

Procesamiento de Lenguaje Natural

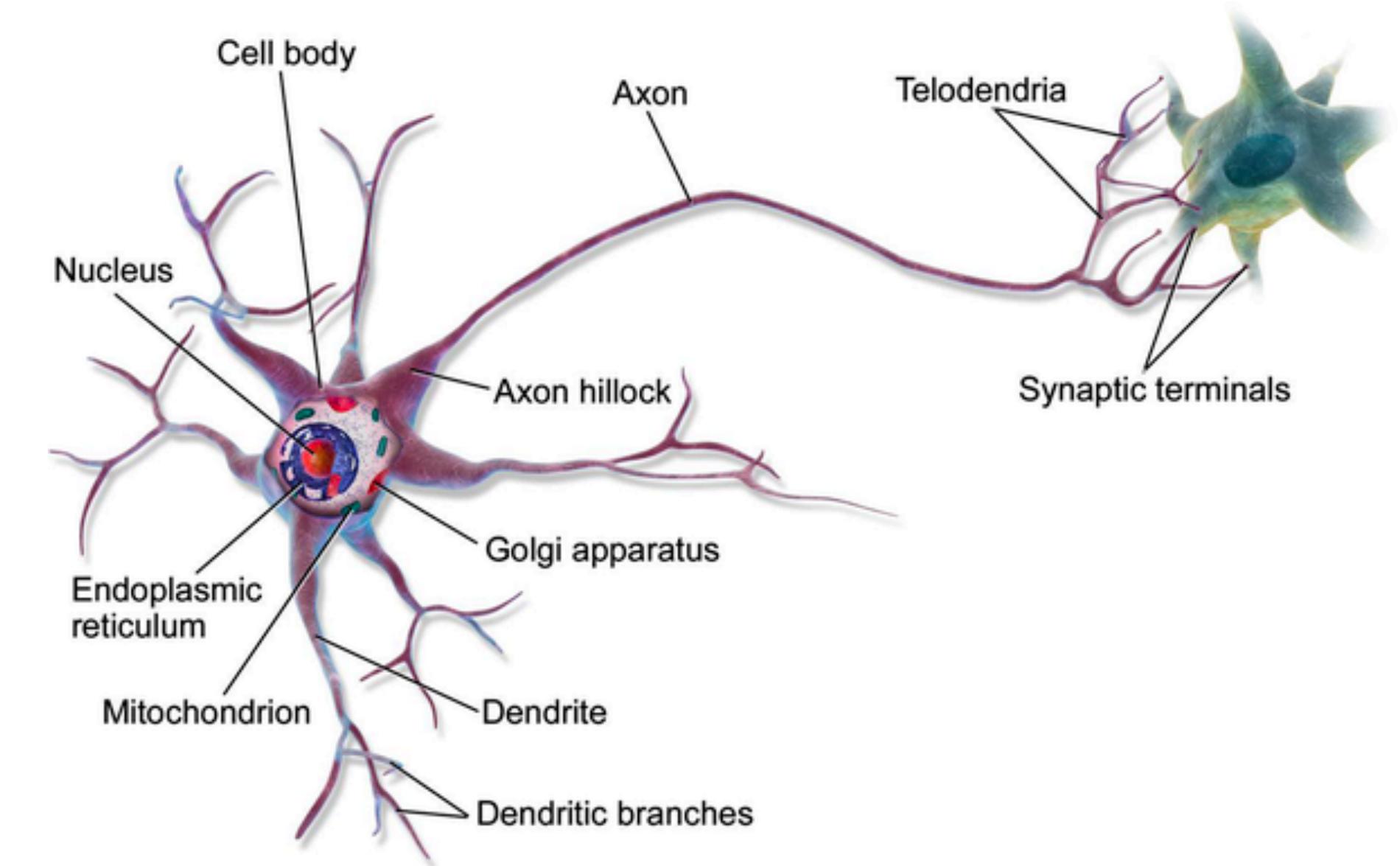
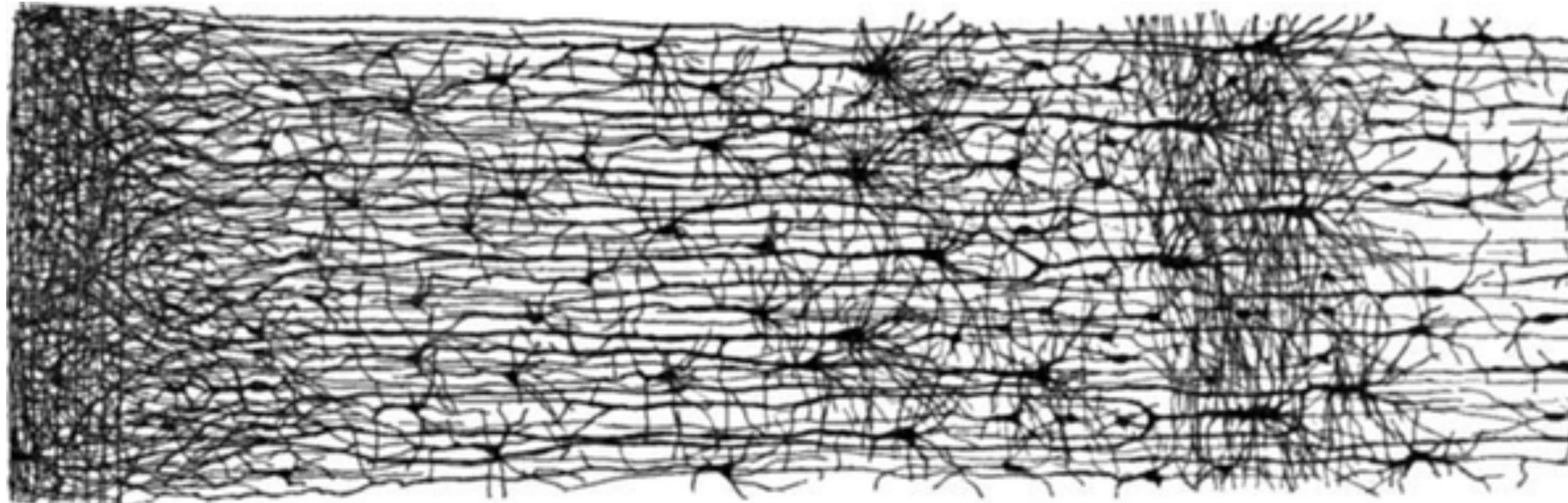
Tópicos Avanzados en Analítica
Maestría en Analítica para la Inteligencia de Negocios

Sergio Alberto Mora Pardo - H2 2023

Deep Learning in NLP

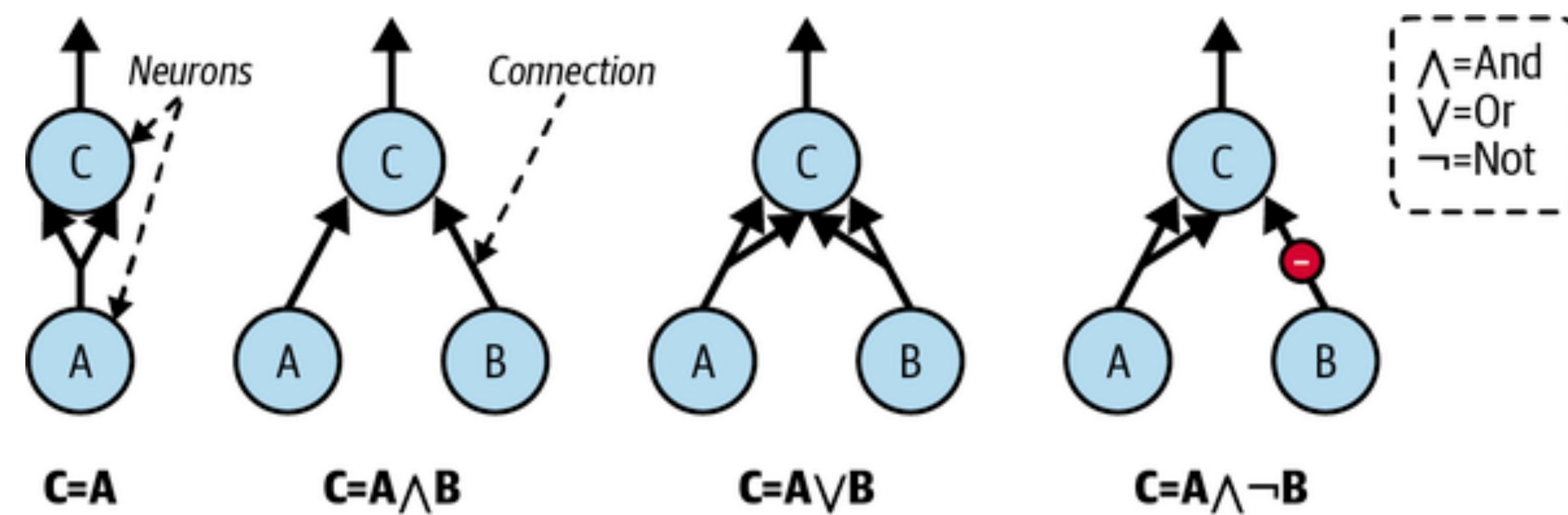
Intro Deep Learning

Biological Neurons



Intro Deep Learning

Biological Neurons



ANN que realiza cálculos lógicos simples

Intro Deep Learning

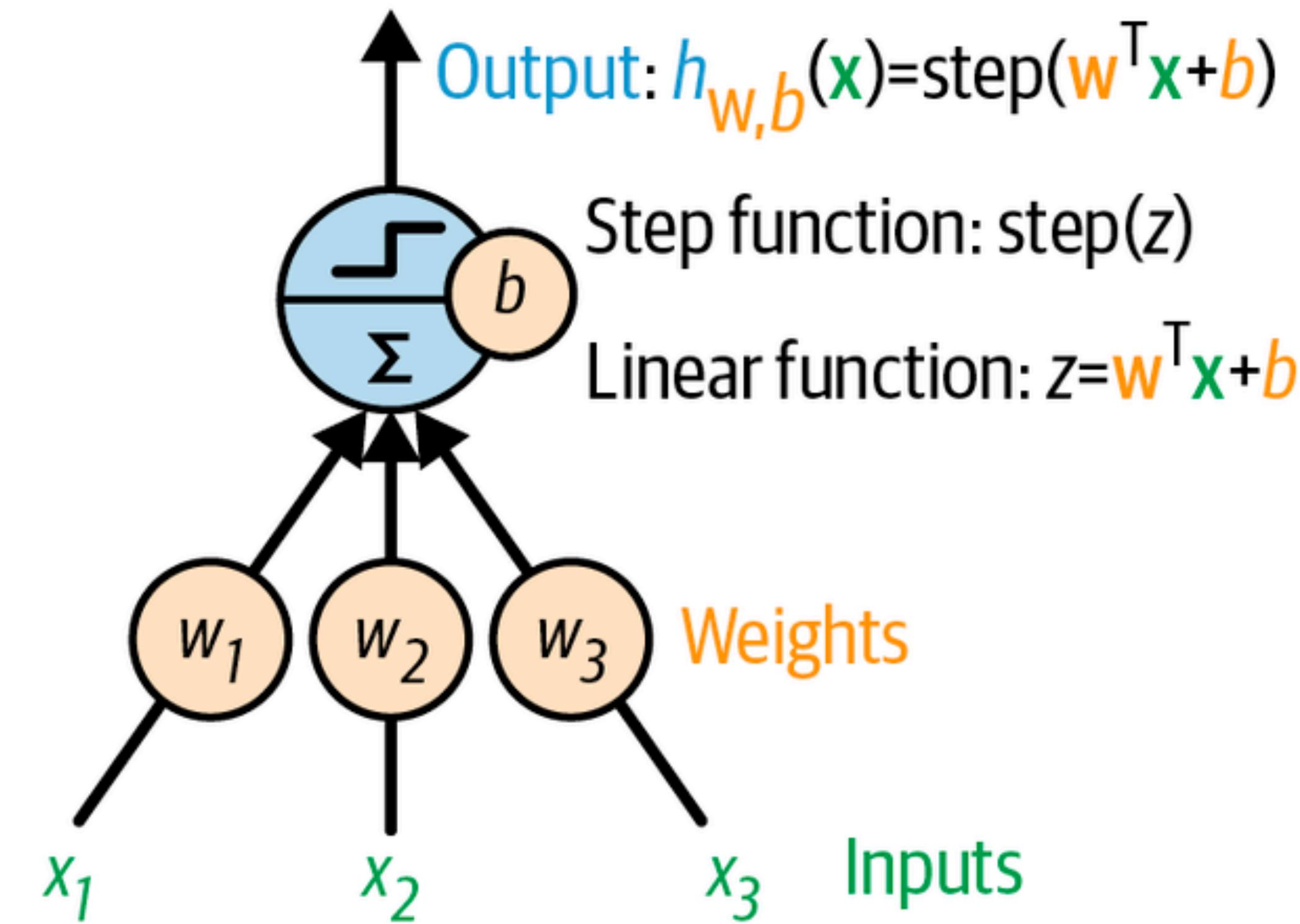
Perceptron

unidad lógica de umbral (TLU)

Las entradas y salidas son números



Luego aplica una función escalonada al resultado



1957 por Frank Rosenblatt

Intro Deep Learning

Perceptron

unidad lógica de umbral (TLU)

Las entradas y salidas son números



Luego aplica una función escalonada al resultado

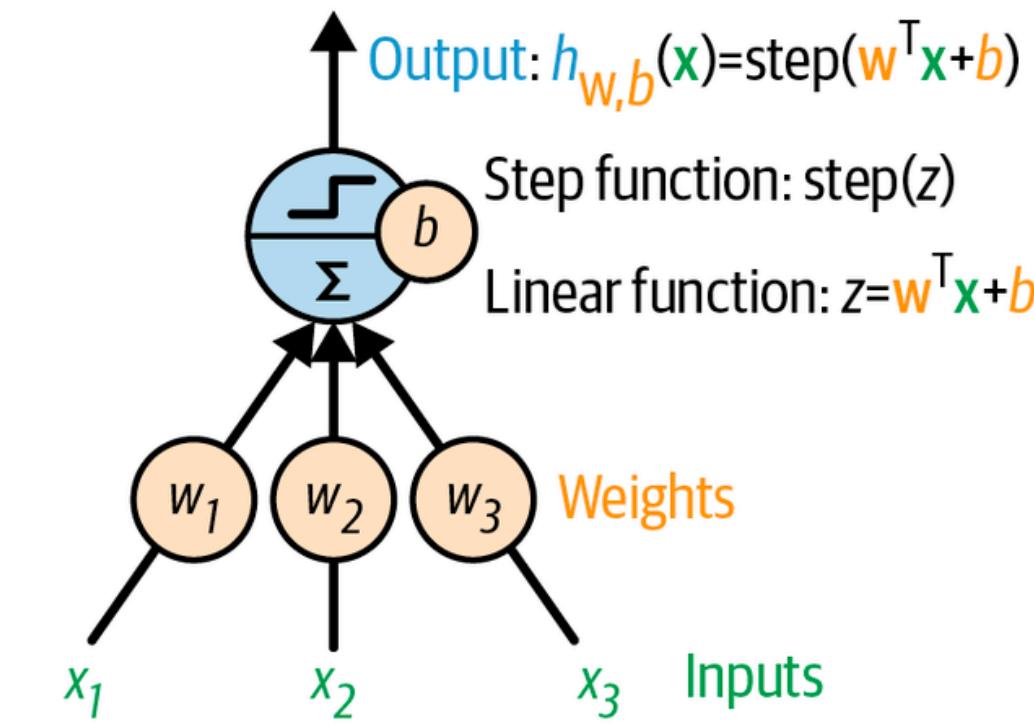
$$z = w_1 x_1 + w_2 x_2 + \dots + w_n x_n + b$$

$$Z = w^T x + b .$$

Función escalonada (h):

$$h_w(x) = \text{step}(z)$$

$$\text{heaviside}(z) = \begin{cases} 0 & \text{if } z < 0 \\ 1 & \text{if } z \geq 0 \end{cases}$$



$$\text{sgn}(z) = \begin{cases} -1 & \text{if } z < 0 \\ 0 & \text{if } z = 0 \\ +1 & \text{if } z > 0 \end{cases}$$

1957 por Frank Rosenblatt

Intro Deep Learning

Perceptron

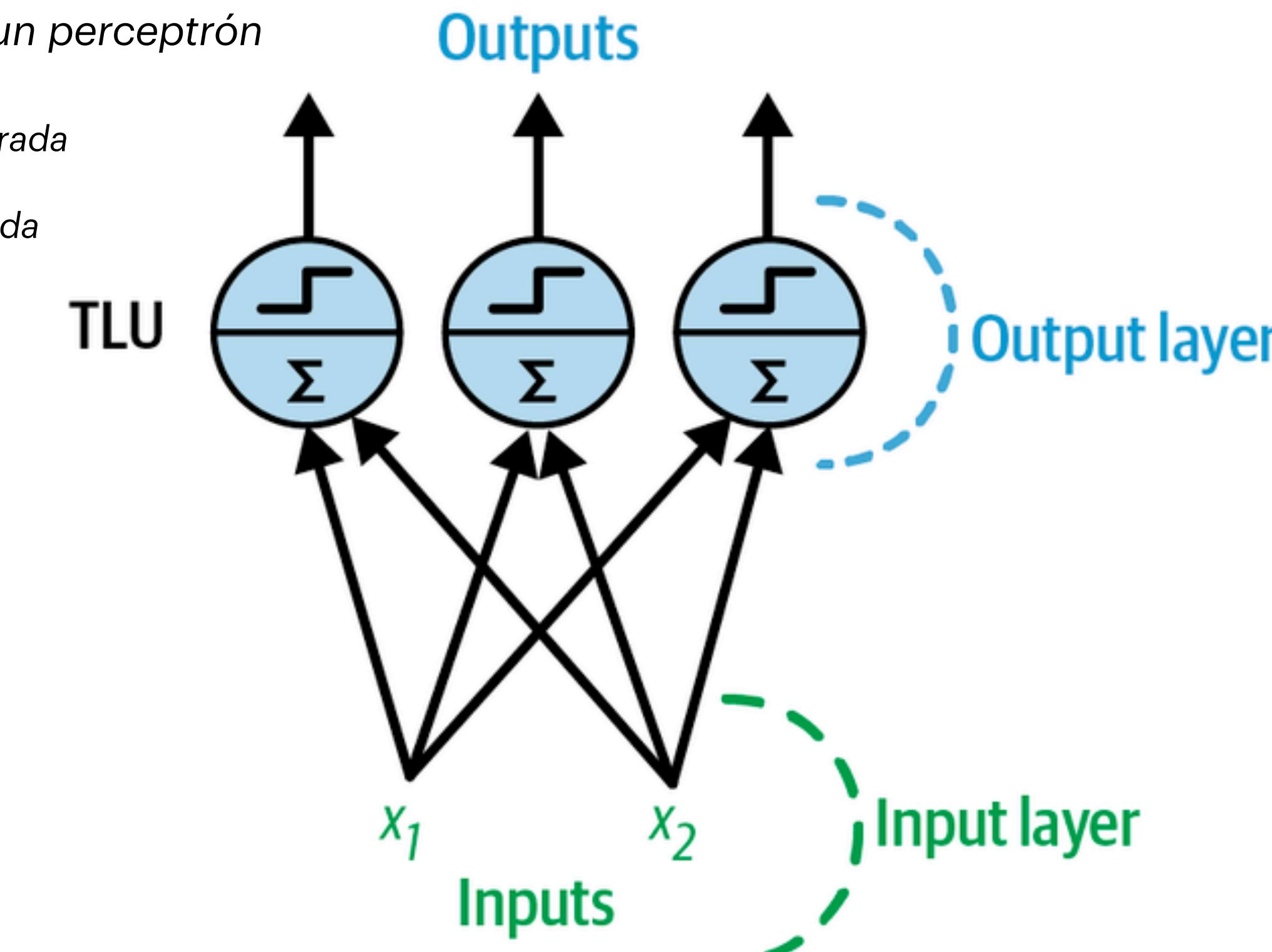
unidad lógica de umbral (TLU)

Las entradas y salidas son números

Luego aplica una función escalonada al resultado

Arquitectura de un perceptrón

- 2 neuronas de entrada
- 3 neuronas de salida



1957 por Frank Rosenblatt

Intro Deep Learning

Perceptron

unidad lógica de umbral (TLU)

Las entradas y salidas son números



Luego aplica una función escalonada al resultado

Arquitectura de un perceptrón

$$h_{\mathbf{W}, \mathbf{b}}(\mathbf{X}) = \phi(\mathbf{X}\mathbf{W} + \mathbf{b})$$

X representa la matriz de características de entrada.

W contiene todos los pesos de conexión

b contiene todos los términos de sesgo

ϕ se llama función de activación

Intro Deep Learning

Perceptron

unidad lógica de umbral (TLU)

Las entradas y salidas son números



Luego aplica una función escalonada al resultado

Arquitectura de un perceptrón

$$h_{\mathbf{W}, \mathbf{b}}(\mathbf{X}) = \phi(\mathbf{X}\mathbf{W} + \mathbf{b})$$

\mathbf{X} *Tiene una fila por instancia y una columna por característica.*

\mathbf{W} *Tiene una fila por entrada y una columna por neurona.*

\mathbf{b} *uno por neurona*

Intro Deep Learning

Perceptron

unidad lógica de umbral (TLU)

Arquitectura de un perceptrón con dos neuronas de entrada y tres de salida

¿Cómo aprendía el preceptron?

$$w_{i,j}^{(\text{next step})} = w_{i,j} + \eta(y_j - \hat{y}_j)x_i$$

$W_{i,j}$ es el peso de la conexión entre la i -ésima entrada y la j -ésima neurona

X_i es el i -ésimo valor de entrada de la instancia de entrenamiento actual.

\hat{y}_j es la salida de la j -ésima neurona de salida para la instancia de entrenamiento actual.

y_i es la salida objetivo de la j -ésima neurona de salida para la instancia de entrenamiento actual.

η es la tasa de aprendizaje

Intro Deep Learning

Perceptron

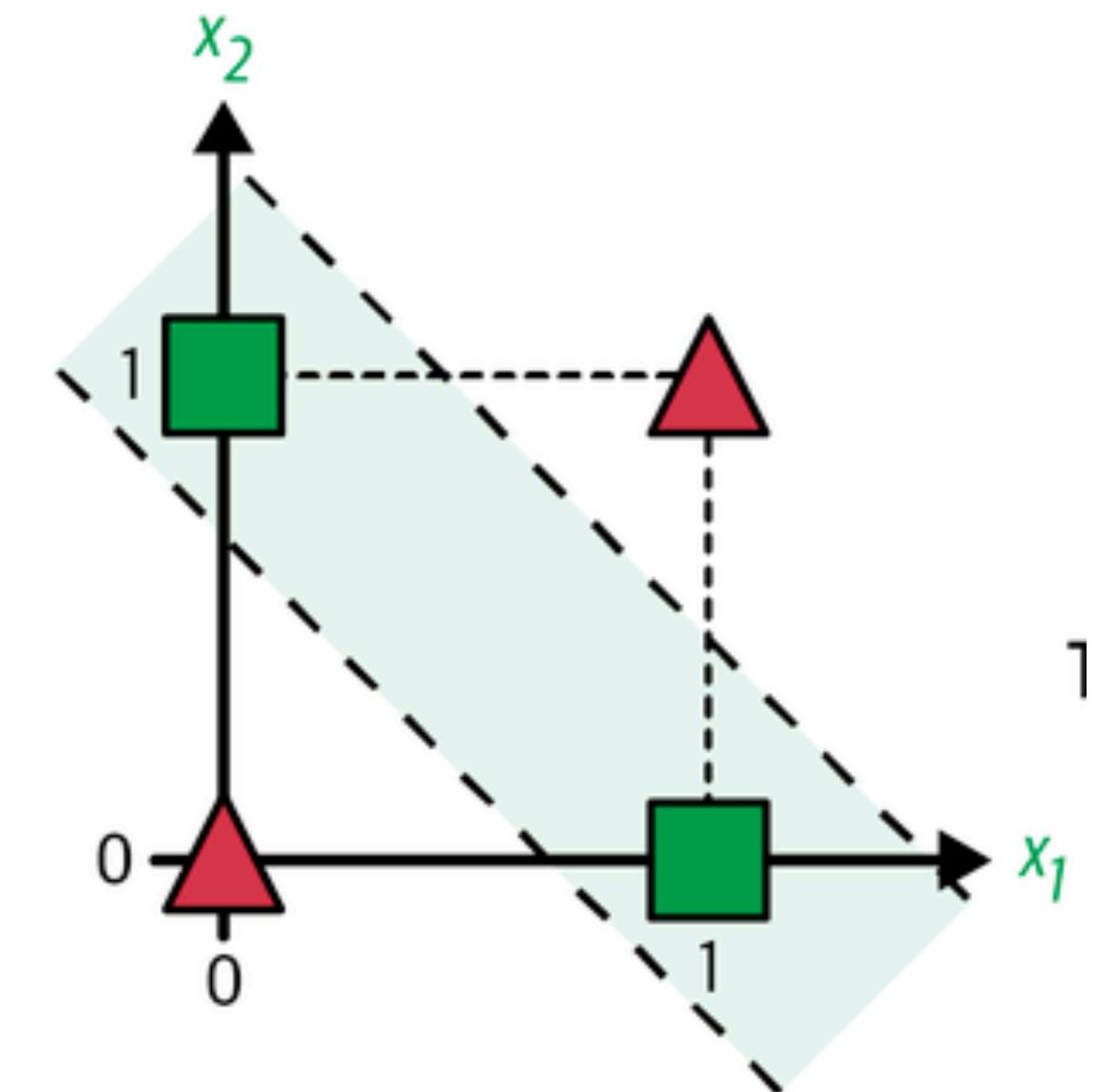
unidad lógica de umbral (TLU)

Arquitectura de un perceptrón con dos neuronas de entrada y tres de salida

1957 por Frank Rosenblatt

$$w_{i,j}(\text{nextstep}) = w_{i,j} + \eta(y_j - \hat{y}_j)x_i$$

Limitaciones de un perceptrón.



Intro Deep Learning

Multi-Layer Perceptron - Backprop

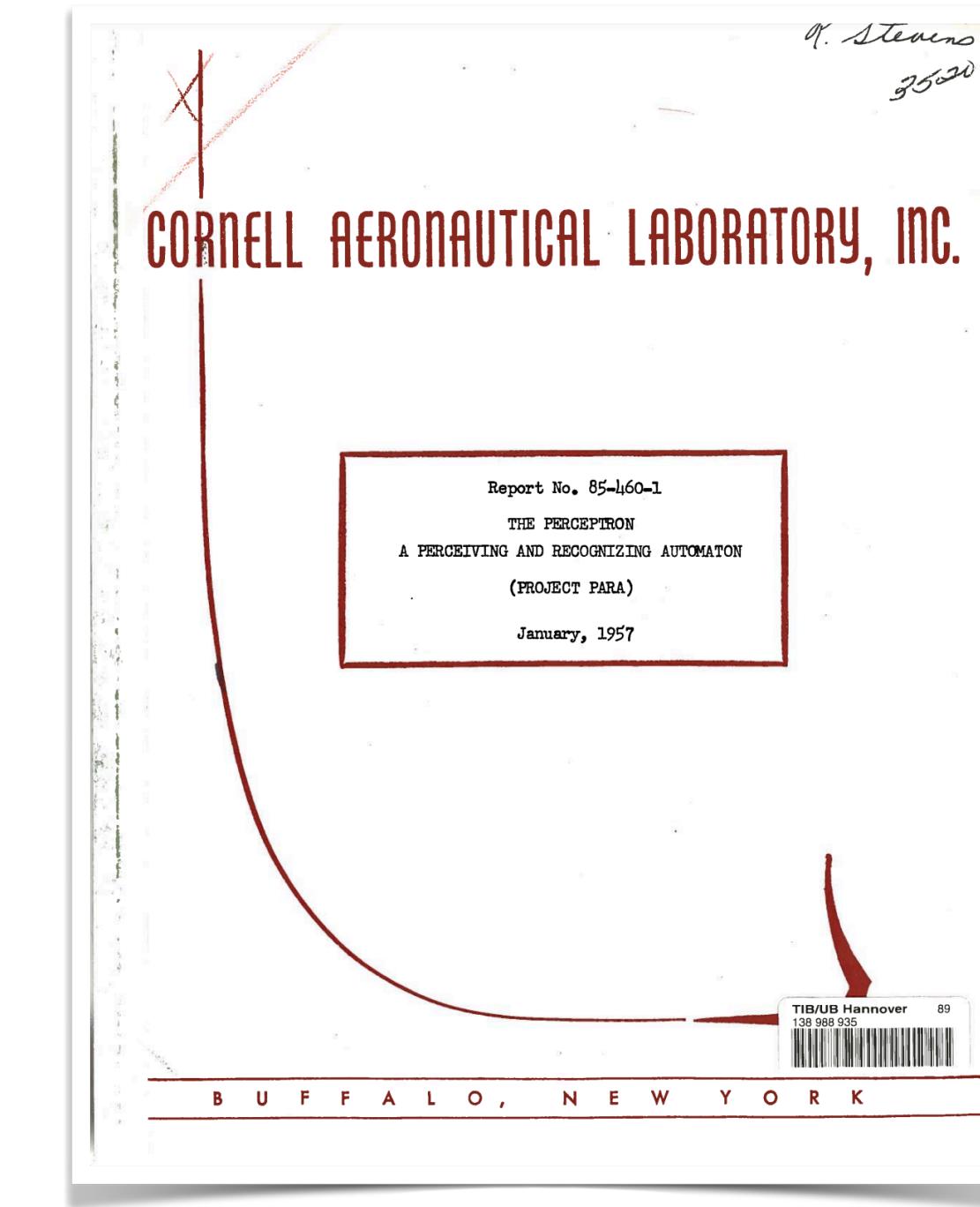
**Perceptrón
Multicapa (MLP)**

Las limitaciones de los perceptrones se pueden eliminar apilando varios perceptrones



Invierno de la I.A.

1957, Frank Rosenblatt



*The Perceptron:
A Perceiving and
recognizing automation.*

Intro Deep Learning

Multi-Layer Perceptron - Backprop

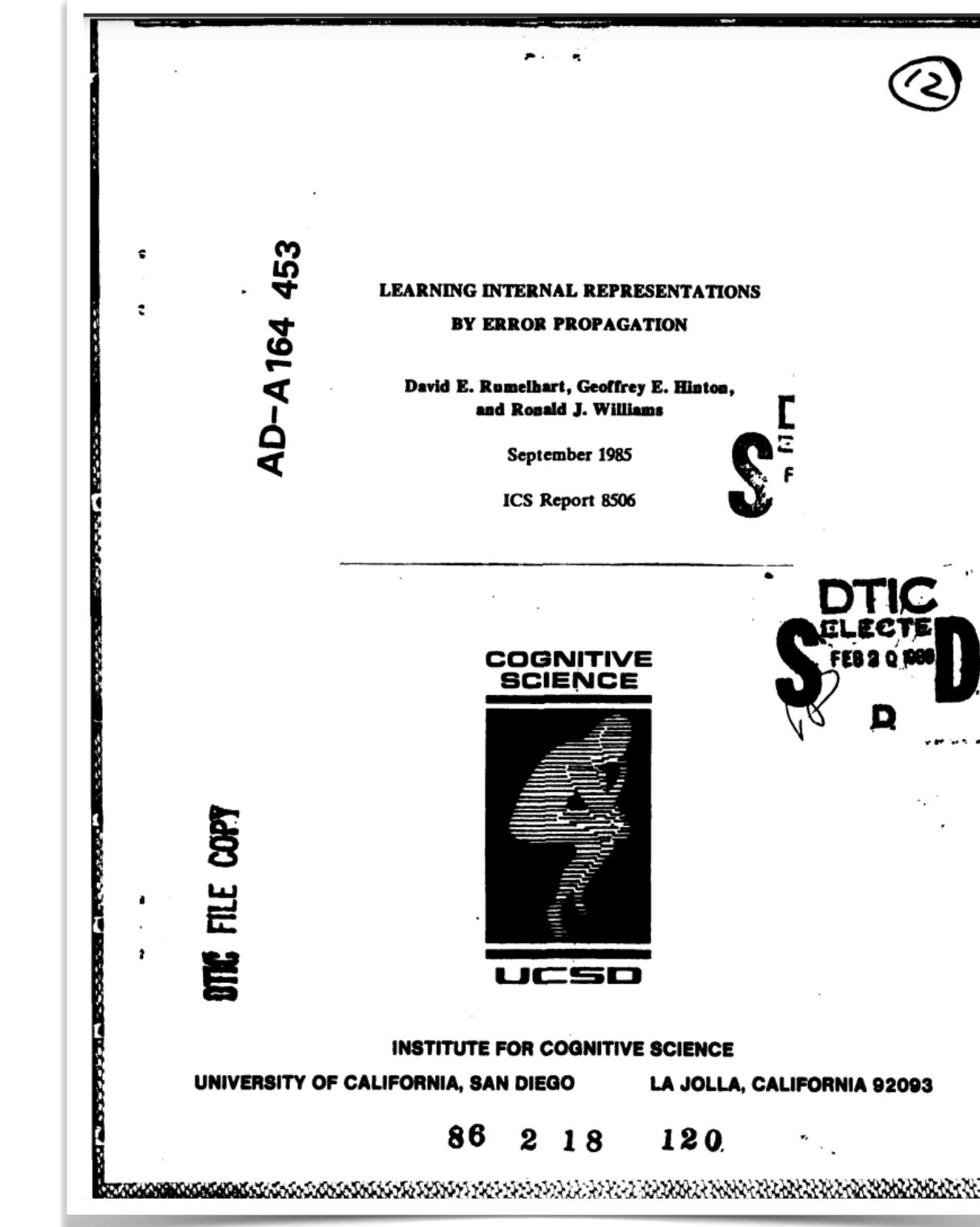
Perceptrón Multicapa (MLP)

Las limitaciones de los perceptrones se pueden eliminar apilando varios perceptrones



Invierno de la I.A.

1985, David Rumelhart, Geoffrey Hinton y Ronald Williams



*Learning Internal
Representation by
error propagation*

Intro Deep Learning

Multi-Layer Perceptron - Backprop

Perceptrón Multicapa (MLP)

Las limitaciones de los perceptrones se pueden eliminar apilando varios perceptrones



Invierno de la I.A.

1986, David Rumelhart, Geoffrey Hinton y Ronald Williams



*Learning
Representations by
back-propagation errors*

Intro Deep Learning

Multi-Layer Perceptron - Backprop

Perceptrón Multicapa (MLP)

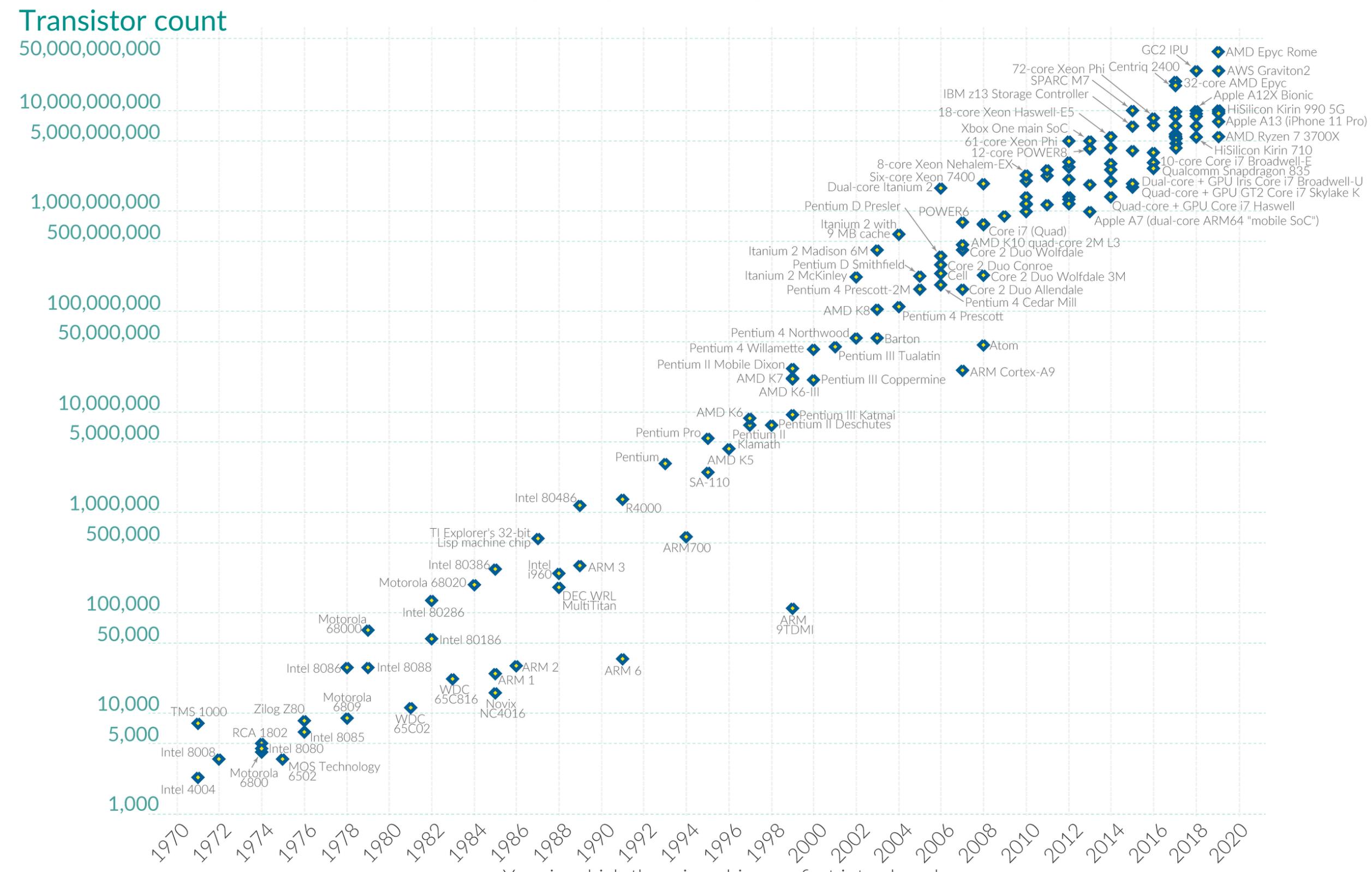
Las limitaciones de los perceptrones se pueden eliminar apilando varios perceptrones



Invierno de la I.A.

Moore's Law: The number of transistors on microchips doubles every two years
 Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important for other aspects of technological progress in computing – such as processing speed or the price of computers.

Our World
in Data



OurWorldInData.org – Research and data to make progress against the world's largest problems.

Licensed under CC-BY by the authors Hannah Ritchie and Max Roser.

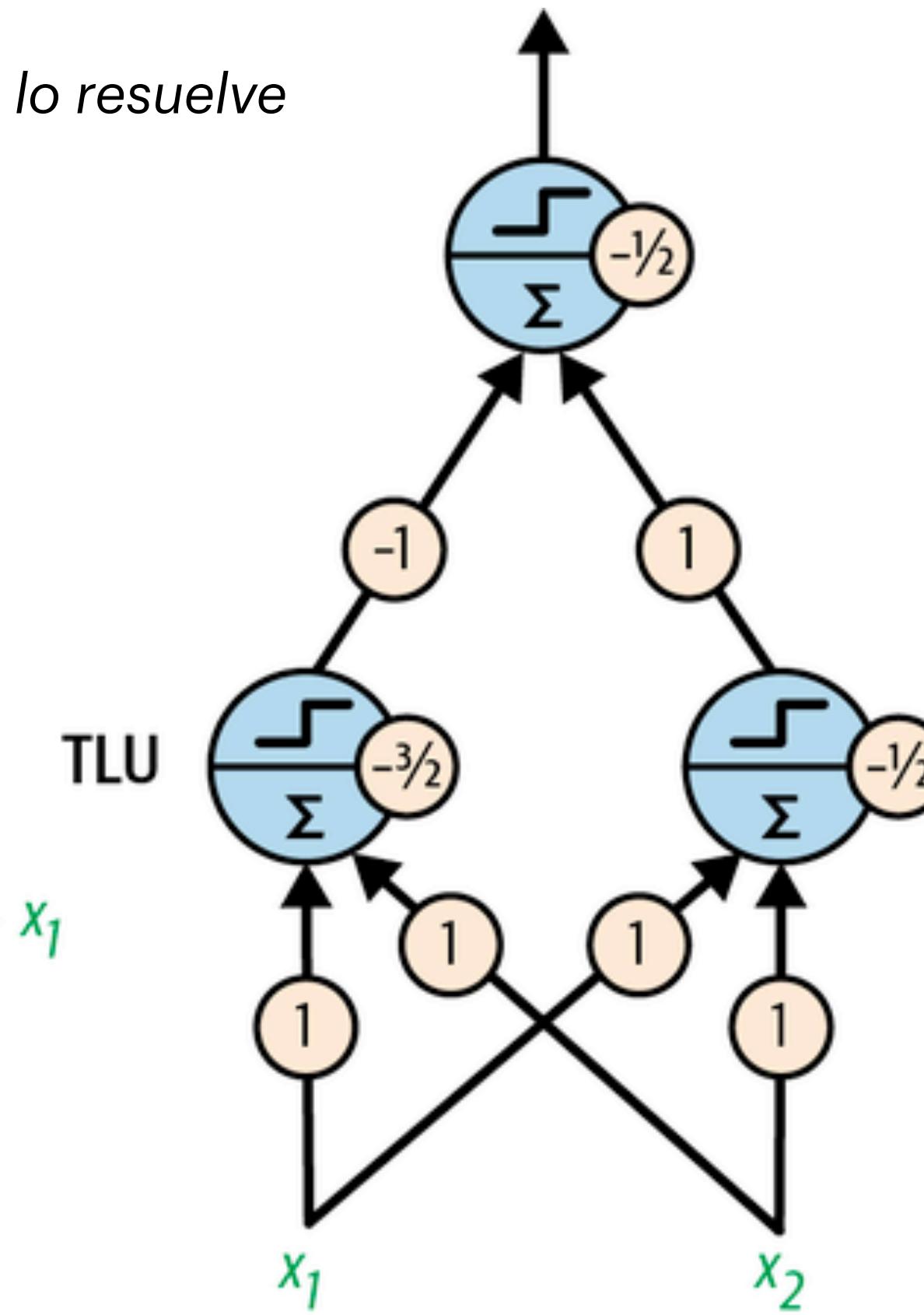
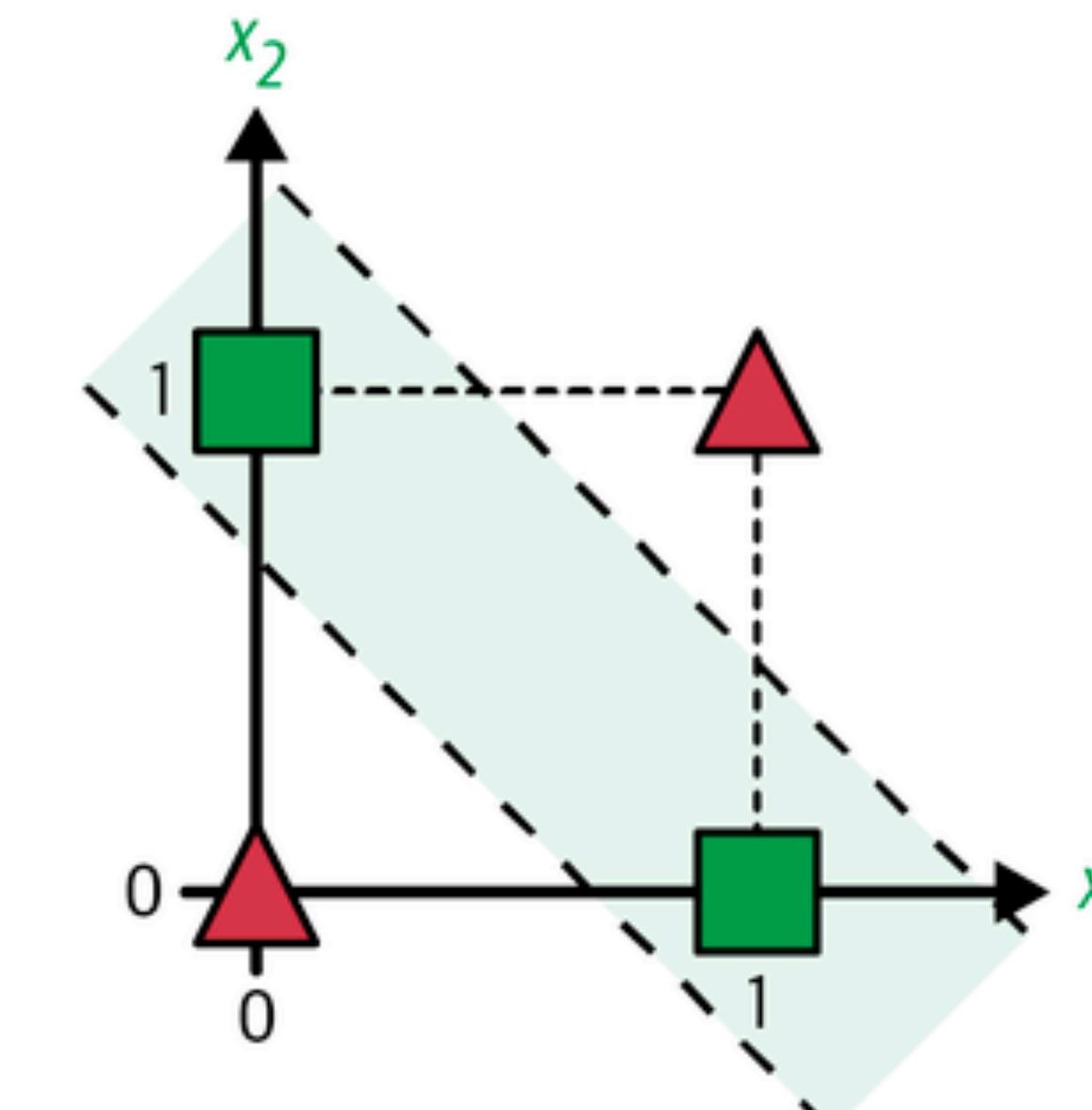
Intro Deep Learning

Multi-Layer Perceptron

Perceptrón Multicapa (MLP)

Las limitaciones de los perceptrones se pueden eliminar apilando varios perceptrones

Problema de clasificación XOR y un MLP que lo resuelve



Perceptrons de 1969 , Marvin Minsky y Seymour Papert

Intro Deep Learning

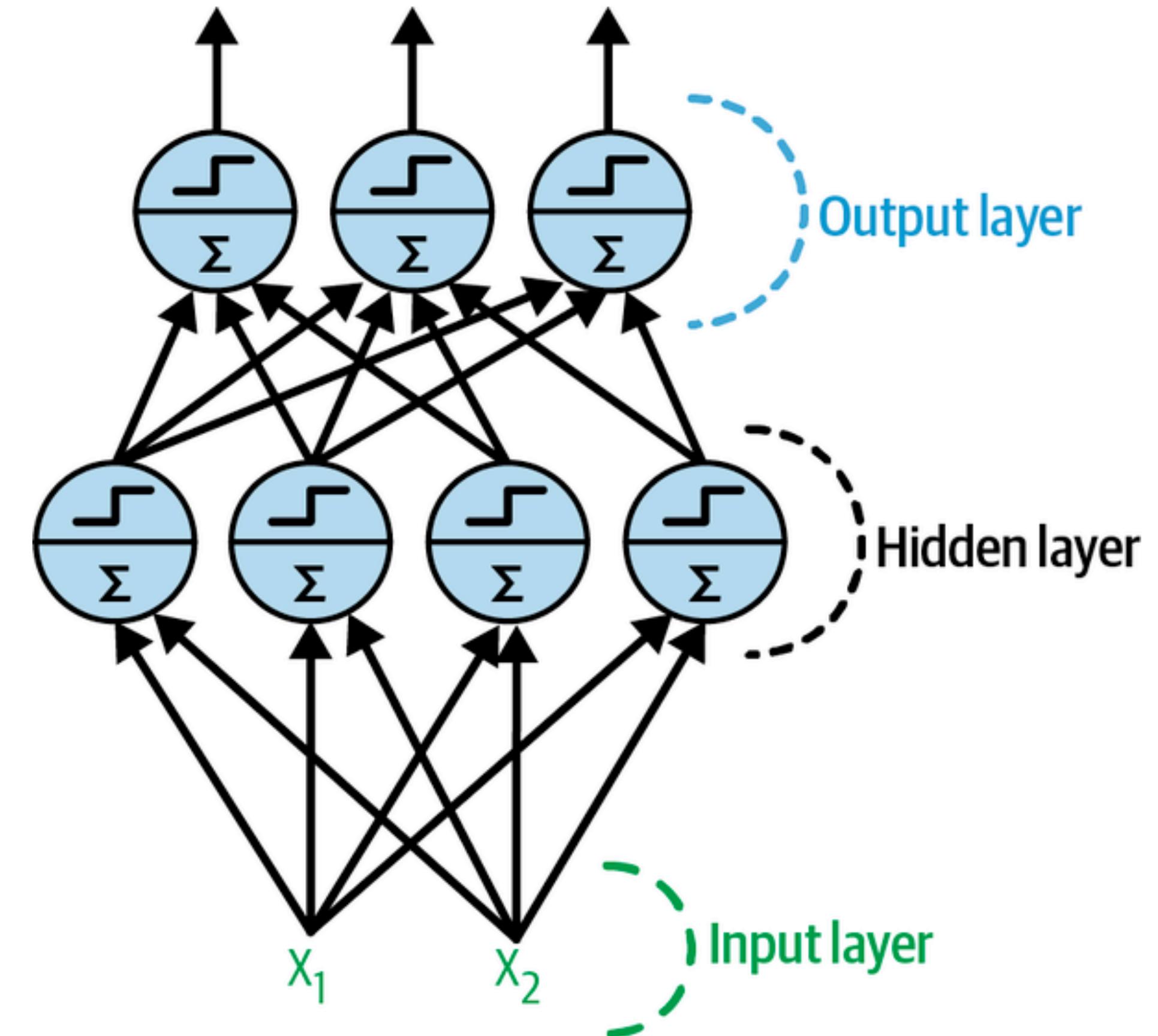
Multi-Layer Perceptron

Perceptrón Multicapa (MLP)

Las limitaciones de los perceptrones se pueden eliminar apilando varios perceptrones

MLP se compone de:

- Una capa de entrada
- Una o más capas ocultas
- Una capa de salida



Perceptrons de 1969 , Marvin Minsky y Seymour Papert

Intro Deep Learning

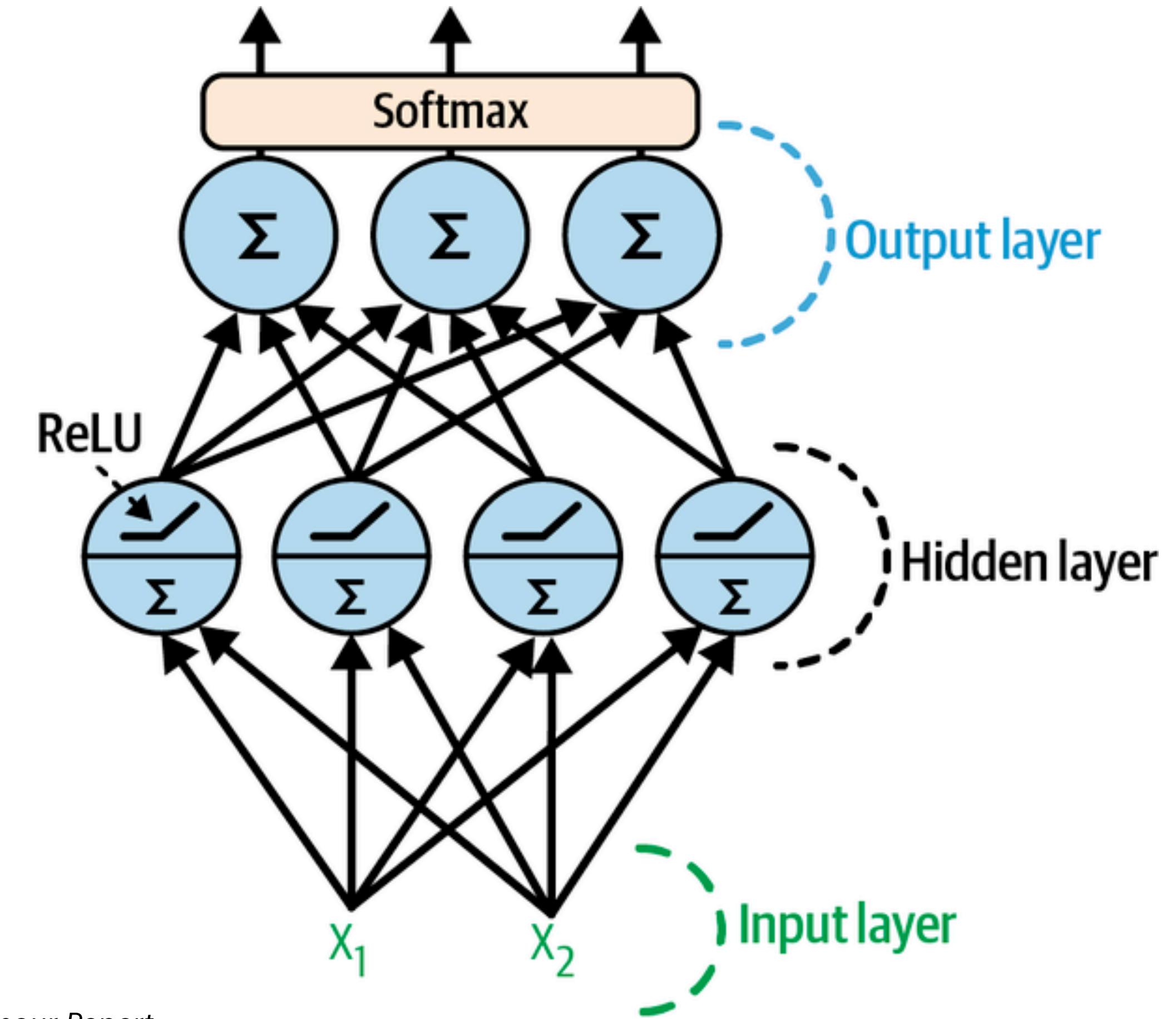
Multi-Layer Perceptron

Perceptrón Multicapa (MLP)

Las limitaciones de los perceptrones se pueden eliminar apilando varios perceptrones

MLP clasificación:

Un MLP moderno (incluidos ReLU y softmax) para clasificación



Perceptrons de 1969 , Marvin Minsky y Seymour Papert

Intro Deep Learning

Multi-Layer Perceptron

Perceptrón Multicapa (MLP)

Las limitaciones de los perceptrones se pueden eliminar apilando varios perceptrones

¿Por qué usamos funciones de activación?

¿Qué pasa si sumamos varias neuronas sin una función de activación?

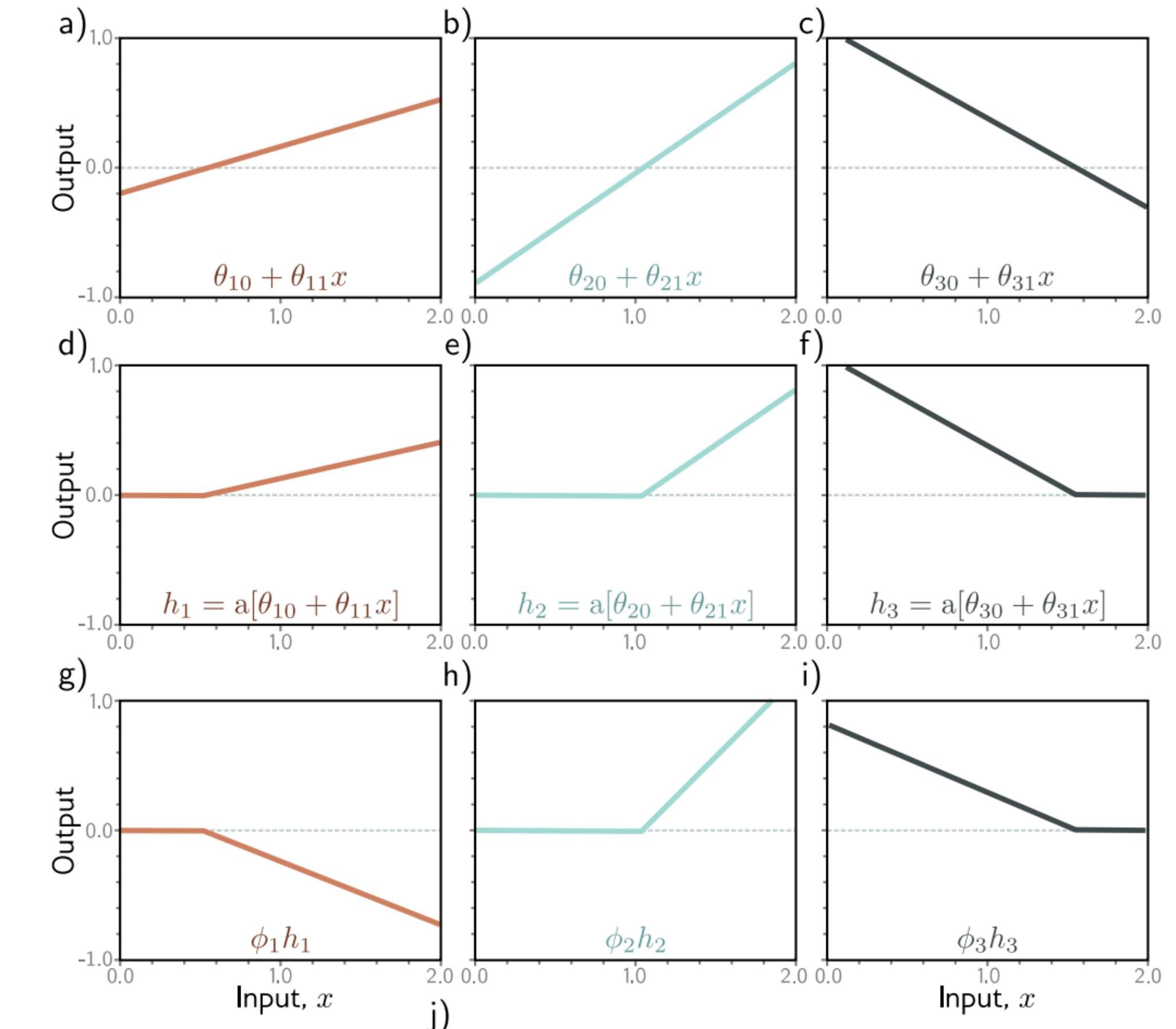
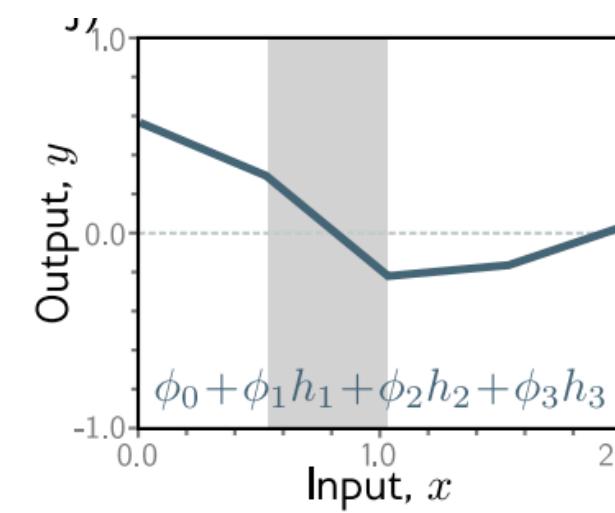
¿Por qué nuestra salida debe ser una salida no lineal?

Intro Deep Learning

Multi-Layer Perceptron

Perceptrón Multicapa (MLP)

Las limitaciones de los perceptrones se pueden eliminar apilando varios perceptrones

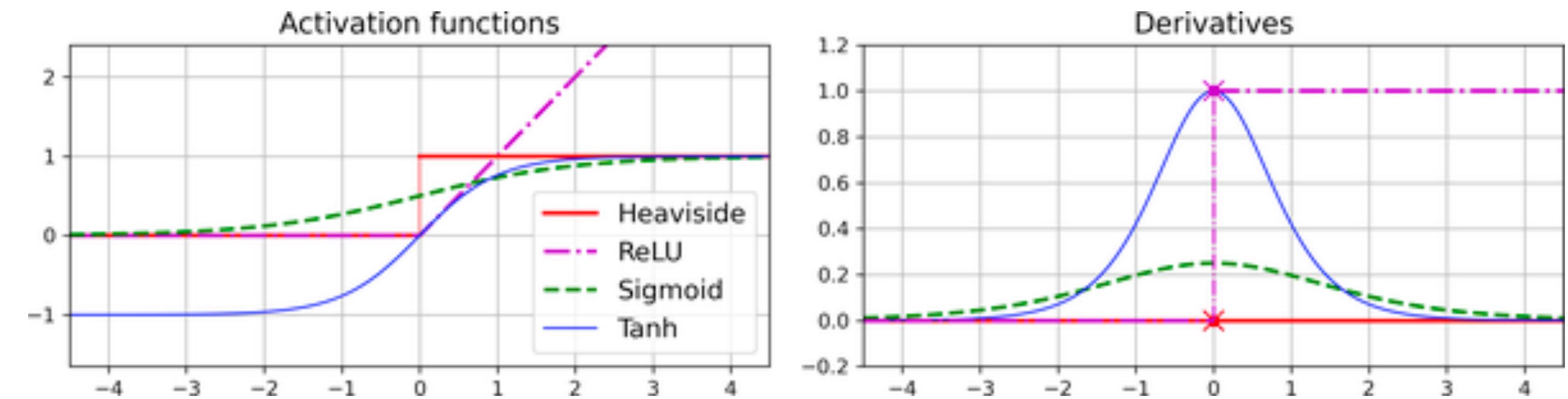


Intro Deep Learning

Multi-Layer Perceptron - Backprop

Perceptrón Multicapa (MLP)

Las limitaciones de los perceptrones se pueden eliminar apilando varios perceptrones



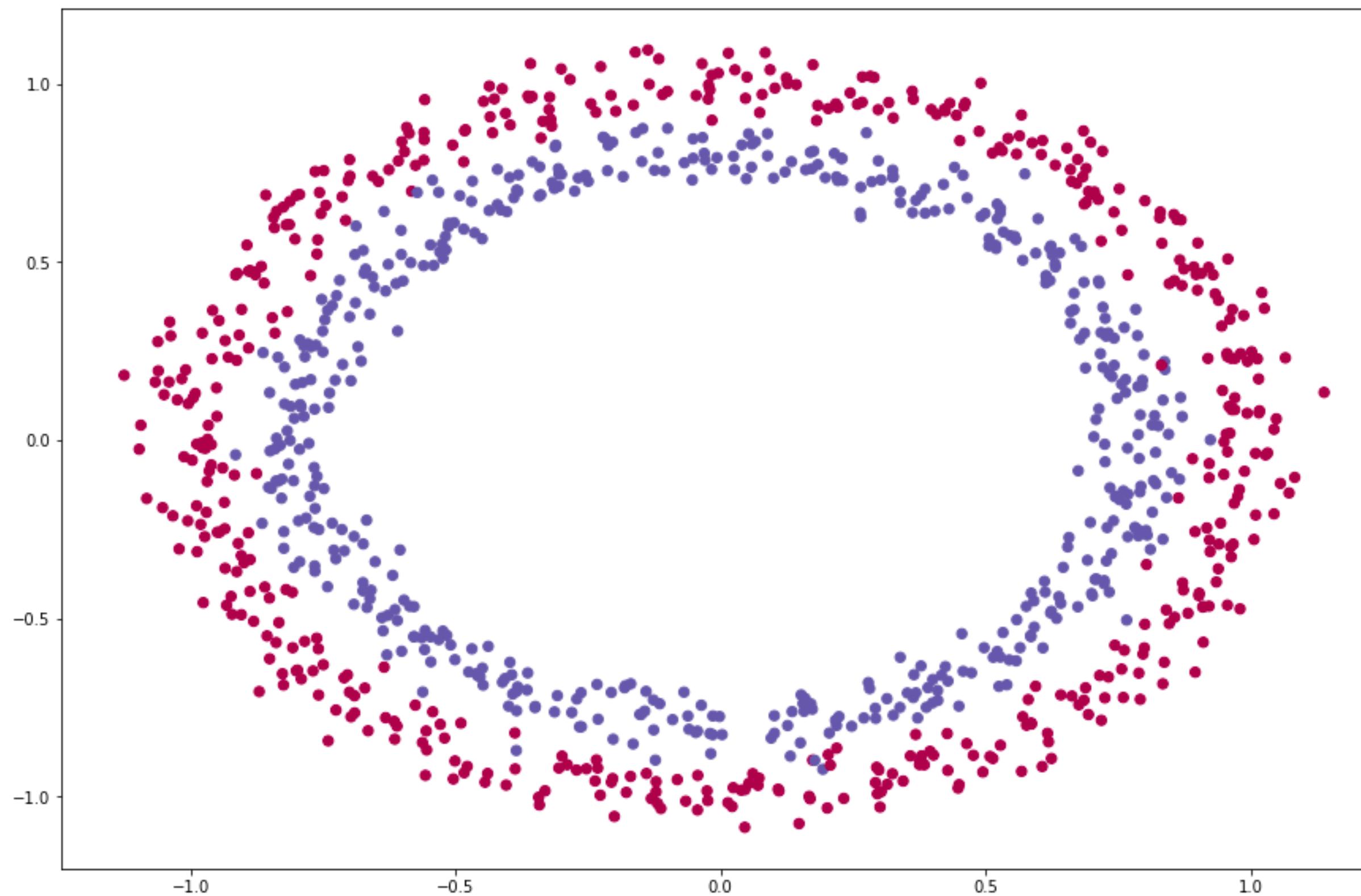
1985, David Rumelhart, Geoffrey Hinton y Ronald Williams

Intro Deep Learning

Multi-Layer Perceptron - Backprop

Perceptrón Multicapa (MLP)

Las limitaciones de los perceptrones se pueden eliminar apilando varios perceptrones



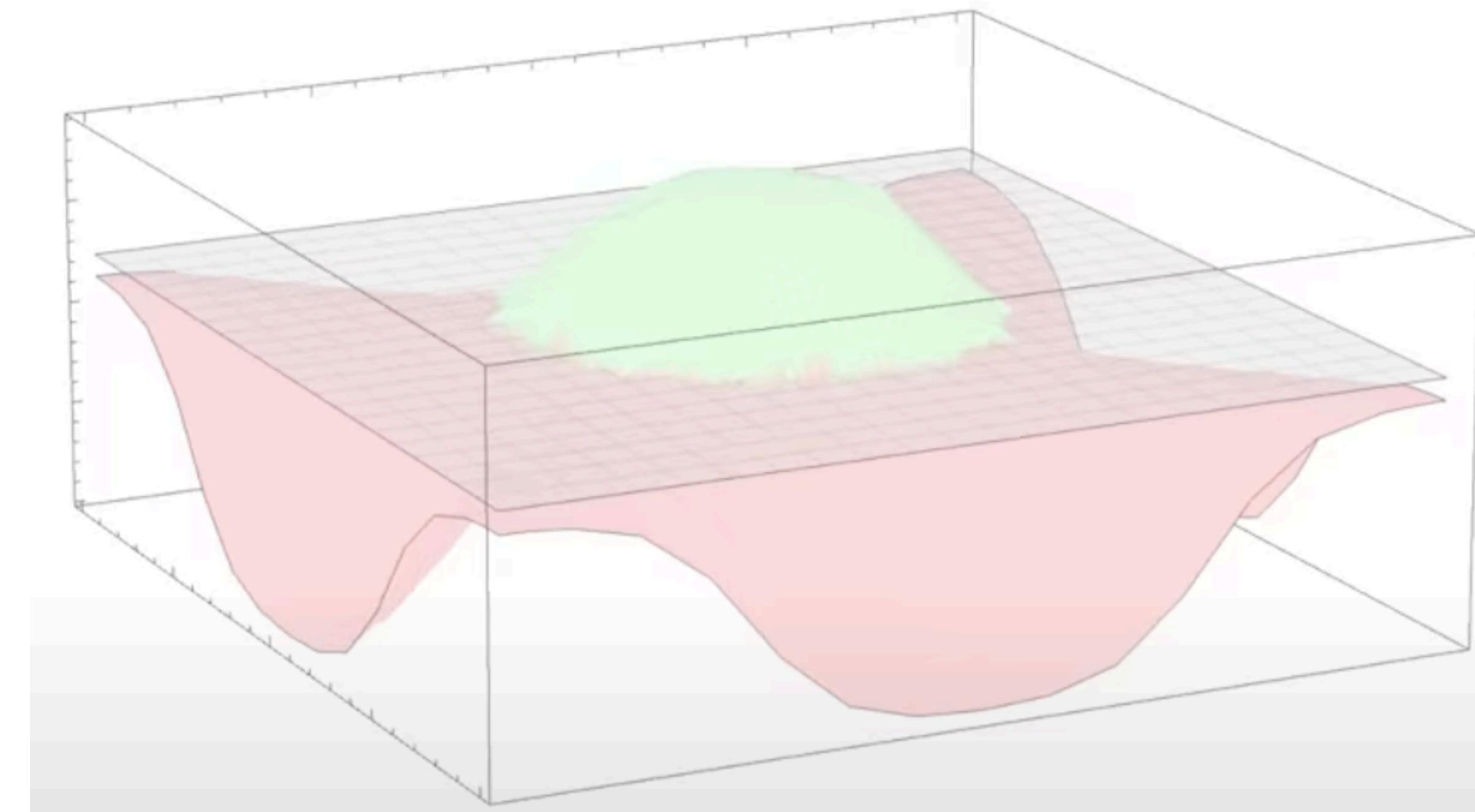
Intro Deep Learning

Multi-Layer Perceptron - Backprop

Perceptrón Multicapa (MLP)

Las limitaciones de los perceptrones se pueden eliminar apilando varios perceptrones

Función geométrica de una red neuronal con 4 neuronas.

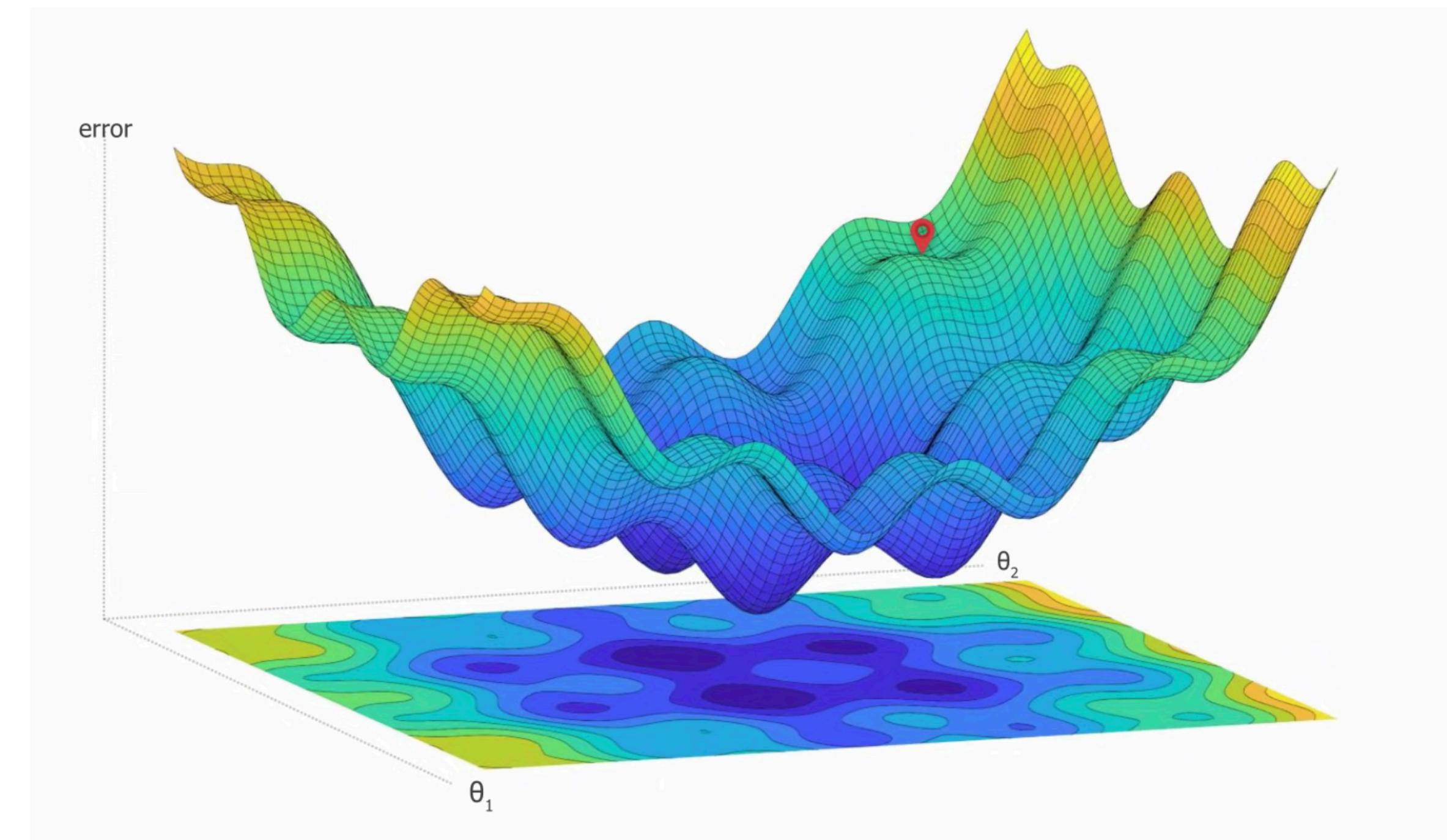


Intro Deep Learning

Backprop - Gradient Descent

Perceptrón Multicapa (MLP)

Las limitaciones de los perceptrones se pueden eliminar apilando varios perceptrones

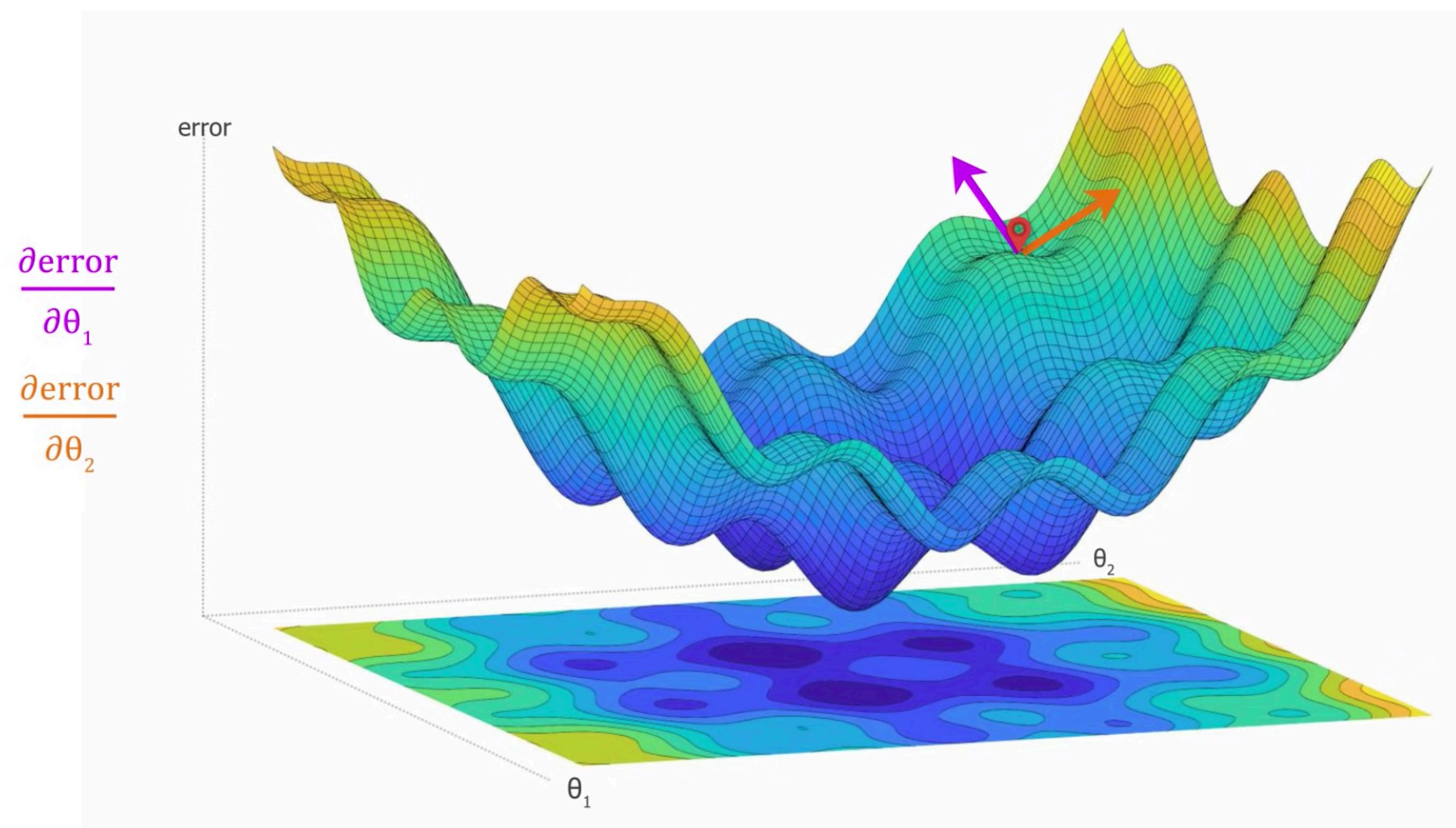


Intro Deep Learning

Backprop - Gradient Descent

Perceptrón Multicapa (MLP)

Las limitaciones de los perceptrones se pueden eliminar apilando varios perceptrones

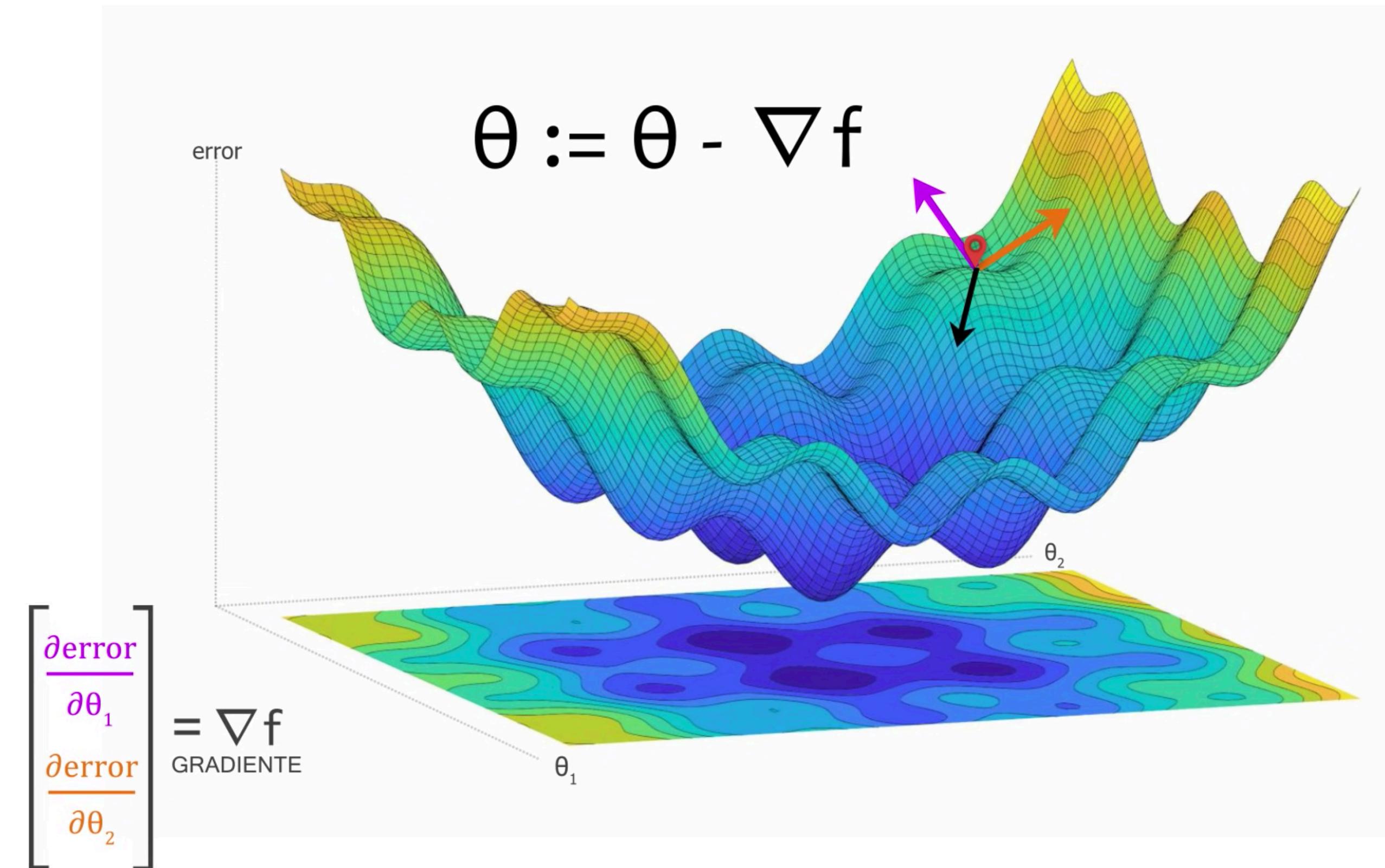


Intro Deep Learning

Backprop - Gradient Descent

Perceptrón Multicapa (MLP)

Las limitaciones de los perceptrones se pueden eliminar apilando varios perceptrones

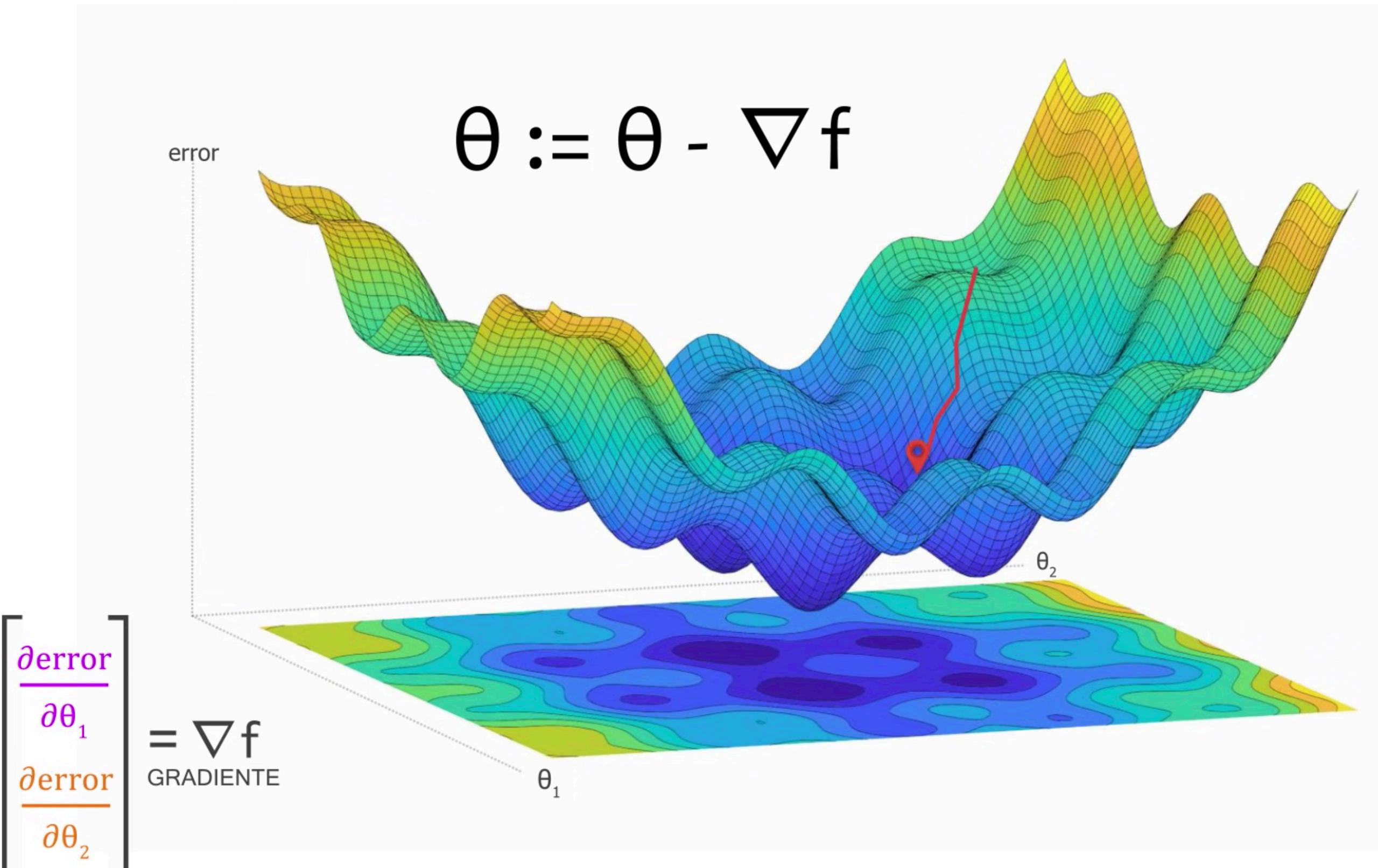


Intro Deep Learning

Backprop - Gradient Descent

Perceptrón Multicapa (MLP)

Las limitaciones de los perceptrones se pueden eliminar apilando varios perceptrones

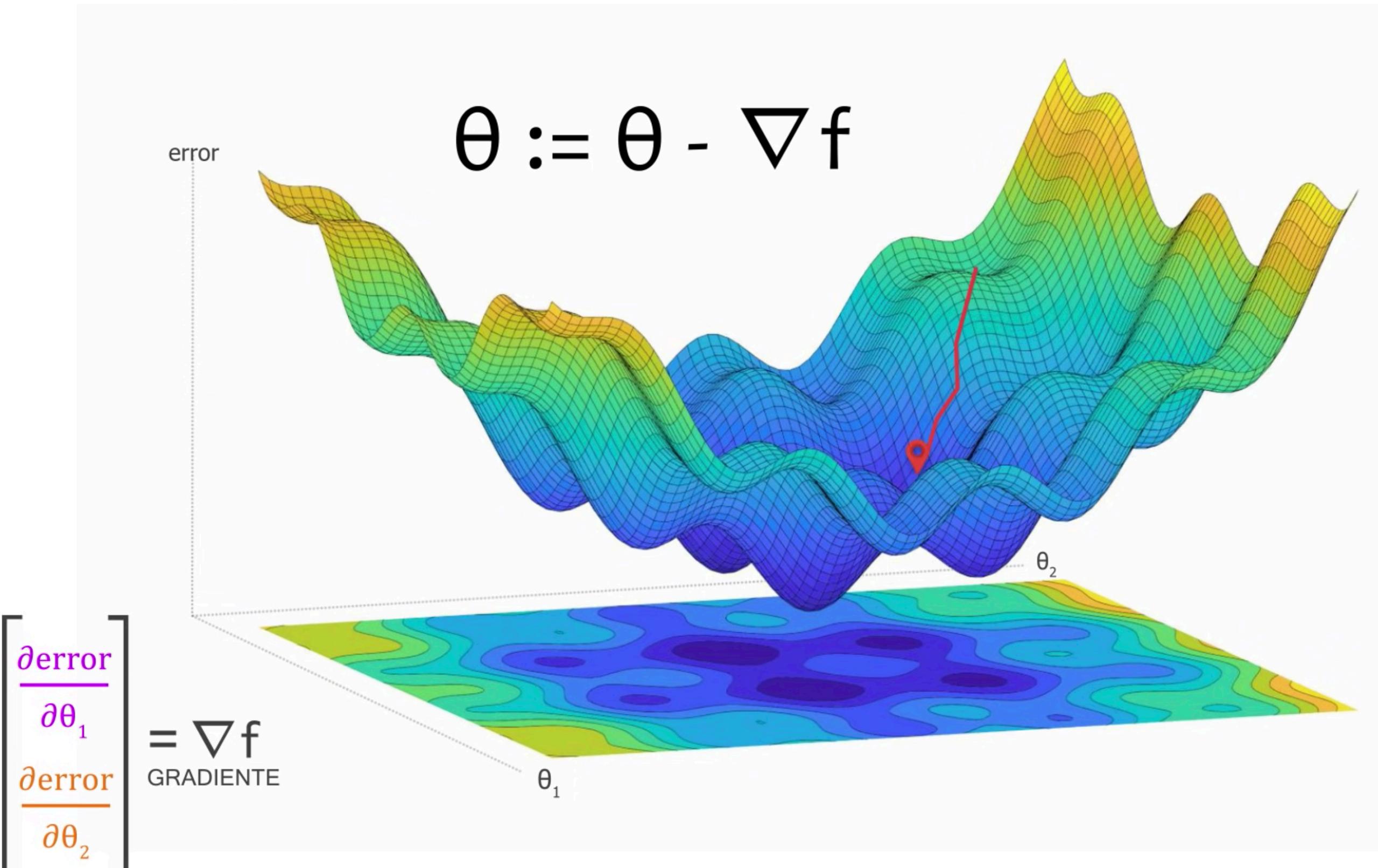


Intro Deep Learning

Backprop - Gradient Descent

Perceptrón Multicapa (MLP)

Las limitaciones de los perceptrones se pueden eliminar apilando varios perceptrones



Intro Deep Learning

Backprop

Perceptrón Multicapa (MLP)

Las limitaciones de los perceptrones se pueden eliminar apilando varios perceptrones

Derivadas de la última capa:

En L capas

$$\frac{\partial C}{\partial W^L}$$



Coste con los pesos W

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



Coste con los sesgos b

Intro Deep Learning

Backprop

Perceptrón Multicapa (MLP)

Las limitaciones de los perceptrones se pueden eliminar apilando varios perceptrones

Derivadas de la última capa:

$$Z^L = W^L X + b^L \quad \text{Suma ponderada}$$

$$a(Z^L) \quad \text{Función de activación}$$

$$C(a(Z^L)) = Error \quad \text{Función de coste}$$

Intro Deep Learning

Backprop

Perceptrón Multicapa (MLP)

Las limitaciones de los perceptrones se pueden eliminar apilando varios perceptrones

Derivadas de la última capa:

$$C(a(Z^L)) = \text{Error} \quad \text{Función de coste}$$



Composición de funciones

Derivada de composición de funciones se aplica con regla de la cadena o 'Chain Rule'

Intro Deep Learning

Backprop

Perceptrón Multicapa (MLP)

Las limitaciones de los perceptrones se pueden eliminar apilando varios perceptrones

Derivadas de la última capa:

Derivada de composición de funciones se aplica con regla de la cadena o 'Chain Rule'

$$\frac{\partial \text{Liebre}}{\partial \text{Perro}} = 3 \quad \frac{\partial \text{Perro}}{\partial \text{Tortuga}} = 5 \quad \frac{\partial \text{Liebre}}{\partial \text{Tortuga}} = ??$$

1 minuto para resolver...

Intro Deep Learning

Backprop

Perceptrón Multicapa (MLP)

Las limitaciones de los perceptrones se pueden eliminar apilando varios perceptrones

Derivadas de la última capa:

Derivada de composición de funciones se aplica con regla de la cadena o 'Chain Rule'

$$\frac{\partial \text{Liebre}}{\partial \text{Perro}} = 3$$

$$\frac{\partial \text{Perro}}{\partial \text{Tortuga}} = 5$$

$$\frac{\partial \text{Liebre}}{\partial \text{Tortuga}} = 15$$

Se deben multiplicar las derivadas intermedias

Intro Deep Learning

Backprop

Perceptrón Multicapa (MLP)

Las limitaciones de los perceptrones se pueden eliminar apilando varios perceptrones

Derivadas de la última capa:

Derivada de composición de funciones se aplica con regla de la cadena o 'Chain Rule'

Se deben multiplicar las derivadas intermedias

$$\frac{\partial \text{Liebre}}{\partial \text{Perro}} \rightarrow \frac{\partial \text{Perro}}{\partial \text{Tortuga}} = \frac{\partial \text{Liebre}}{\partial \text{Tortuga}}$$

Tip: encadenar numerador con denominador...

Intro Deep Learning

Backprop

Perceptrón Multicapa (MLP)

Las limitaciones de los perceptrones se pueden eliminar apilando varios perceptrones

Derivadas de la última capa:

Derivada de composición de funciones se aplica con regla de la cadena o 'Chain Rule'

$$\frac{\partial \text{Liebre}}{\partial \text{Perro}} \quad \frac{\partial \text{Perro}}{\partial \text{Tortuga}} = \frac{\partial \text{Liebre}}{\partial \text{Tortuga}}$$

Intro Deep Learning

Backprop

Perceptrón Multicapa (MLP)

Las limitaciones de los perceptrones se pueden eliminar apilando varios perceptrones

Derivadas de la última capa:

$$\frac{\partial C}{\partial W^L}$$

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

Derivada de composición de funciones se aplica con regla de la cadena o 'Chain Rule'

CHAIN RULE!

$$Z^L = W^L a^{L-1} + b^L$$

$$C(a^L(Z^L))$$

Intro Deep Learning

Backprop

Perceptrón Multicapa (MLP)

Las limitaciones de los perceptrones se pueden eliminar apilando varios perceptrones

Derivadas de la última capa:

Derivada de composición de funciones se aplica con regla de la cadena o 'Chain Rule'

$$\frac{\partial C}{\partial W^L} = \frac{\partial C}{\partial a^L} \cdot \frac{\partial a^L}{\partial z^L} \cdot \frac{\partial z^L}{\partial W^L}$$

$$\frac{\partial C}{\partial b^L} = \frac{\partial C}{\partial a^L} \cdot \frac{\partial a^L}{\partial z^L} \cdot \frac{\partial z^L}{\partial b^L}$$

$$Z^L = W^L a^{L-1} + b^L \quad C(a^L(Z^L))$$

Intro Deep Learning

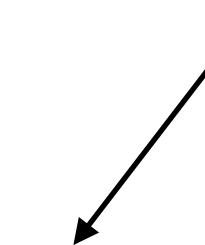
Backprop

Perceptrón Multicapa (MLP)

Las limitaciones de los perceptrones se pueden eliminar apilando varios perceptrones

Derivadas de la última capa:

$$\frac{\partial C}{\partial W^L} = \frac{\partial C}{\partial a^L} \cdot \frac{\partial a^L}{\partial z^L} \cdot \frac{\partial z^L}{\partial W^L}$$



$$\frac{\partial C}{\partial a_j^L} = (a_j^L - y_j)$$

Derivada de la función de coste

¿Que tanto varía el costo con respecto al output de la última capa?

$$C(a_j^L) = \frac{1}{2} \sum_j (y_j - a_j^L)^2$$

Error cuadrático medio

Intro Deep Learning

Backprop

Perceptrón Multicapa (MLP)

Las limitaciones de los perceptrones se pueden eliminar apilando varios perceptrones

Derivadas de la última capa:

$$\frac{\partial C}{\partial W^L} = \frac{\partial C}{\partial a^L} \cdot \frac{\partial a^L}{\partial z^L} \cdot \frac{\partial z^L}{\partial W^L}$$

$$\frac{\partial a^L}{\partial z^L} = a^L(z^L) \cdot (1 - a^L(z^L))$$

Derivada de la función de activación

¿Que tanto varía el output de la neurona cuando variamos la suma ponderada de la neurona?

$$a^L(z^L) = \frac{1}{1 + e^{-z^L}}$$

Función sigmoide

Intro Deep Learning

Backprop

Perceptrón Multicapa (MLP)

Las limitaciones de los perceptrones se pueden eliminar apilando varios perceptrones

Derivadas de la última capa:

$$\frac{\partial C}{\partial W^L} = \frac{\partial C}{\partial a^L} \cdot \frac{\partial a^L}{\partial z^L} \cdot \frac{\partial z^L}{\partial W^L}$$
$$\frac{\partial z^L}{\partial W^L} = a^{L-1} \quad \frac{\partial z^L}{\partial b^L} = 1$$

Derivando la suma ponderada

¿Que tanto varía la suma ponderada de la neurona con los parámetros?

$$z^L = \sum_i a_i^{L-1} \cdot w_i^L + b^L$$

Suma ponderada

Intro Deep Learning

Backprop

Perceptrón Multicapa (MLP)

Las limitaciones de los perceptrones se pueden eliminar apilando varios perceptrones

Derivadas de la última capa:

$$\frac{\partial C}{\partial W^L} = \frac{\partial C}{\partial a^L} \cdot \frac{\partial a^L}{\partial z^L} \cdot \frac{\partial z^L}{\partial W^L}$$

$$\frac{\partial C}{\partial b^L} = \frac{\partial C}{\partial a^L} \cdot \frac{\partial a^L}{\partial z^L} \cdot \frac{\partial z^L}{\partial b^L}$$

Resumiendo

$$\frac{\partial C}{\partial a_j^L} = (a_j^L - y_j)$$

$$\frac{\partial a^L}{\partial z^L} = a^L(Z^L) \cdot (1 - a^L(Z^L))$$

$$\frac{\partial z^L}{\partial W^L} = a_i^{L-1} \quad \frac{\partial z^L}{\partial b^L} = 1$$

Intro Deep Learning

Backprop

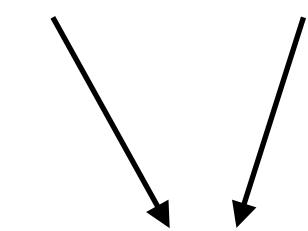
Perceptrón Multicapa (MLP)

Las limitaciones de los perceptrones se pueden eliminar apilando varios perceptrones

Error imputado
a la neurona

$$\delta^L$$

Derivadas de la última capa:

$$\frac{\partial C}{\partial W^L} = \frac{\partial C}{\partial a^L} \cdot \frac{\partial a^L}{\partial z^L} \cdot \frac{\partial z^L}{\partial W^L}$$

$$\frac{\partial C}{\partial Z^L}$$

Responsabilidad de la
neurona en el resultado

Resumiendo

Error a un pequeño cambio en la
suma de la neurona.

Derivada grande

Afecta
considerablemente
el coste

Derivada pequeña

Da igual como se
varía la suma, no
afectará el coste

Intro Deep Learning

Backprop

Perceptrón Multicapa (MLP)

Las limitaciones de los perceptrones se pueden eliminar apilando varios perceptrones

*Error imputado
a la neurona*

δ^L

Derivadas de la última capa:

$$\frac{\partial C}{\partial W^L} = \frac{\partial C}{\partial a^L} \cdot \frac{\partial a^L}{\partial z^L} \cdot \frac{\partial z^L}{\partial W^L} \longrightarrow \frac{\partial C}{\partial W^L} = \delta^L \cdot \frac{\partial z^L}{\partial W^L}$$

$$\frac{\partial C}{\partial b^L} = \frac{\partial C}{\partial a^L} \cdot \frac{\partial a^L}{\partial z^L} \cdot \frac{\partial z^L}{\partial b^L} \longrightarrow \frac{\partial C}{\partial b^L} = \delta^L \cdot \frac{\partial z^L}{\partial b^L}$$

Resumiendo

Intro Deep Learning

Backprop

Perceptrón Multicapa (MLP)

Las limitaciones de los perceptrones se pueden eliminar apilando varios perceptrones

*Error imputado
a la neurona*

$$\delta^L$$

Derivadas de la última capa:

$$\frac{\partial C}{\partial W^L} = \delta^L \cdot \frac{\partial Z^L}{\partial W^L} \longrightarrow$$

$$\frac{\partial C}{\partial b^L} = \delta^L \cdot \frac{\partial Z^L}{\partial b^L} \longrightarrow$$

Derivando

$$\frac{\partial C}{\partial W^L} = \delta^L \cdot a_i^{L-1}$$

$$\frac{\partial C}{\partial b^L} = \delta^L \cdot 1$$

Intro Deep Learning

Backprop

Perceptrón Multicapa (MLP)

Las limitaciones de los perceptrones se pueden eliminar apilando varios perceptrones

Error imputado
a la neurona

$$\delta^L = \frac{\partial C}{\partial a^L} \cdot \frac{\partial a^L}{\partial Z^L}$$

Derivadas de la última capa:

$$\frac{\partial C}{\partial W^L} = \delta^L \cdot a_i^{L-1} \longrightarrow$$

Derivando

Error de la neurona por la función de activación de la capa anterior.

$$\frac{\partial C}{\partial b^L} = \delta^L \cdot 1 \longrightarrow$$

Derivada del coste con respecto al bias es igual al error de la neurona

Intro Deep Learning

Backprop

Perceptrón Multicapa (MLP)

Las limitaciones de los perceptrones se pueden eliminar apilando varios perceptrones

¿Cómo derivamos la capa anterior?

$$\frac{\partial C}{\partial W^{L-1}}$$

CHAIN RULE!

$$\frac{\partial C}{\partial b^{L-1}}$$

$$C(a^L(W^L \cdot a^{L-1}(W^{L-1} \cdot a^{L-2} + b^{L-1}) + b^L))$$

Intro Deep Learning

Backprop

Perceptrón Multicapa (MLP)

Las limitaciones de los perceptrones se pueden eliminar apilando varios perceptrones

¿Cómo derivamos la capa anterior?

$$\frac{\partial C}{\partial W^{L-1}} = \frac{\partial C}{\partial a^L} \cdot \frac{\partial a^L}{\partial Z^L} \cdot \frac{\partial Z^L}{\partial a^{L-1}} \cdot \frac{a^{L-1}}{\partial Z^{L-1}} \cdot \frac{\partial Z^{L-1}}{\partial W^{L-1}}$$

$$\frac{\partial C}{\partial b^{L-1}} = \frac{\partial C}{\partial a^L} \cdot \frac{\partial a^L}{\partial Z^L} \cdot \frac{\partial Z^L}{\partial a^{L-1}} \cdot \frac{a^{L-1}}{\partial Z^{L-1}} \cdot \frac{\partial Z^{L-1}}{\partial b^{L-1}}$$

$$C(a^L(W^L \cdot a^{L-1}(W^{L-1} \cdot a^{L-2} + b^{L-1}) + b^L))$$

Intro Deep Learning

Backprop

Perceptrón Multicapa (MLP)

Las limitaciones de los perceptrones se pueden eliminar apilando varios perceptrones

¿Cómo derivamos la capa anterior?

$$\frac{\partial C}{\partial W^{L-1}} = \delta^L \cdot W^L \cdot \underbrace{\frac{\partial Z^L}{\partial a^{L-1}} \cdot \frac{a^{L-1}}{\partial Z^{L-1}} \cdot \frac{\partial Z^{L-1}}{\partial W^{L-1}}}_{\text{Derivada de la función de activación}}$$

$$\frac{\partial C}{\partial b^{L-1}} = \delta^L \cdot \underbrace{\frac{\partial Z^L}{\partial a^{L-1}} \cdot \frac{a^{L-1}}{\partial Z^{L-1}} \cdot \frac{\partial Z^{L-1}}{\partial b^{L-1}}}_{1}$$

Intro Deep Learning

Backprop

Perceptrón Multicapa (MLP)

Las limitaciones de los perceptrones se pueden eliminar apilando varios perceptrones

¿Cómo derivamos la capa anterior?

W^L

¿Cómo varia la suma ponderada de una capa cuando se varía el output de una neurona en la capa previa?

$$\frac{\partial C}{\partial W^{L-1}} = \frac{\partial C}{\partial a^L} \cdot \frac{\partial a^L}{\partial Z^L} \cdot \boxed{\frac{\partial Z^L}{\partial a^{L-1}}} \cdot \frac{a^{L-1}}{\partial Z^{L-1}} \cdot \frac{\partial Z^{L-1}}{\partial W^{L-1}}$$

$$\frac{\partial C}{\partial b^{L-1}} = \frac{\partial C}{\partial a^L} \cdot \frac{\partial a^L}{\partial Z^L} \cdot \boxed{\frac{\partial Z^L}{\partial a^{L-1}}} \cdot \frac{a^{L-1}}{\partial Z^{L-1}} \cdot \frac{\partial Z^{L-1}}{\partial b^{L-1}}$$

Intro Deep Learning

Backprop

Perceptrón Multicapa (MLP)

Las limitaciones de los perceptrones se pueden eliminar apilando varios perceptrones

Resumiendo back-propagation

1. Computo del error de la última capa

$$\delta^L = \frac{\partial C}{\partial a^L} \cdot \frac{\partial a^L}{\partial Z^L}$$

2. Retropropagamos el error a la capa anterior

$$\delta^{L-1} = W^L \cdot \delta^L \cdot \frac{\partial a^{L-1}}{\partial Z^{L-1}}$$

3. Calculamos las derivadas de la capa usando el error

$$\frac{\partial C}{\partial b^{L-1}} = \delta^{L-1} \quad \frac{\partial C}{\partial W^{L-1}} = \delta^{L-1} \cdot a^{L-2}$$

¡Backprop es derivar solo 4 expresiones!

Intro Deep Learning

Backprop

Perceptrón Multicapa (MLP)

Las limitaciones de los perceptrones se pueden eliminar apilando varios perceptrones

Resumiendo back-propagation

1. Computo del error de la última capa

$$\delta^L = \frac{\partial C}{\partial a^L} \cdot \frac{\partial a^L}{\partial Z^L}$$

2. Retropropagamos el error a la capa anterior

$$\delta^{L-1} = W^L \cdot \delta^L \cdot \frac{\partial a^{L-1}}{\partial Z^{L-1}}$$

3. Calculamos las derivadas de la capa usando el error

$$\frac{\partial C}{\partial b^{L-1}} = \delta^{L-1} \quad \frac{\partial C}{\partial W^{L-1}} = \delta^{L-1} \cdot \alpha$$

¡Backprop es derivar solo 4 expresiones!

