



# Deep Learning

## Fundamentos de la IA

**DuocUC**



ESCUELA DE  
INFORMÁTICA Y  
TELECOMUNICACIONES



Próxima clase entregaré los detalles del encargo!



## EVALUACIONES

P  
o  
n  
d  
e  
r  
a  
c  
i  
ó  
n

F  
e  
c  
h  
a  
s

Unidad 1  
(prueba)

15-04-23

10%

Unidad 2 (Encargo +  
Presentación)

06-05-23

40%

Unidad 2 (Encargo  
+ Presentación)

03-06-23

25%

Unidad 4  
(Encargo +  
Presentación)

01-07-23

25%

70%

Examen transversal (Encargo + Presentación) 08-07-23



30%

# Calendario

## MARZO

SM	LU	MA	MI	JU	VI	SA	DO
09			1	2	3	4	5
10	6	7	8	9	10	11	12
11	13	14	15	16	17	18	19
12	20	21	22	23	24	25	26
13	27	28	29	30	31		

## ABRIL

SM	LU	MA	MI	JU	VI	SA	DO
13						1	2
14	3	4	5	6	7		9
15	10	11	12	13	14	15	16
16	17	18	19	20	21	22	23
17	24	25	26	27	28		30

## MAYO

SM	LU	MA	MI	JU	VI	SA	DO
18	1	2	3	4	5	6	7
19	8	9	10	11	12	13	14
20	15	16	17	18	19	20	21
21	22	23	24	25	26	27	28
22	29	30	31				

## JUNIO

SM	LU	MA	MI	JU	VI	SA	DO
22				1	2	3	4
23	5	6	7	8	9	10	11
24	12	13	14	15	16	17	18
25	19	20	21	22	23	24	25
26	26	27	28	29	30		

## JULIO

SM	LU	MA	MI	JU	VI	SA	DO
26						1	2
27	3	4	5	6	7	8	9
28	10	11	12	13	14	15	16
29	17	18	19	20	21	22	23
30	24	25	26	27	28	29	30
31	31						

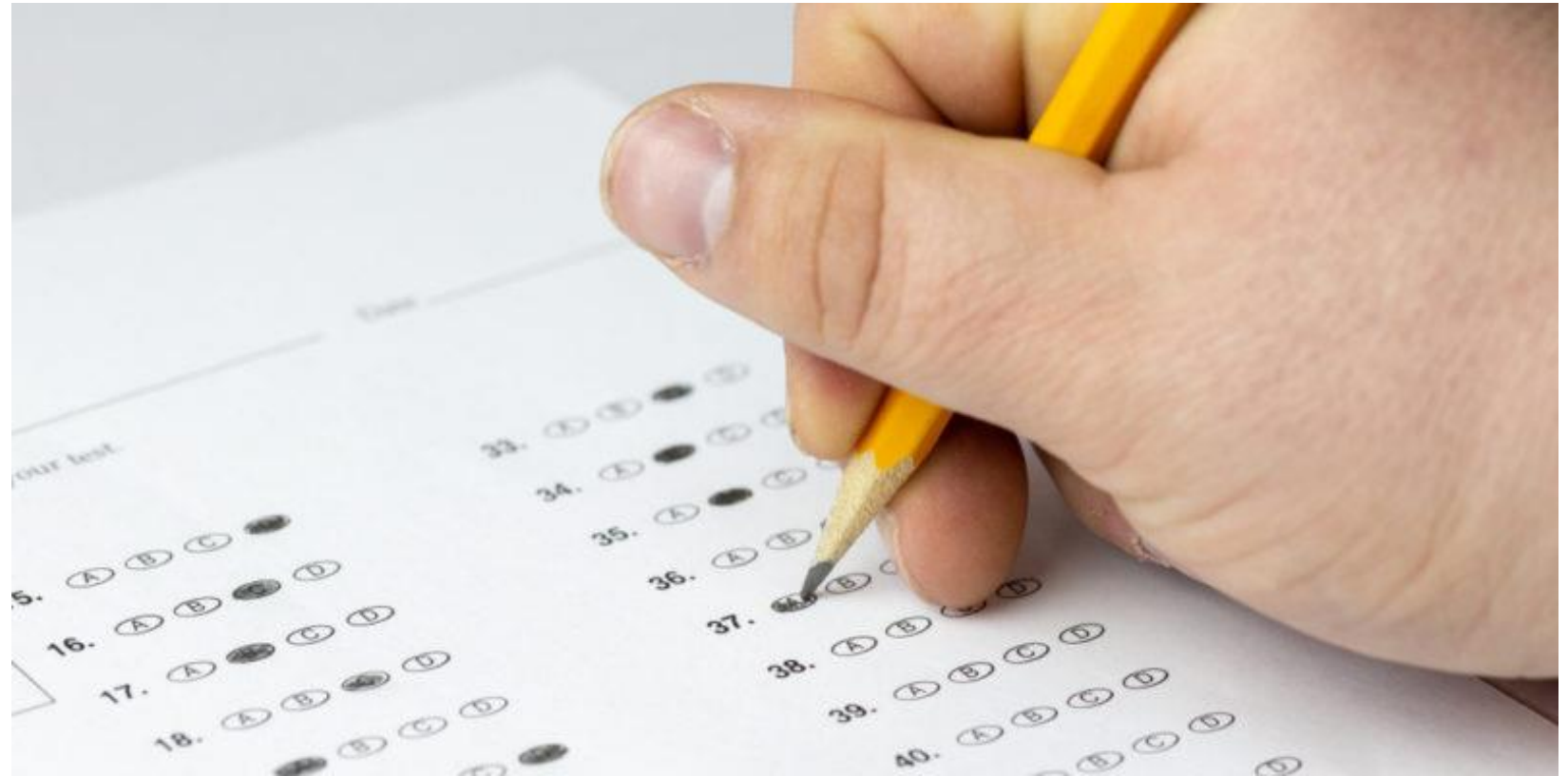
# Evaluación 1

La evaluación es individual

Está disponible en la  
plataforma de AVA.

El profesor compartirá el  
código para acceder a esta.

El tiempo es de 45 minutos.



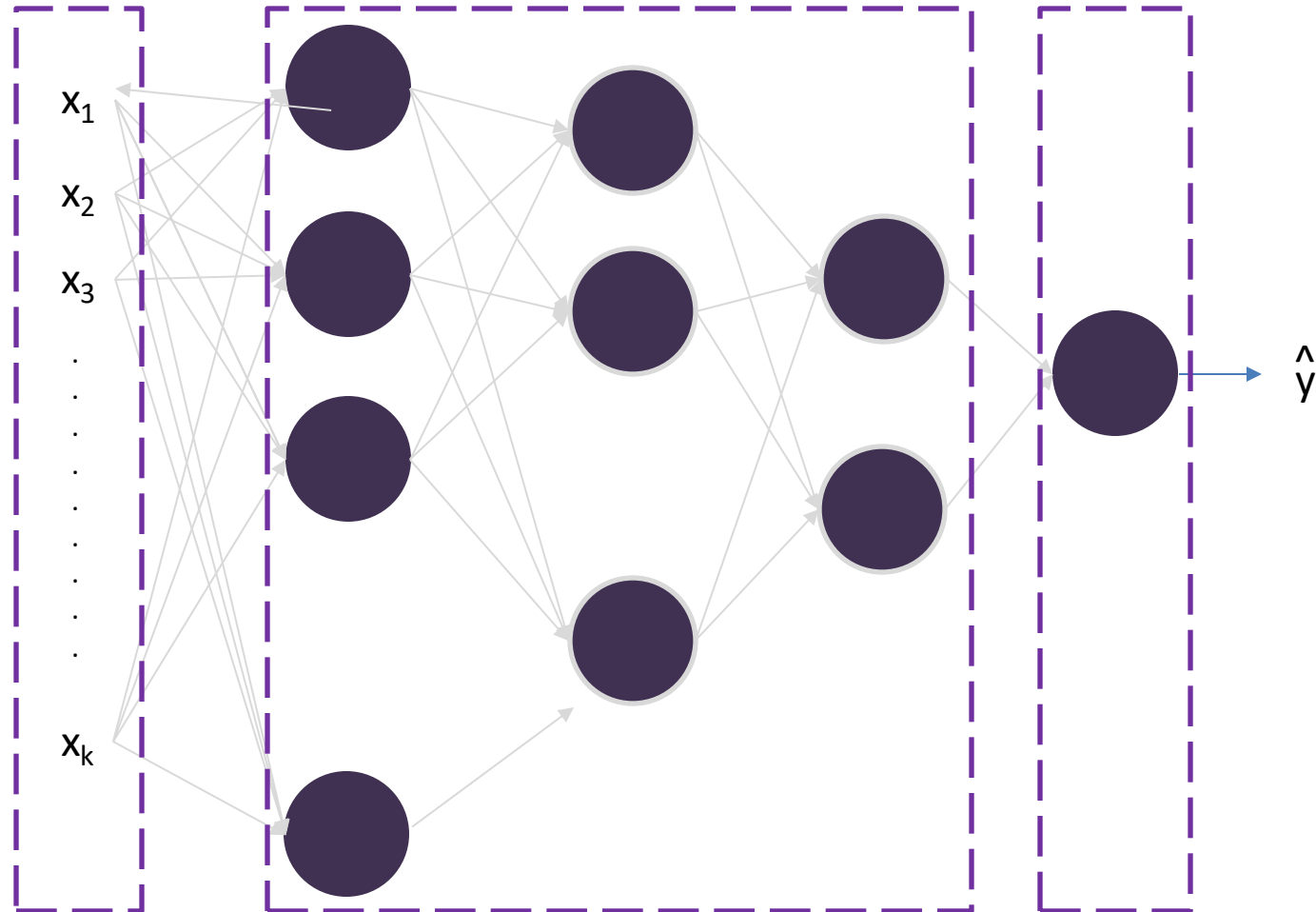




# Backpropagation

Ajuste y corrección del aprendizaje

## Función Error (Loss)



# Función Error (Loss)

Datos de entrada

Resultado esperado

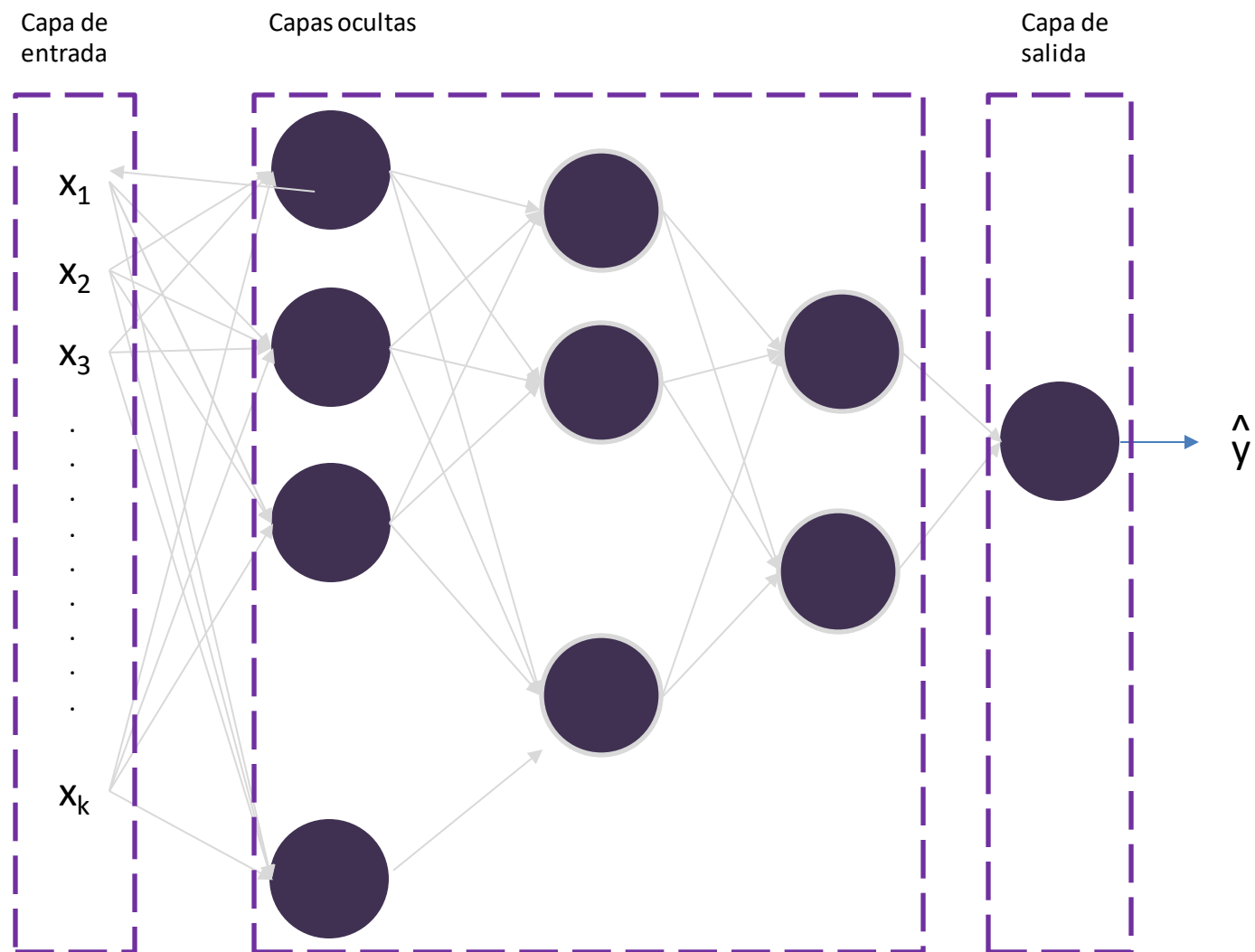
$x_i$

$y_i$

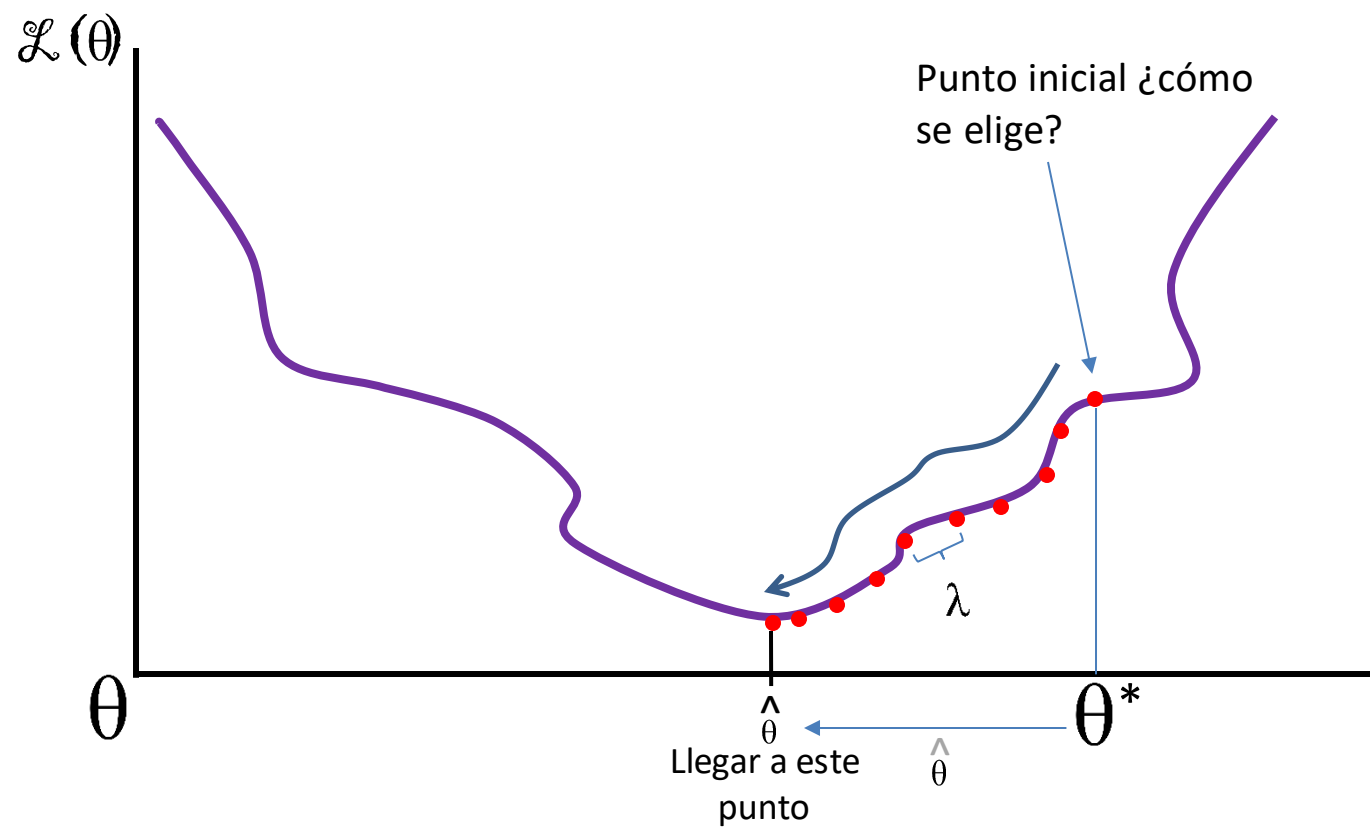
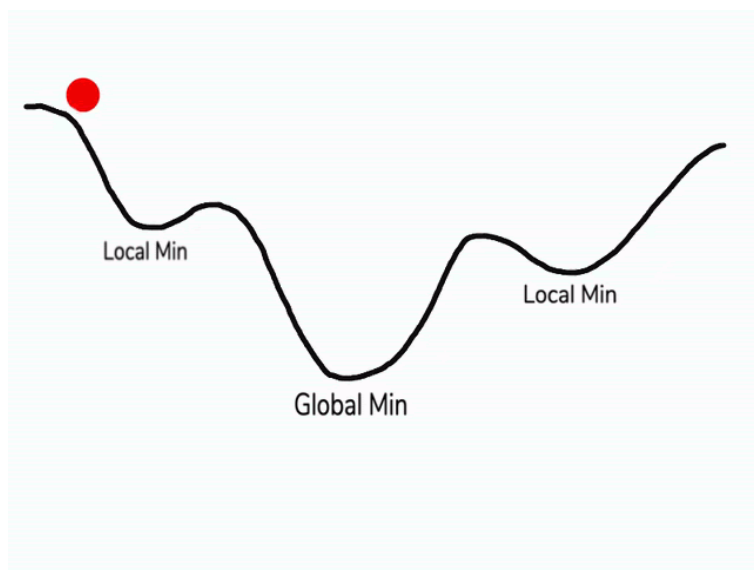
$\hat{y}_i$

Resultado que  
calculó la red

$$\mathcal{L} = \frac{1}{n} \sum_{i=1}^n \text{error}(\hat{y}^{(i)}; y^{(i)})$$



# Descenso del Gradiente





# Descenso del Gradiente

Diagram illustrating the Gradient Descent formula:

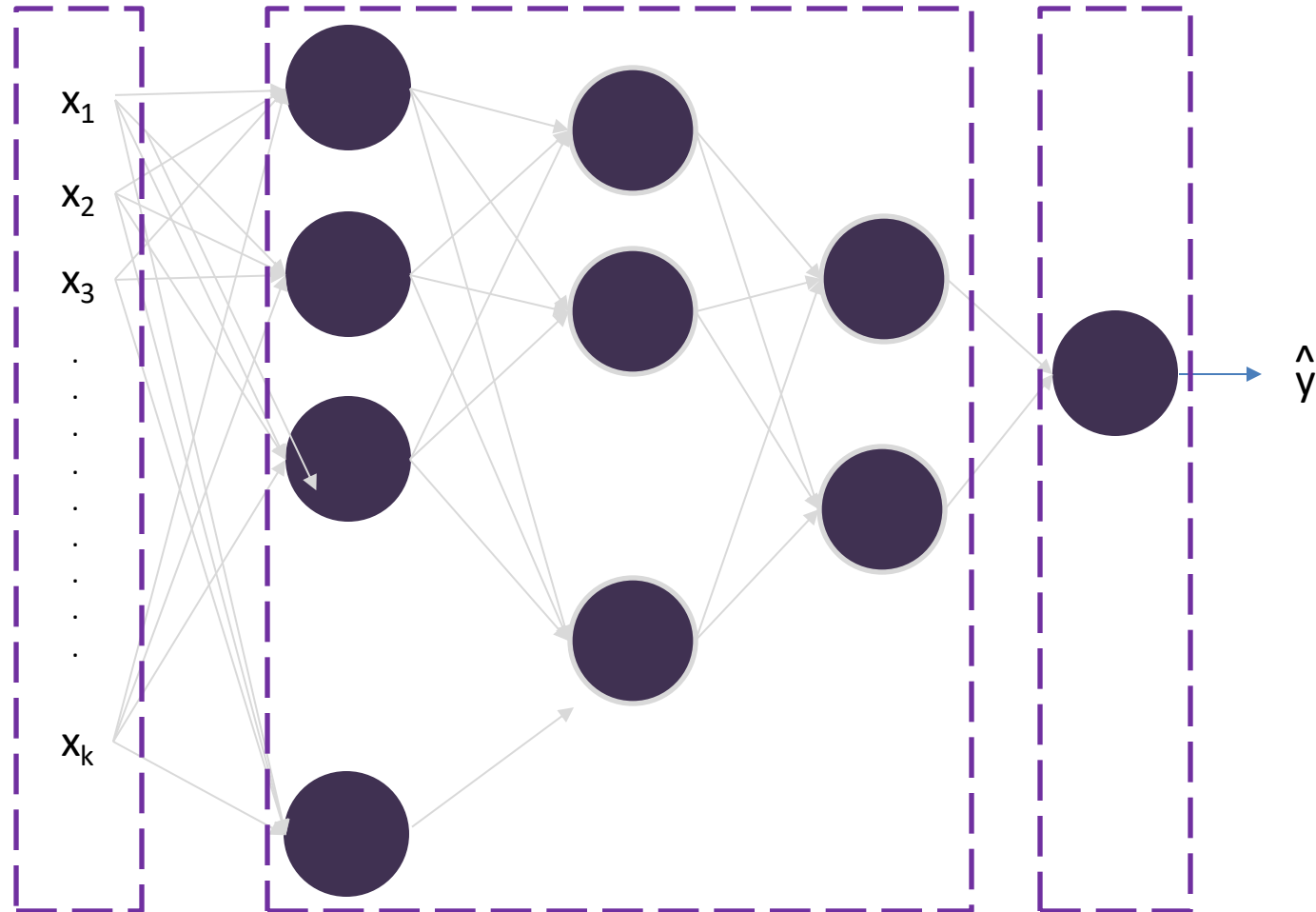
$$\hat{\theta} = \hat{\theta} - \lambda \left. \frac{d\mathcal{L}}{d\theta} \right|_{\hat{\theta}}$$

Annotations:

- Valor previo de qué? (Previous value of what?) points to  $\hat{\theta}$  on the right side of the equation.
- Por qué es negativo? (Why is it negative?) points to the minus sign.
- Valor actualizado de qué? (Updated value of what?) points to the entire right-hand side of the equation.
- Qué representa lambda? (What does lambda represent?) points to  $\lambda$ .
- Gradiente (Gradient) points to the derivative term  $\left. \frac{d\mathcal{L}}{d\theta} \right|_{\hat{\theta}}$ .

# Backpropagation

Composición de funciones



# Backpropagation

Descenso del gradiente es un algoritmo que permite ajustar los parámetros (pesos y bias) y para esto usa derivadas parciales de los parámetros.

Para calcular derivadas DE MANERA EFICIENTE, se utiliza el algoritmo de backpropagation.

**\*\*El ajuste repetitivo de los parámetros se llama entrenamiento\*\***

Dos detalles destacables:

1.- Backpropagation hace un uso intensivo de la regla de la cadena para calcular las derivadas parciales.

*La regla de la cadena es una norma de la derivación que nos dice que, teniendo una variable  $y$  que depende de  $u$ , y si esta depende a la variable  $x$ , entonces*

*la razón de cambio de  $y$  respecto a  $x$  puede estimarse como el producto de la derivada de  $y$  con respecto a  $u$  por la derivada de  $u$  respecto a  $x$ .*

$$y = f(u)$$

$$u = f(x)$$

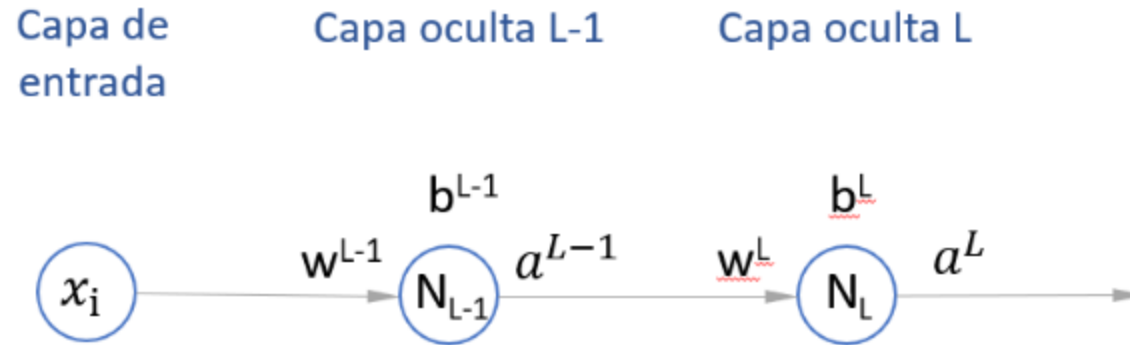
$$y = f(f(x))$$

$$\frac{dy}{dx} = \frac{dy}{du} \times \frac{du}{dx}$$

2.- Backpropagation puede ser utilizado en otras aplicaciones, en redes neuronales es uno de los usos que se le da. En general, backpropagation es un algoritmo matemático para calcular derivadas de manera eficiente.



# Backpropagation



Vamos a llamar a la segunda capa oculta (la más próxima a la salida de la red) **L** y a la primera capa, **L-1**.

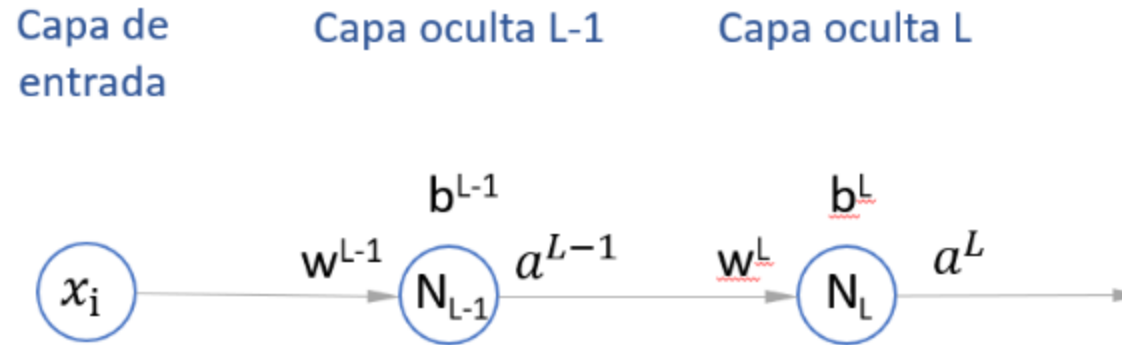
El comportamiento de la neurona artificial de la capa L (la neurona  $N_L$ ) viene determinado por el peso del enlace que la une con el valor devuelto por la capa anterior  $a^{L-1}$ ,  $w^L$ , y por el bias a añadir,  $b^L$ .

La derivada parcial de la función de coste con respecto a estos dos parámetros,  $w^L$  y  $b^L$ :

$$\frac{\partial C}{\partial w^L} \quad \frac{\partial C}{\partial b^L}$$



# Backpropagation



Vamos a llamar a la segunda capa oculta (la más próxima a la salida de la red) **L** y a la primera capa, **L-1**.

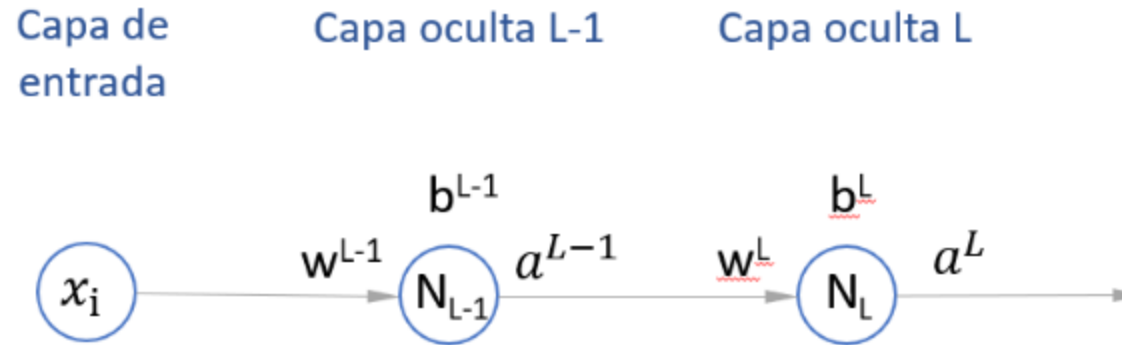
En estas expresiones  $C$  es la función de coste resultante de analizar el comportamiento de la red neuronal con todas las muestras del conjunto de entrenamiento. Para una de las muestras, la  $i$ -ésima, las expresiones anteriores vendrían dadas por:

Derivadas parciales de una muestra en concreto

$$\frac{\partial C_i}{\partial w^L} \quad \frac{\partial C_i}{\partial b^L}$$

Supongamos, para concretar, que la función de coste viene dada por el error cuadrático medio comentado:

# Backpropagation



Vamos a llamar a la segunda capa oculta (la más próxima a la salida de la red) **L** y a la primera capa, **L-1**.

En esta expresión **y** es el valor que debería devolver la red para la muestra i-ésima (es decir, la etiqueta asociada a dicha muestra) y  $\hat{y}$  es el valor que en realidad devuelve la red para la muestra en cuestión. Este valor viene dado, como ya sabemos, por la multiplicación del valor devuelto por la neurona anterior por peso del enlace, más el bias, y pasando el resultado de esta combinación lineal por la función de activación  $\sigma$

$$\hat{y} = \sigma(w^L a^{L-1} + b^L)$$

Es decir, la función de coste puede expresarse también como:

$$C_i = (y - \sigma(w^L a^{L-1} + b^L))^2$$

Por comodidad, llamemos  $z^L$  a la combinación lineal  $w^L a^{L-1} + b^L$ :

$$z^L = w^L a^{L-1} + b^L$$

De forma que:

$$\hat{y} = \sigma(z^L)$$

# Backpropagation

Pues bien, el cálculo nos dice que la derivada de la función  $C_i$  (El error) con respecto a  $w^L$  (es decir, cómo varía  $C_i$  cuando variamos  $w^L$ ) coincide con la derivada de  $C_i$  con respecto a  $\sigma(\hat{y})$ , multiplicado por la derivada de  $\sigma$  respecto de  $z^L$ , multiplicado por la derivada de  $z^L$  con respecto a  $w^L$ . Esto es lo que se llama regla de la cadena:

$$\frac{\partial C_i}{\partial w^L} = \frac{\partial C_i}{\partial \sigma} \frac{\partial \sigma}{\partial z^L} \frac{\partial z^L}{\partial w^L}$$



La buena noticia es que todas estas derivadas parciales son fácilmente calculables:

- La derivada parcial de  $C_i$  con respecto a  $\sigma$  es, para la función de coste escogida,  $2(y - \sigma)$ .
- La derivada parcial de  $\sigma$  con respecto a  $z^L$  es la derivada de la función de activación con la que estemos trabajando.
- La derivada parcial de  $z^L$  con respecto a  $w^L$  es  $a^{L-1}$ .

De modo análogo, la derivada parcial de  $C_i$  con respecto a  $b^L$  vendría dada por una expresión semejante a la anterior:

$$\frac{\partial C_i}{\partial b^L} = \frac{\partial C_i}{\partial \sigma} \frac{\partial \sigma}{\partial z^L} \frac{\partial z^L}{\partial b^L}$$

...lo que resulta, incluso, más fácil de calcular pues la derivada parcial de  $z^L$  con respecto a  $b^L$  es 1.

# Backpropagation

Sin entrar más profundamente en las matemáticas, el hecho es que aplicando la regla de la cadena, podemos ir de atrás hacia delante calculando las derivadas parciales de la función de coste con respecto a todos los pesos y los bias. A medida que vamos calculando derivadas parciales, las de la capa oculta a calcular a continuación (capa más próxima a la entrada de la red pues recordemos que estamos yendo de atrás adelante) hará uso de las derivadas parciales ya calculadas, por lo que el cálculo del gradiente resulta relativamente sencillo.



# Video



[https://www.youtube.com/watch?v=eNlqz\\_noix8](https://www.youtube.com/watch?v=eNlqz_noix8)



<https://www.youtube.com/watch?v=M5QHwkkHgAA>

## ACTIVIDAD DESCENSO DEL GRADIENTE Y BACKPROPAGATION



**Pregunta 1.** Las redes neuronales que hemos visto hasta ahora, pueden estar compuestas de billones de parámetros que necesitan ser actualizados para encontrar la combinación perfecta que resuelve el problema. Imaginarán, que no es una tarea sencilla, ni barata, desde el punto de vista de cómputo y tiempo. Según lo anterior, **¿Qué es backpropagation y cómo se aplica?** Fundamenta tu respuesta.

**Pregunta 2.** Basados en los diferentes ajustes a las redes y sus técnicas, **¿Qué es el descenso del gradiente? y ¿Cuál es la diferencia con backpropagation?** Fundamenta tu respuesta.

**Pregunta 3.** Según lo aprendido hasta ahora, **¿Con qué se entrena la red: backpropagation o descenso del gradiente?**



Discuta con sus compañeros y responda las 3 preguntas.



# Evaluación 2

## Grupos

Grupo	Caso	Estudiantes
1	C	FRANCISCO LARA TRONCCI / AARON ARRIAGADA CARRASCO
2	B	JOAQUIN FRITZ ASTORGA / JAVIER NEGRETE SALAS
3	A	ROBERTO CIFUENTES ORREGO / GUSTAVO LUNA BRITO
4	B	ANDRES FLORES COFRE / DANILO TRONCOSO VARGAS
5	A	MICHELL MIRANDA MENDEZ
?	?	DENISSE ALCANTARA PINA / DIEGO CONCHA RAMOS

- Descargar hoja de respuestas desde AVA
- Generador de tablas en LaTeX online: <https://www.tablesgenerator.com/>

## Casos

- A. Fashion MNIST
- B. MNIST
- C. CIFAR-10