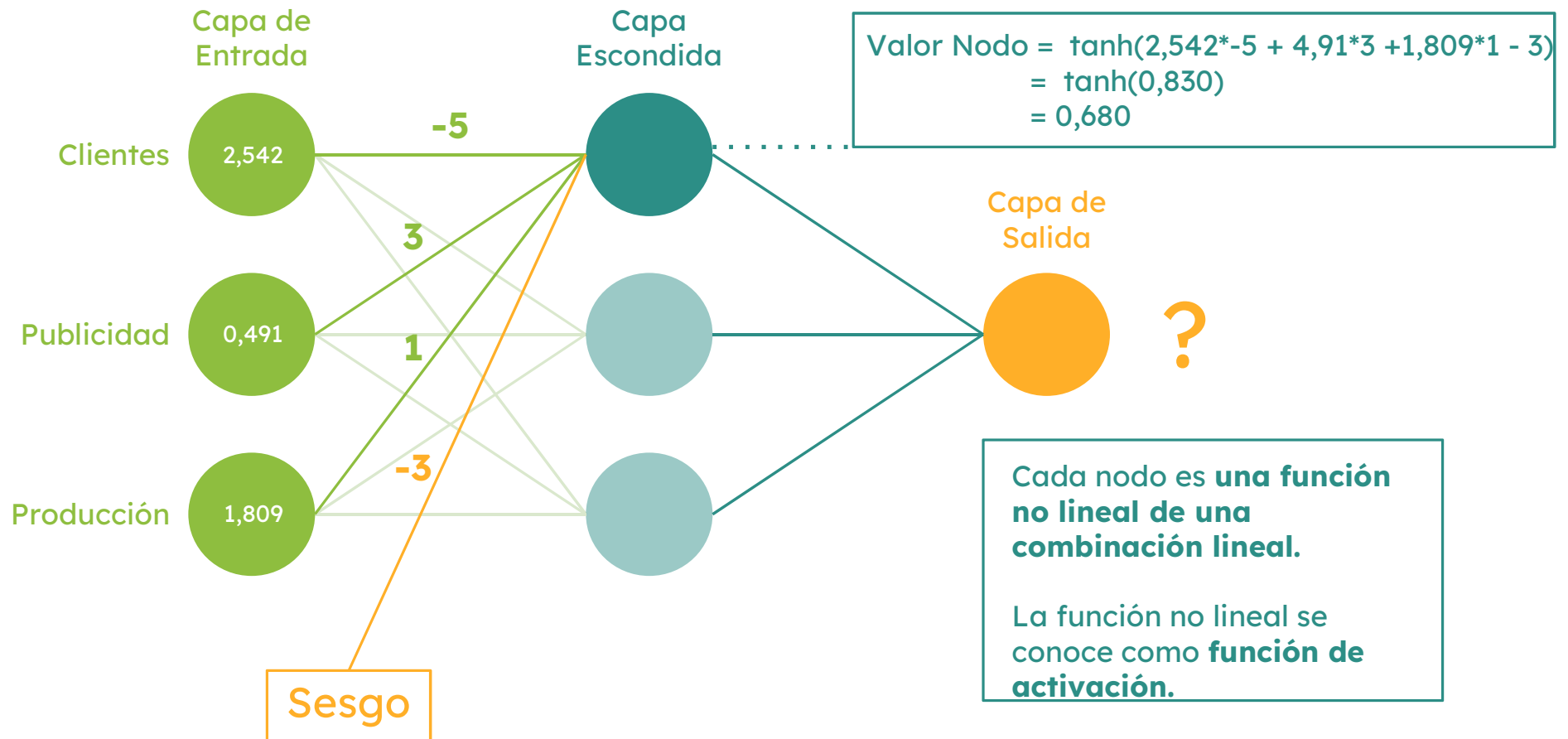
Redes Neuronales Feedforward

¿Cómo se computan los nodos de la capa escondida?

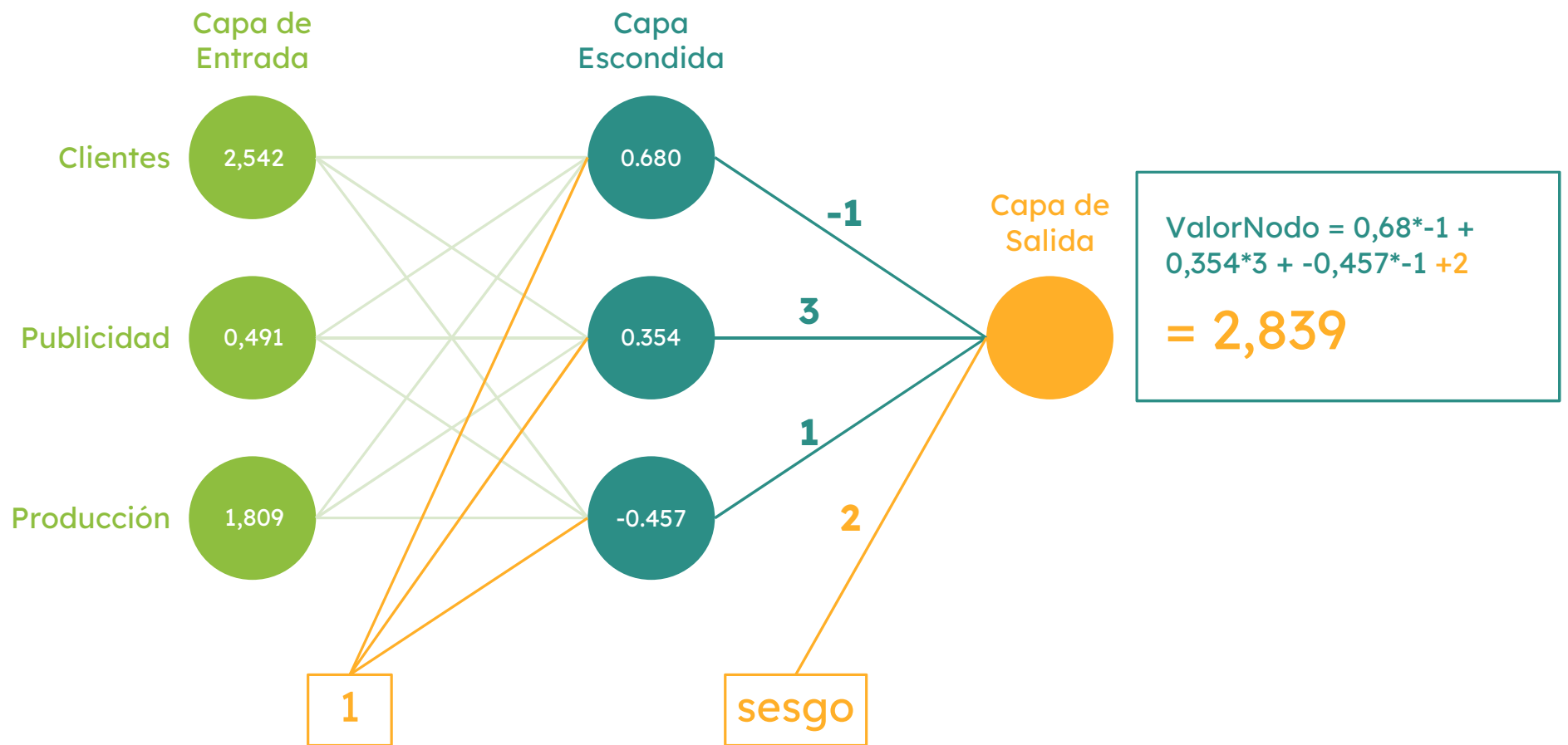


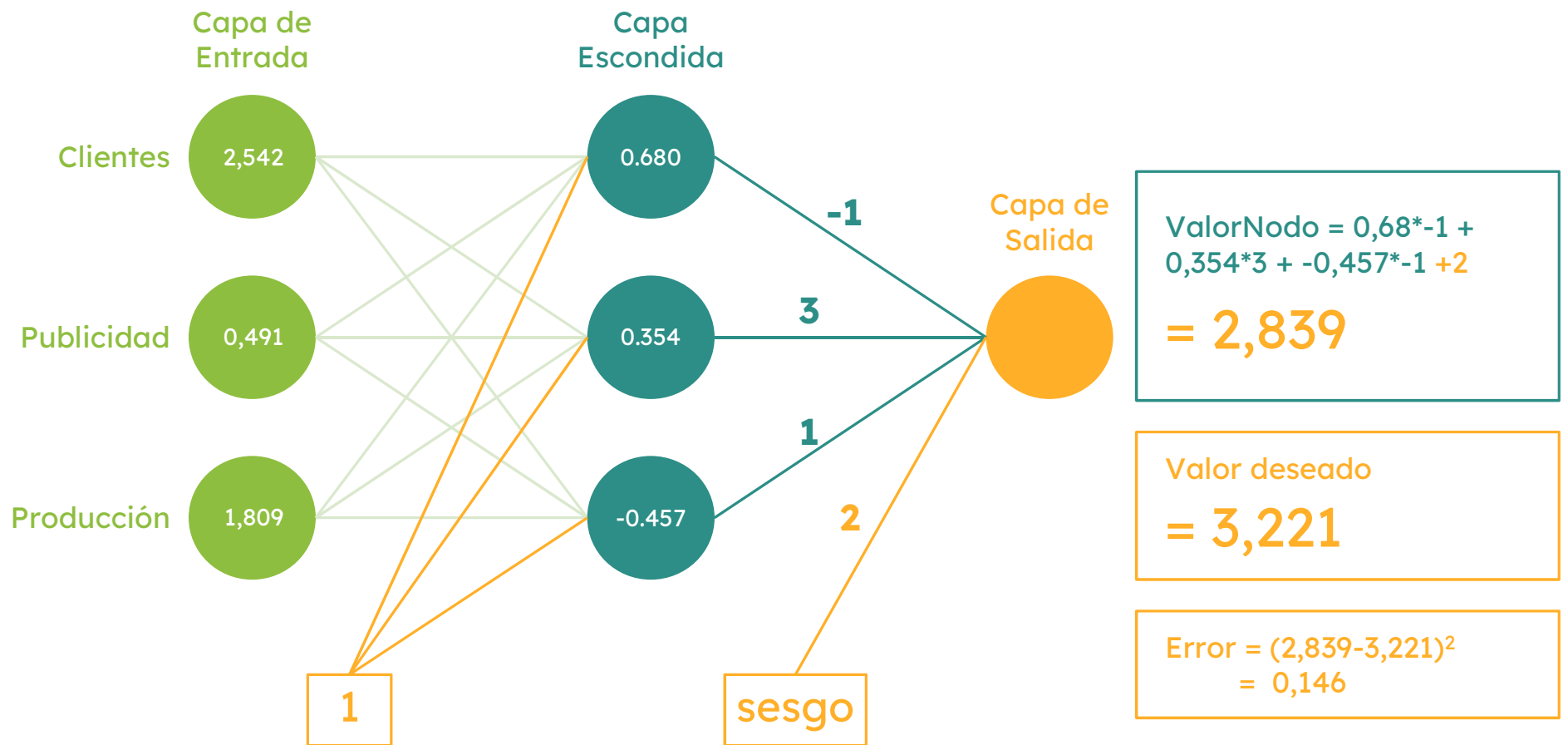
Cada nodo es **una función no lineal de una combinación lineal**.

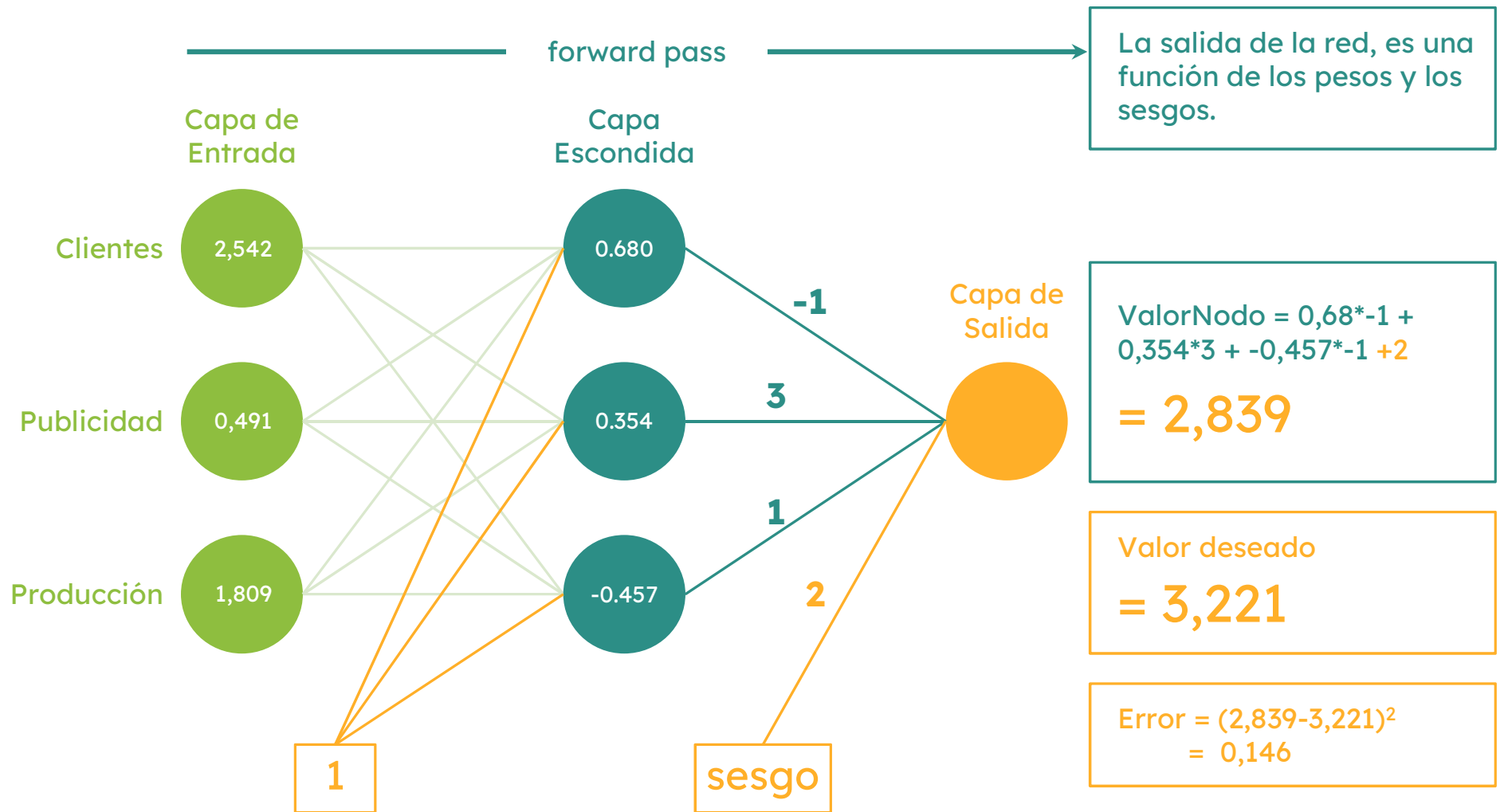
La función no lineal se conoce como **función de activación**.



...Y ahora computamos la capa de salida









Primera pregunta

¿Cómo ajustamos los pesos y sesgos para disminuir el error?

Gradiente descendiente

$$\text{peso} \leftarrow \text{peso} - \eta \frac{\partial \text{Error}}{\partial \text{peso}}$$

$$\text{sesgo} \leftarrow \text{sesgo} - \eta \frac{\partial \text{Error}}{\partial \text{sesgo}}$$

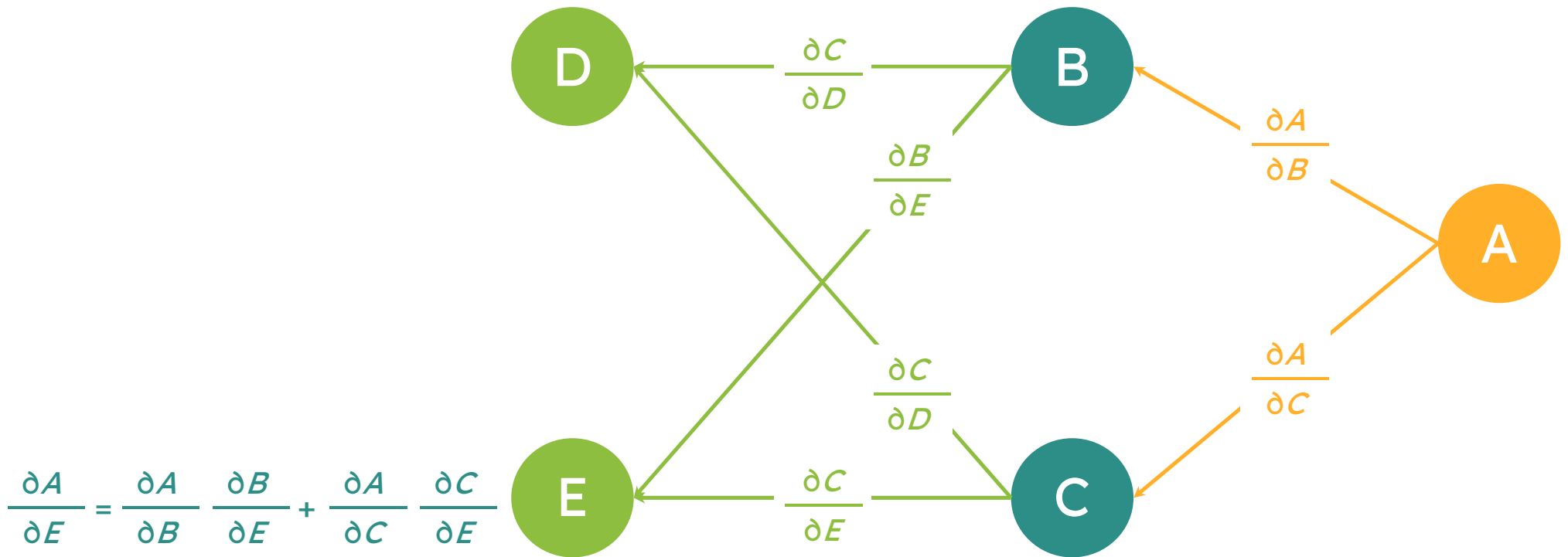
$\eta \in (0,1)$ es la tasa de aprendizaje



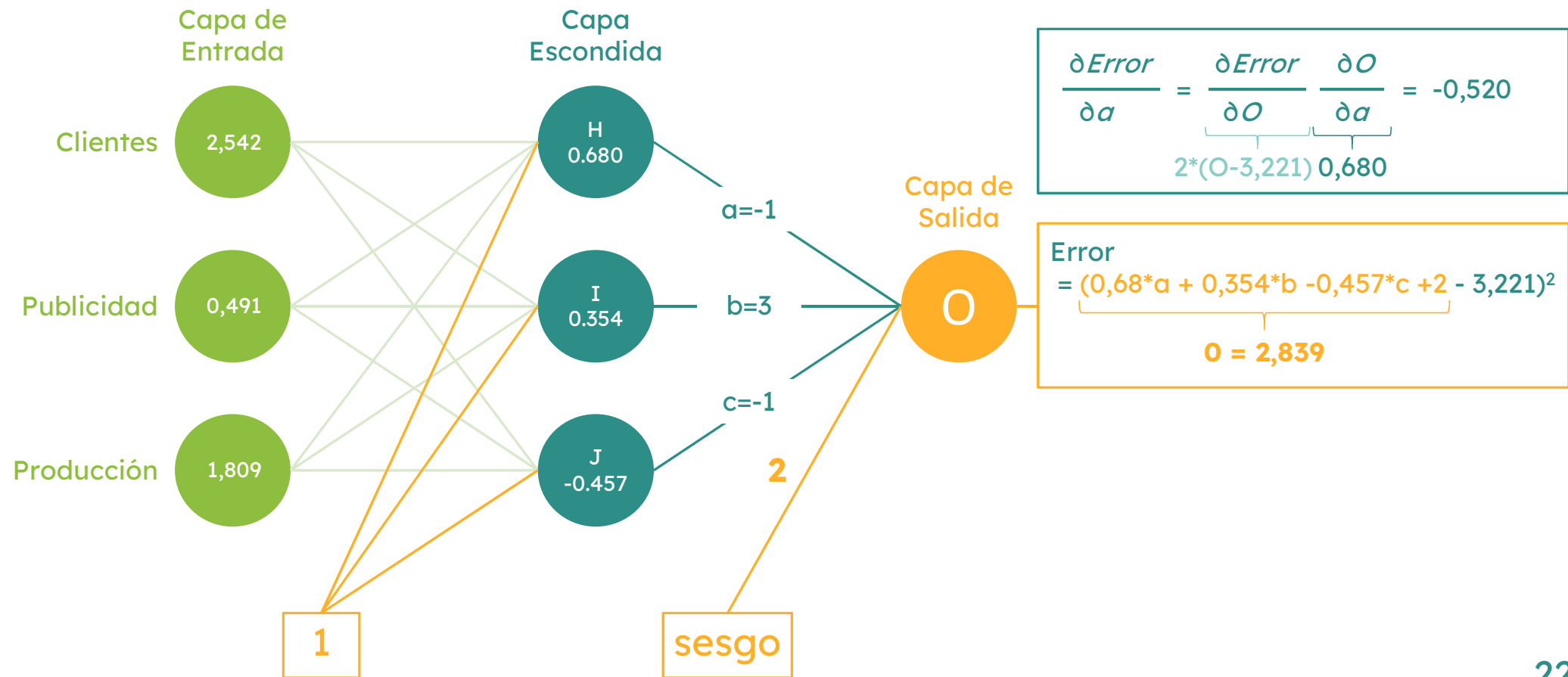
Segunda pregunta

¿Cómo calculamos esa derivada parcial?

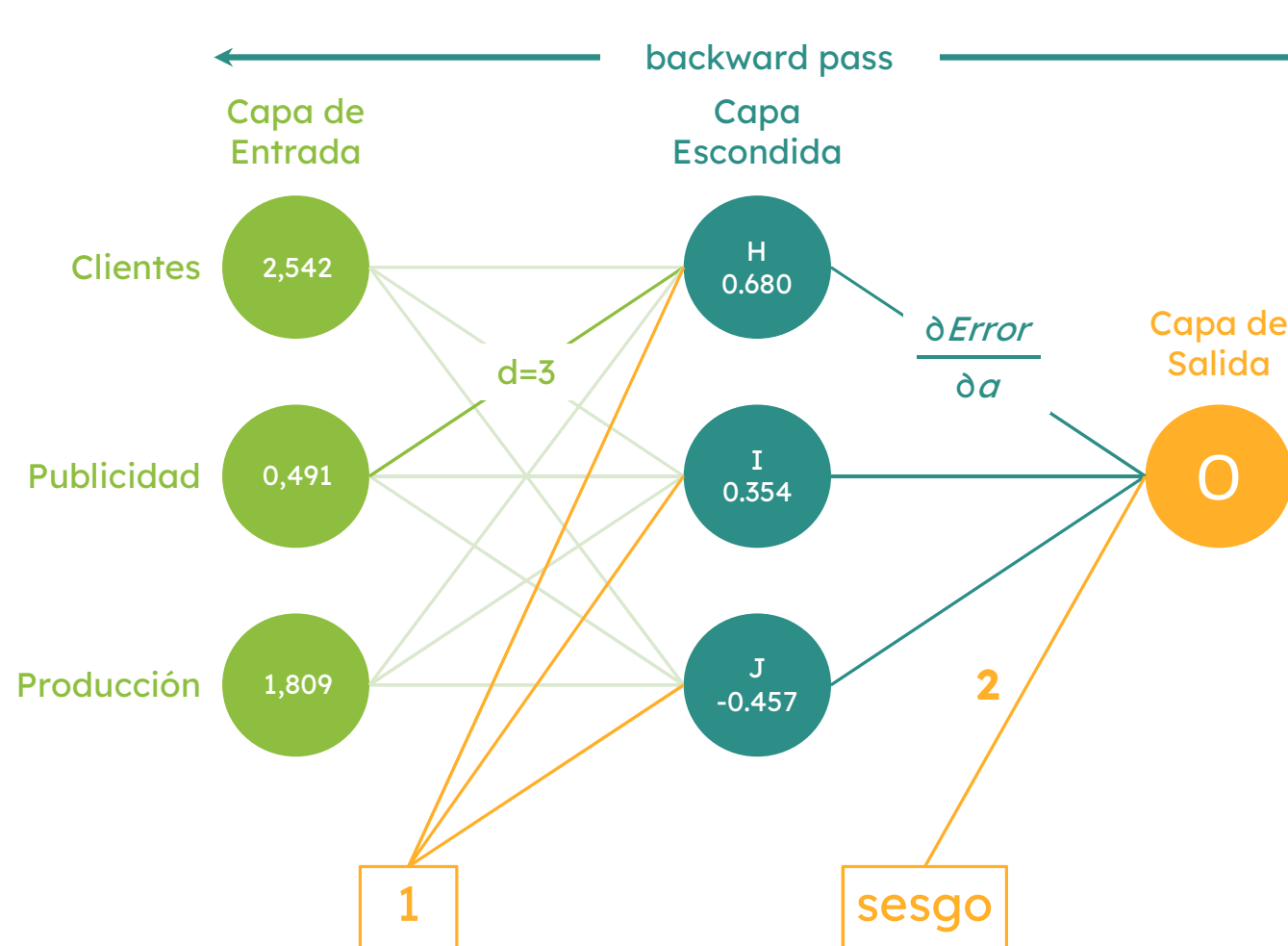
Regla de la cadena



Calculando derivadas de la capa de salida



Calculando derivadas de la capa escondida



Este algoritmo se conoce como **backpropagation** ya que las derivadas se calculan desde la capa de salida hasta la capa de entrada.

$$\frac{\partial Error}{\partial d} = \underbrace{\frac{\partial Error}{\partial a}}_{-0.52} \underbrace{\frac{\partial a}{\partial H}}_{\tanh'(0,83)=0,070} \underbrace{\frac{\partial H}{\partial d}}_{0,491} = -0,02$$



Esto se puede implementar fácilmente con librerías

```
[1]: import tensorflow as tf
      from keras.models import Sequential
      from keras.layers import Dense
      from keras.callbacks import History
      from keras.optimizers import SGD
```

```
[2]: hist = History()
      model = Sequential()
      model.add(Dense(3, input_dim=3, activation='tanh'))

      model.add(Dense(3, activation='tanh'))

      model.add(Dense(1, activation='linear'))

      model.compile(optimizer=SGD(learning_rate=0.001), loss='mse')
      model.summary()
```

Model: "sequential"

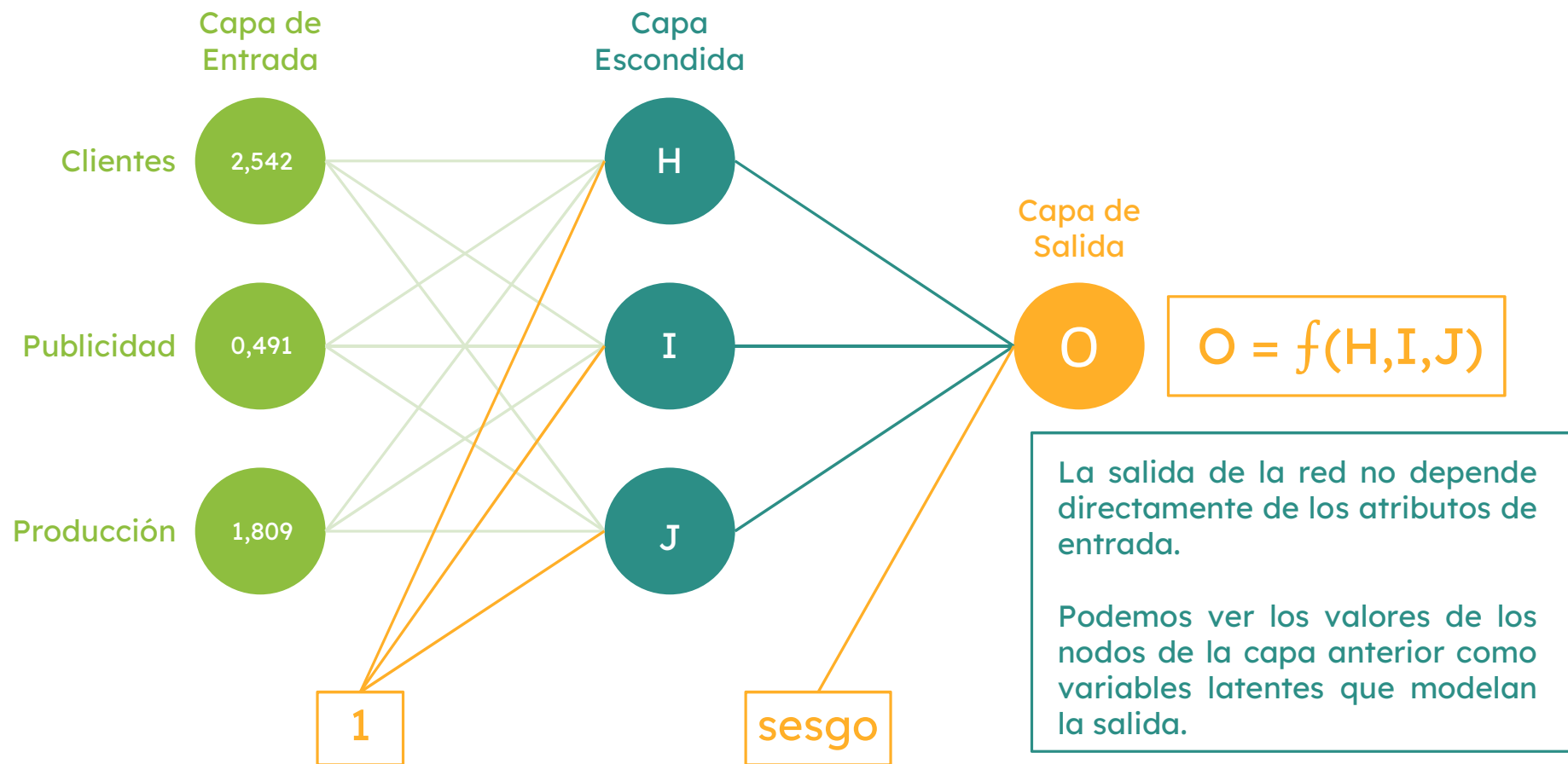
Layer (type)	Output Shape	Param #
dense (Dense)	(None, 3)	12
dense_1 (Dense)	(None, 3)	12
dense_2 (Dense)	(None, 1)	4

```
=====
Total params: 28
Trainable params: 28
Non-trainable params: 0
=====
```

```
[ ]: history = model.fit(Xm[train], ym[train], validation_data=(Xm[val], ym[val]),
                        epochs= 100, batch_size=1)
```



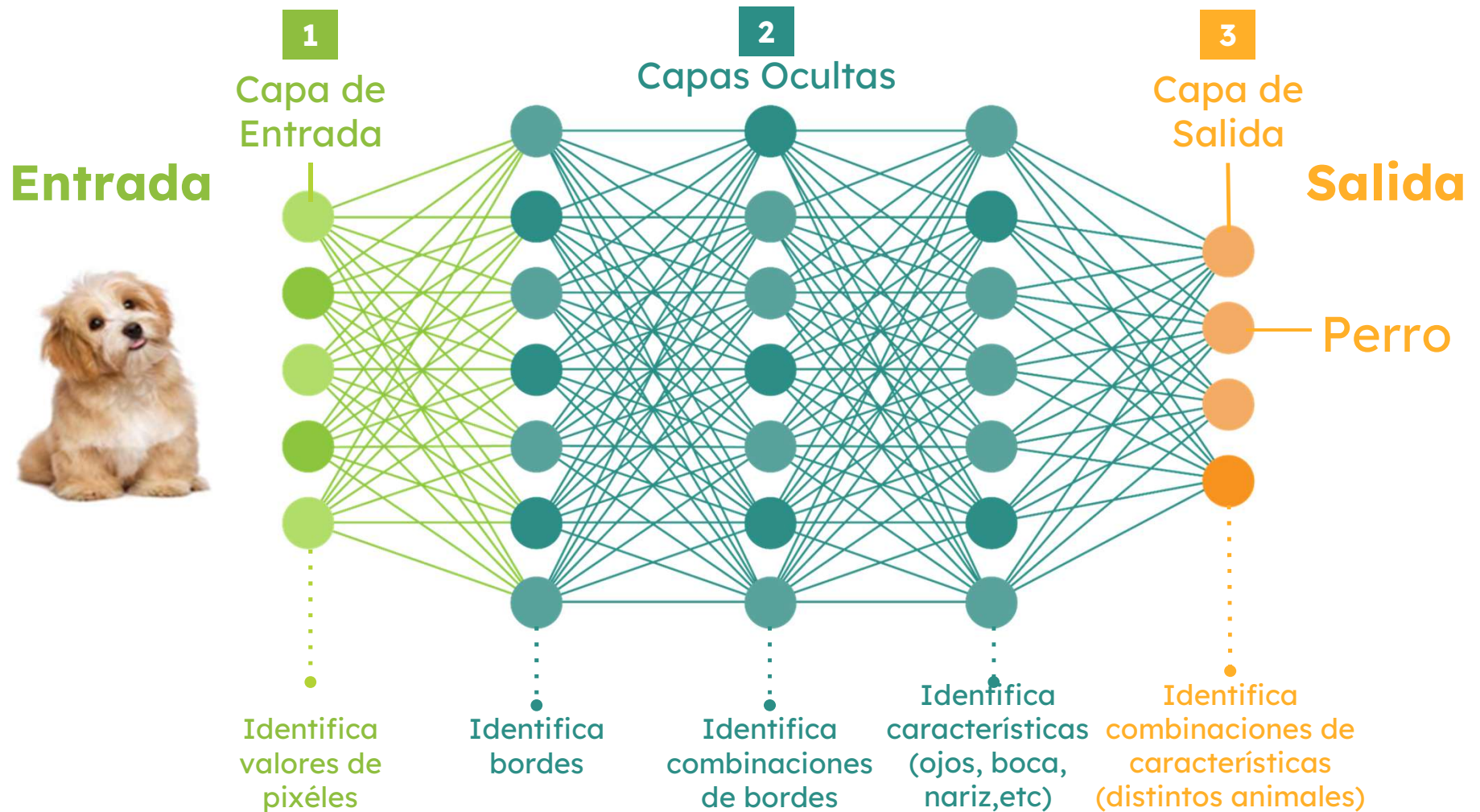
Repensando las redes neuronales



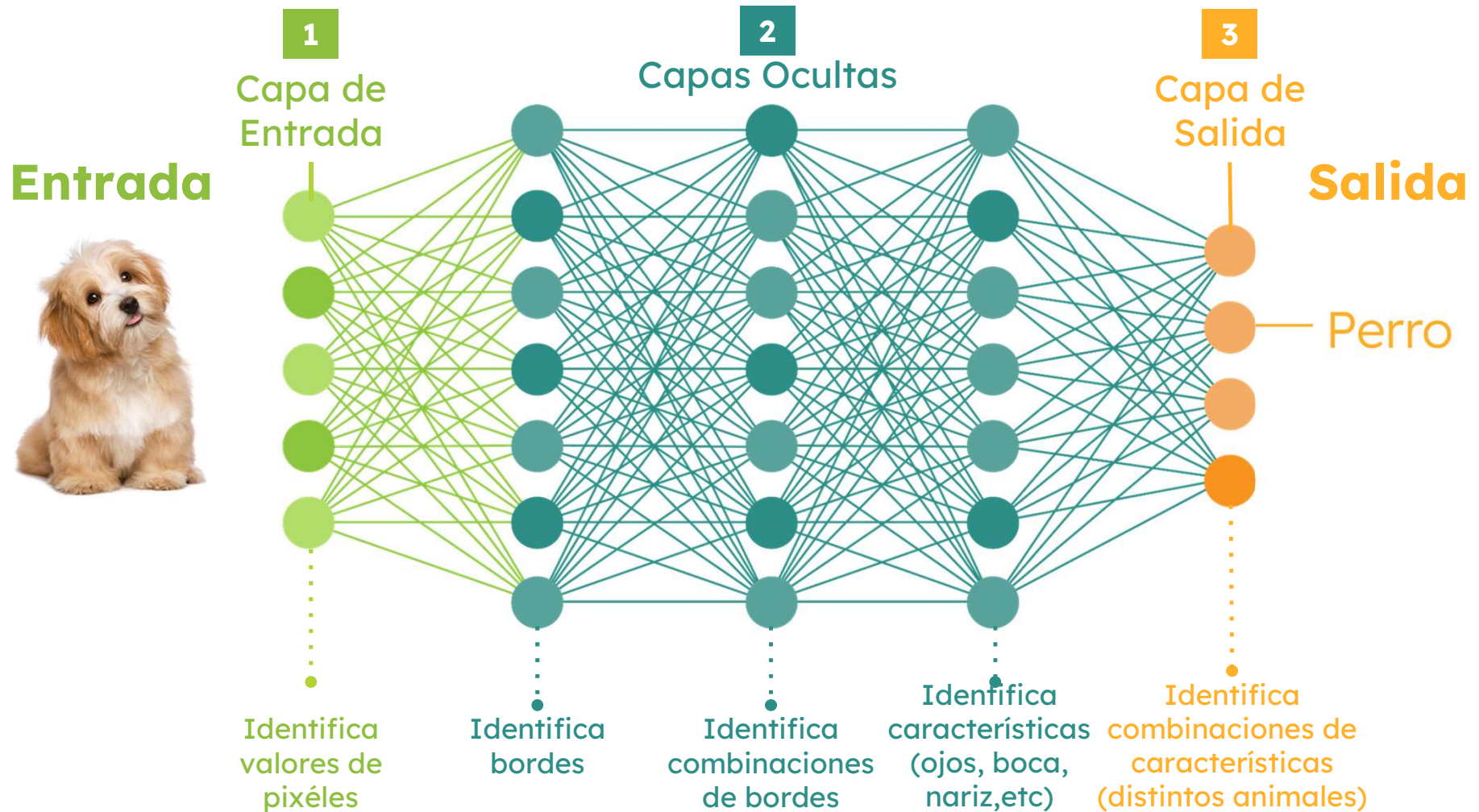


Redes Neuronales Profundas

Esquema simplificado de red neuronal profunda



Esquema simplificado de red neuronal profunda



Diapositiva 36

- 1 este gif puede ser con loop?
Carlos Valle Vidal; 13-09-2022

Podemos agregar más capas...

Al agregar más capas, podemos aprender una **jerarquía de atributos**, esto nos permite reciclar los atributos de las capas anteriores y con ello podemos obtener **representaciones más compactas**.

