FUNDACIÓN YPF

Módulo VII: Introducción a Redes Neuronales y Deep Learning.

Clase 25: Intro. RN - Perceptrón simple

FUNDACIÓN YPF

¿Ponemos a grabar el taller?

FUNDACIÓN YPF

¿QUÉ VAMOS A VER HOY?

- Introducción a Redes Neuronales
- Descenso por gradiente
- Librerías de Python para RN
- Concepto Perceptrón Simple

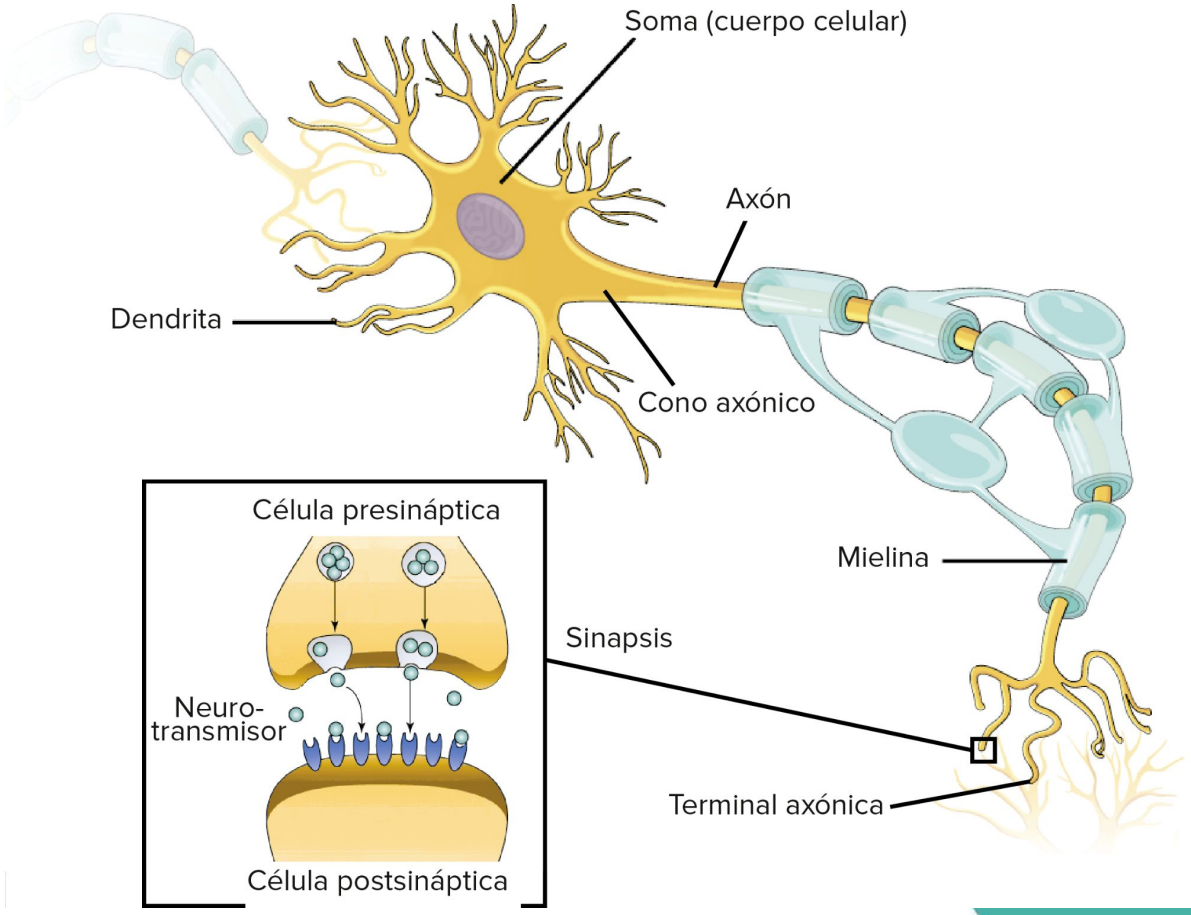
FUNDACIÓN YPF

Redes Neuronales

FUNDACIÓN YPF

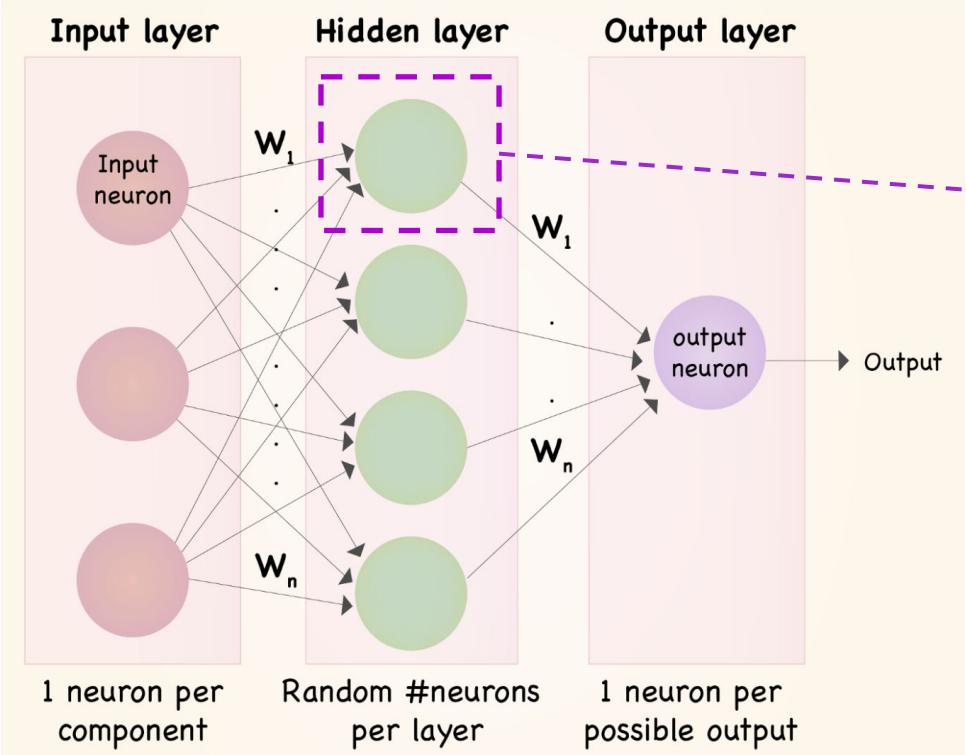
Redes Neuronales

Algoritmo cuyo unidad básica de procesamiento está inspirada en la célula fundamental del sistema nervioso humano: la **neurona**



FUNDACIÓN YPF

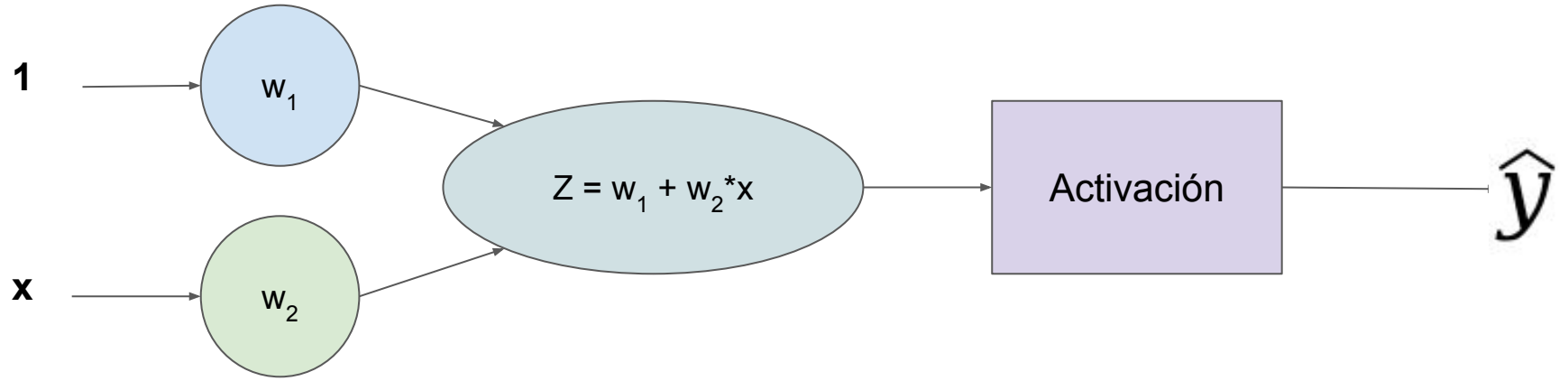
Redes Neuronales



Neurona: Unidades de procesamiento que intercambian datos o información

FUNDACIÓN YPF

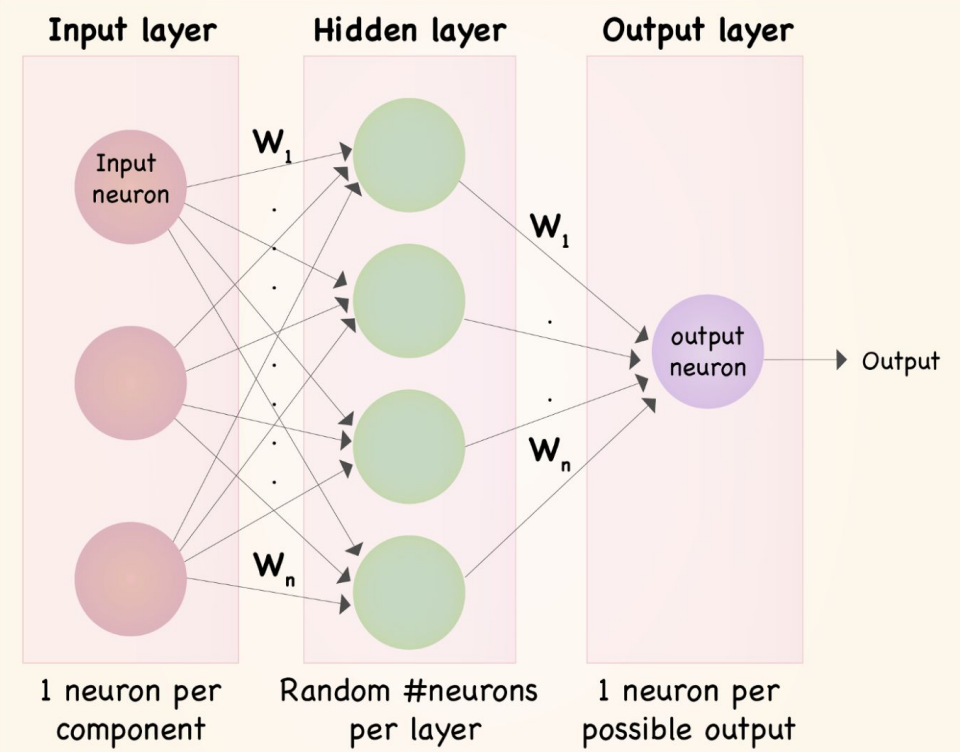
Neurona básica



Perceptrón Simple

FUNDACIÓN YPF

Redes Neuronales

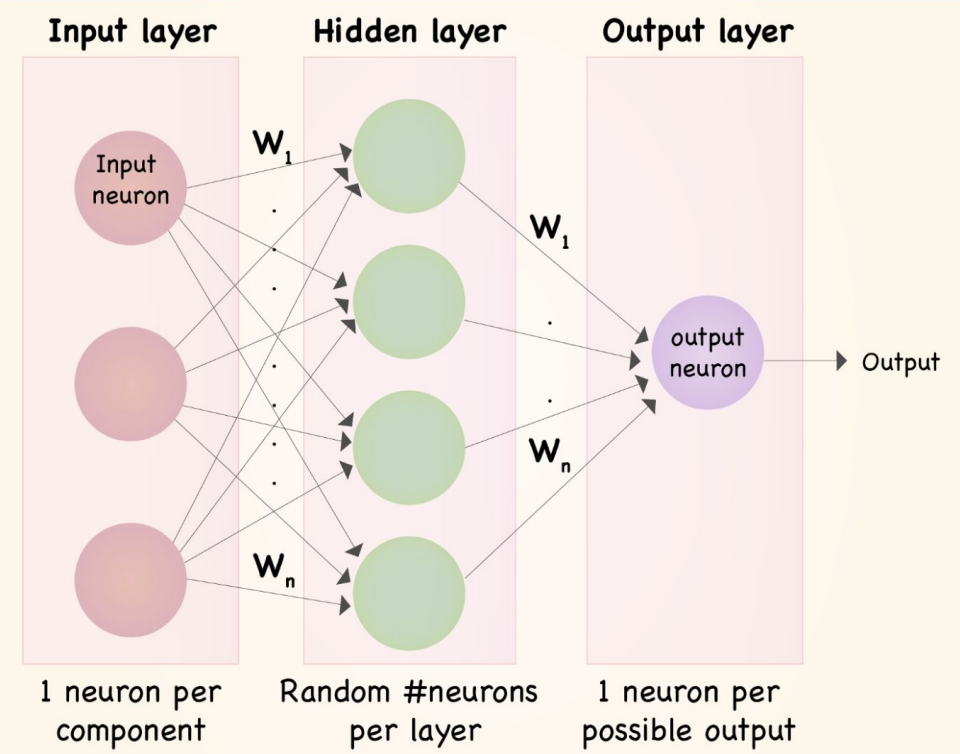


Cada neurona tiene sus propios pesos/parámetros.

Podemos encontrar miles a millones de parámetros para toda la red para aplicaciones comunes.

FUNDACIÓN YPF

Redes Neuronales



Cada neurona tiene sus propios pesos/parámetros.

Deep Learning consiste en encontrar esos pesos de manera eficiente, bajo la condición de realizar correctamente una tarea objetivo.

FUNDACIÓN YPF

Aplicaciones

Las redes neuronales son particularmente eficientes al trabajar con **datos “sin estructura”:** Imágenes/videos, Audio, Texto (NLP), juegos, y más.

FUNDACIÓN YPF

Algunas aplicaciones

Imágenes

- Clasificación de Imágenes en Facebook y Google
- DeepFakes
- Visión por computadora para vehículos autónomos

Juego

- AlphaGo
- Deep Reinforcement Learning

Texto

- Google Translate
- ChatGPT
- BERT/ELMO
- Sentiment Analysis

FUNDACIÓN YPF

Implementación en Python

FUNDACIÓN YPF

Scikit-learn

Esta librería ofrece una **limitada oferta** para desarrollar redes neuronales. Hasta hace pocos años no era posible implementar redes neuronales. Sin embargo, su **sintaxis** es de las **más sencillas** si queremos implementar modelos de clasificación o regresión simples utilizando deep learning.

FUNDACIÓN YPF

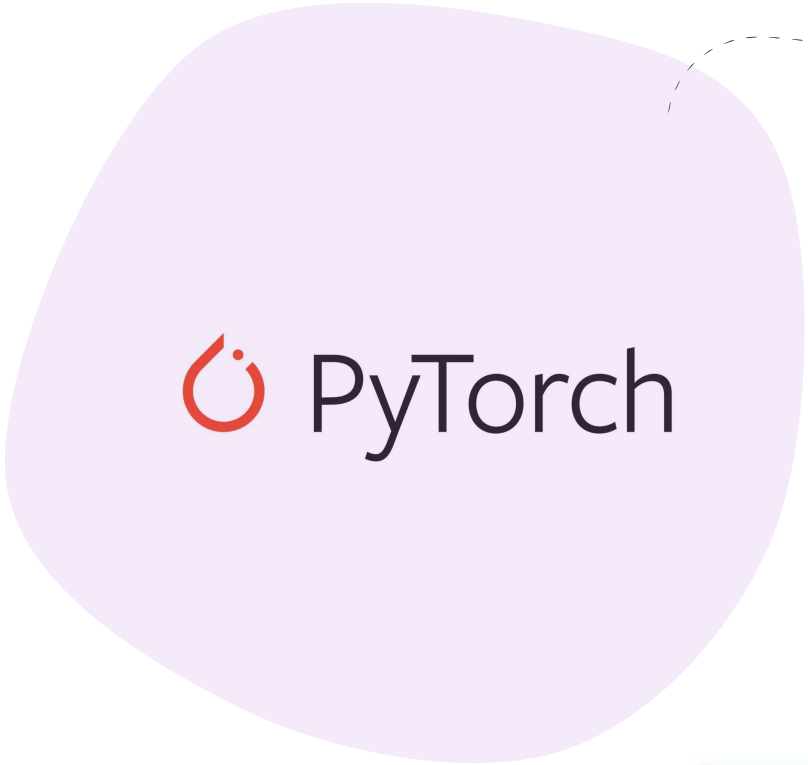
Keras

Librería de **alto nivel** desarrollada sobre TensorFlow. Es muy amigable al usuario, lo que permite implementar diversos tipos de redes neuronales de una **manera** relativamente **sencilla** y **rápida**. Sin embargo, tiene una **limitada personalización** de la **estructura** de las redes neuronales.

FUNDACIÓN YPF

Pytorch

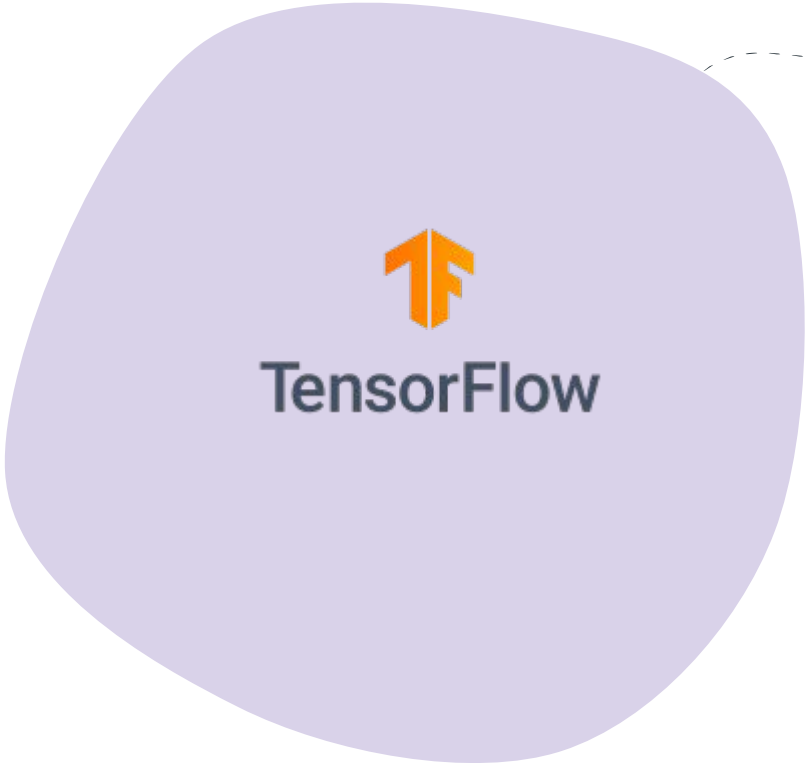
Librería de **bajo nivel** lanzada en 2017. Ofrece una manera simple de implementar redes neuronales. Sin embargo, deployar una red neuronal desarrollada en Pytorch es relativamente complejo. Es más popular en ambientes **académicos** que en la **industria.**



FUNDACIÓN YPF

TensorFlow

Librería de **bajo nivel** lanzada por Google en 2015. Tiene una curva de aprendizaje muy grande y es difícil de usar. Sin embargo, ofrece la mejor flexibilidad y personalización de estructura y de distintos features. Deployar una red neuronal desarrollada en Tensorflow es relativamente simple. Es bastante popular en la **industria.**



FUNDACIÓN YPF

Descanso

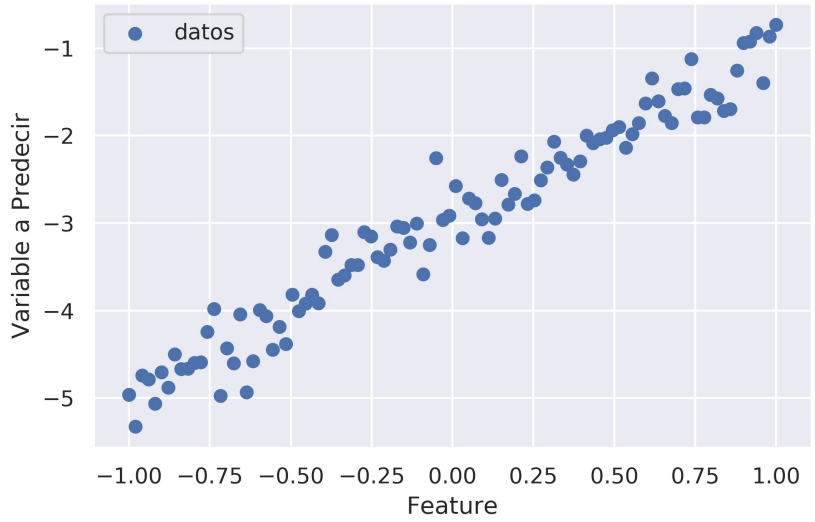
Nos vemos en 5 minutos

FUNDACIÓN YPF

Descenso por gradiente

FUNDACIÓN YPF

Buscando el mejor modelo



Buscamos Y = mX + b que mejor ajuste a los datos

m: pendiente
b: ordenada al origen

1 - Definimos **qué es *mejor ajuste a los datos*.**

2 - **Probamos con distintos valores de *m* y *b*,** y nos quedamos con el que mejor ajusta.

FUNDACIÓN YPF

Buscando el mejor modelo - Descenso

En el caso que vimos, podría hacerse una búsqueda manual o de fuerza bruta para encontrar los parámetros m y b, utilizando **función de costo.**

- **¿Es posible hacer esto en las redes neuronales?** Es muy costoso computacionalmente y podemos perder el mínimo.
- **¿Cómo lo hacemos?**

El descenso por gradiente es un algoritmo que estima numéricamente dónde una función genera sus **valores más bajos.** Es decir, que encuentra **mínimos locales** aproximando la solución con números.

FUNDACIÓN YPF

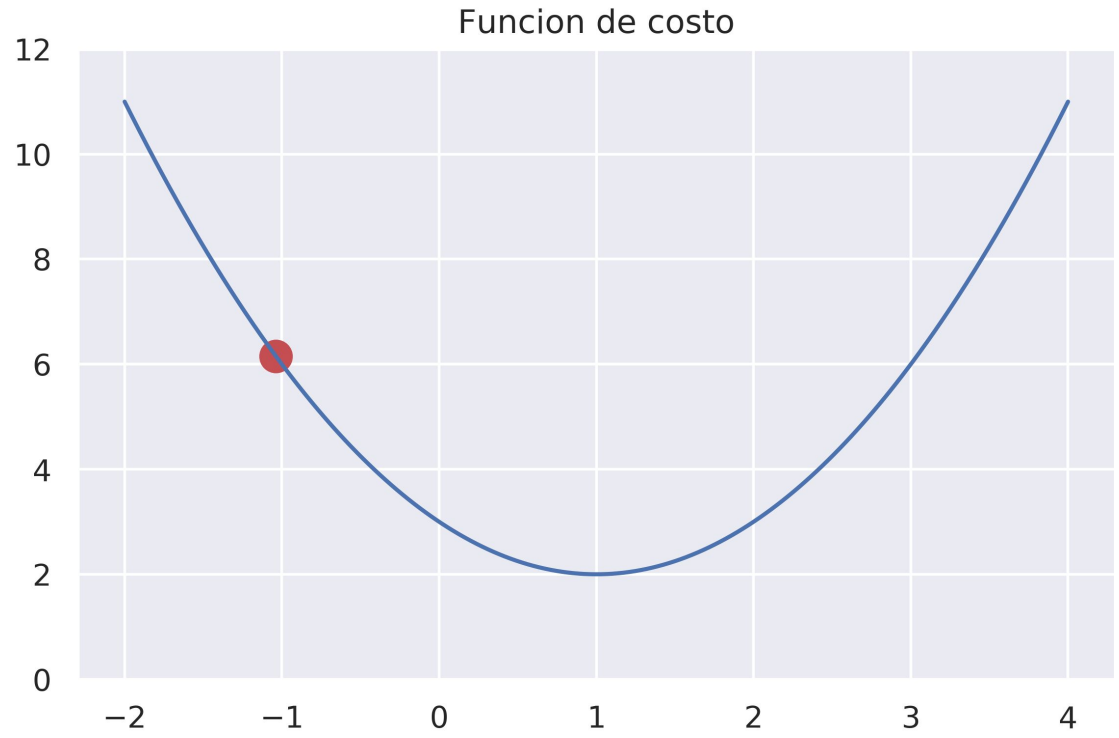
Descenso por gradiente

Queremos explorar el mínimo, pero no hicimos una exploración exhaustiva de la función de costo:

FUNDACIÓN YPF

Descenso por gradiente

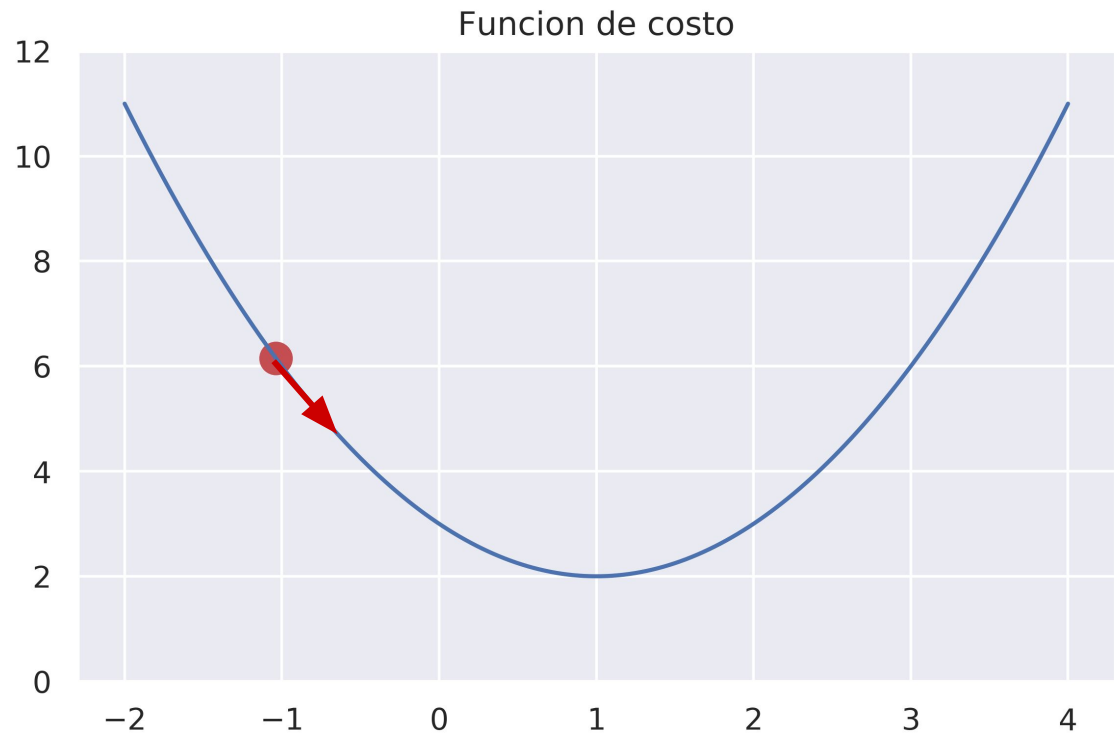
Calculamos el costo para ciertos valores al azar de los parámetros.



FUNDACIÓN YPF

Descenso por gradiente

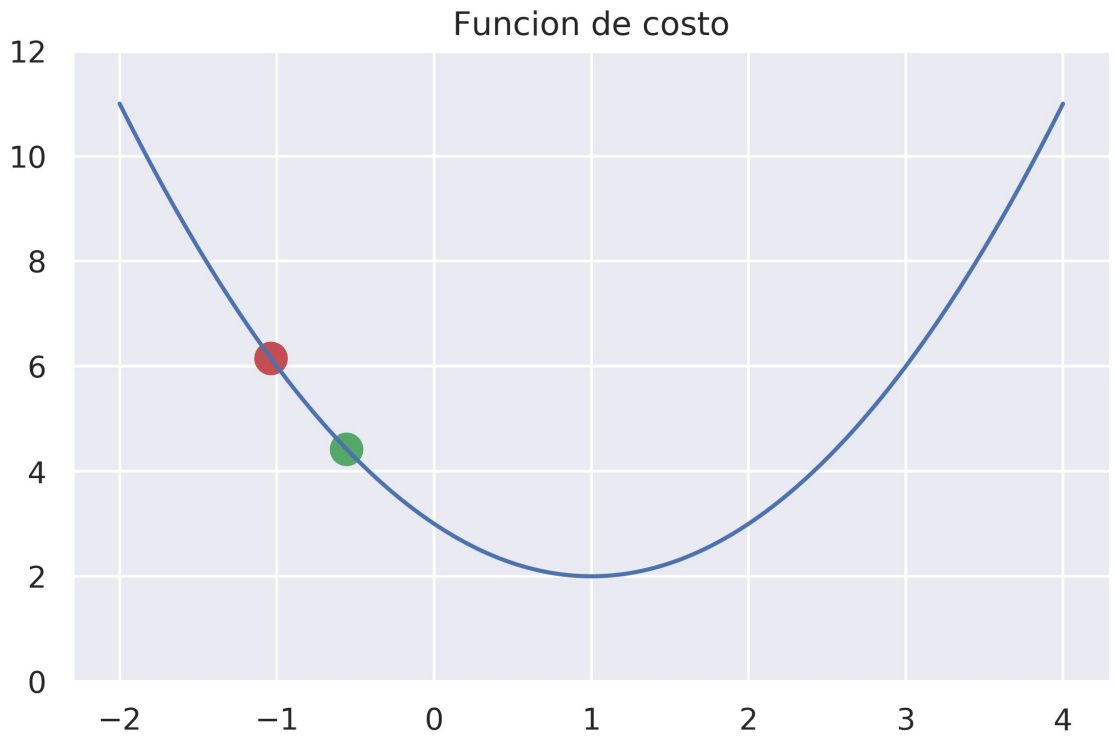
Nos fijamos la dirección de decrecimiento en ese punto. O sea, derivamos o calculamos el gradiente.



FUNDACIÓN YPF

Descenso por gradiente

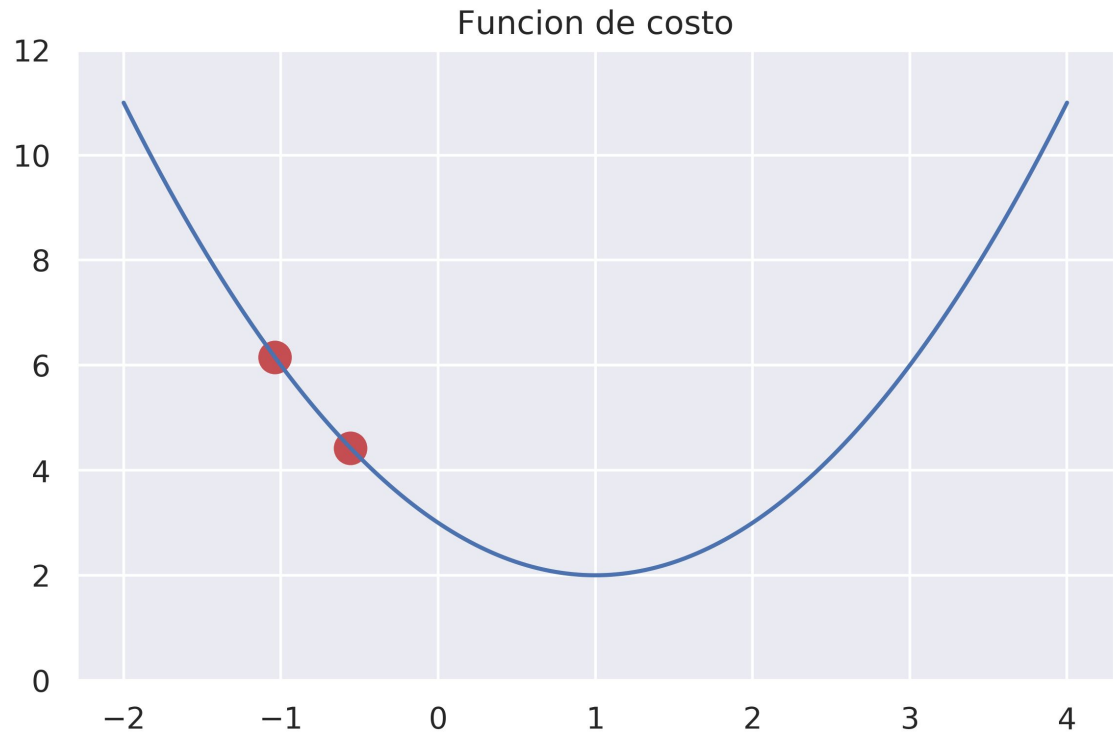
Actualizamos los valores de los parámetros.



FUNDACIÓN YPF

Descenso por gradiente

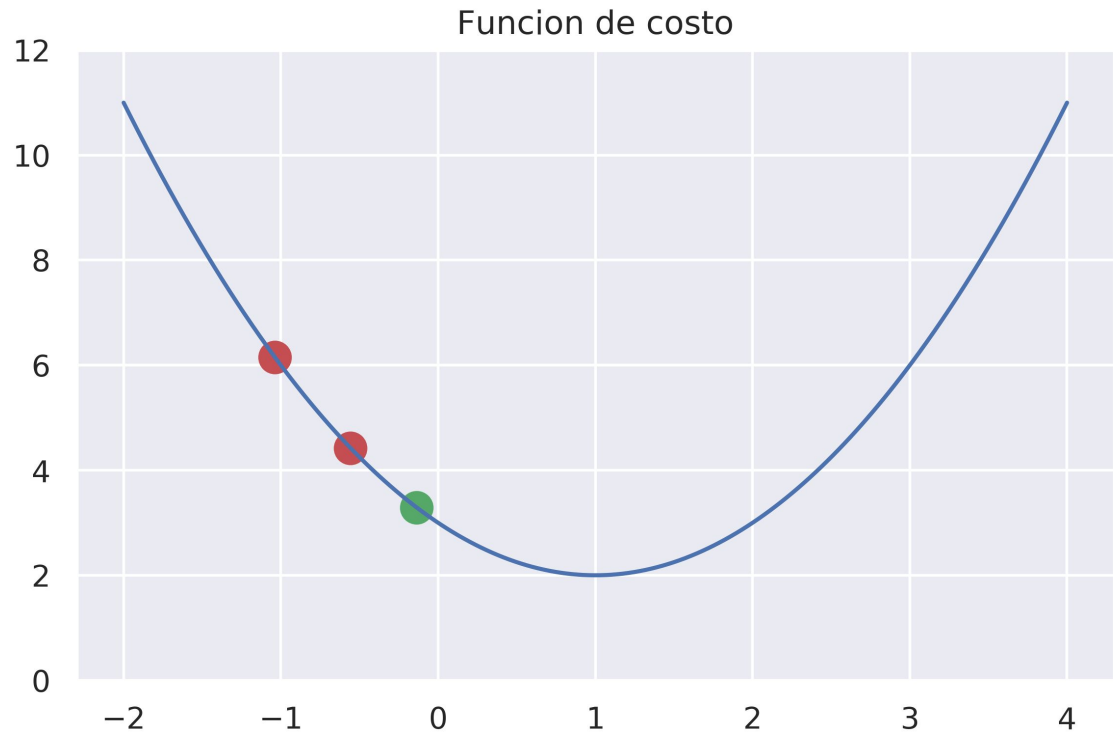
Y repetimos hasta converger: dirección de decrecimiento en el punto



FUNDACIÓN YPF

Descenso por gradiente

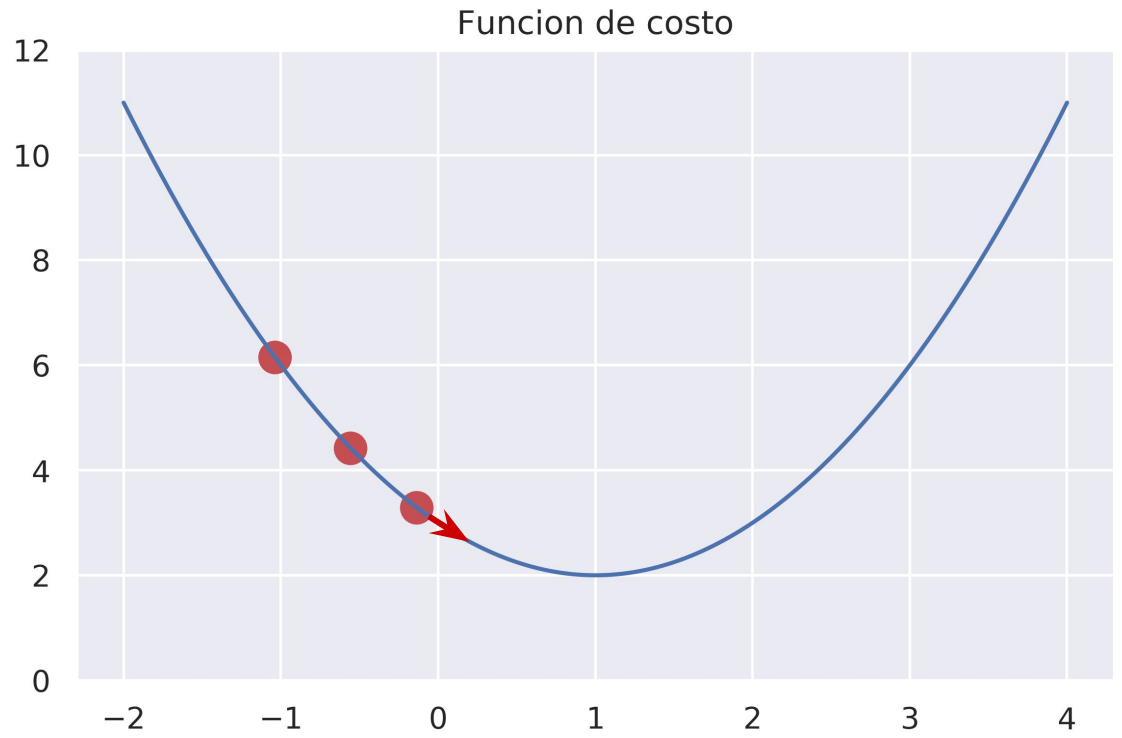
Y repetimos hasta converger: **actualizamos los valores de los parámetros.**



FUNDACIÓN YPF

Descenso por gradiente

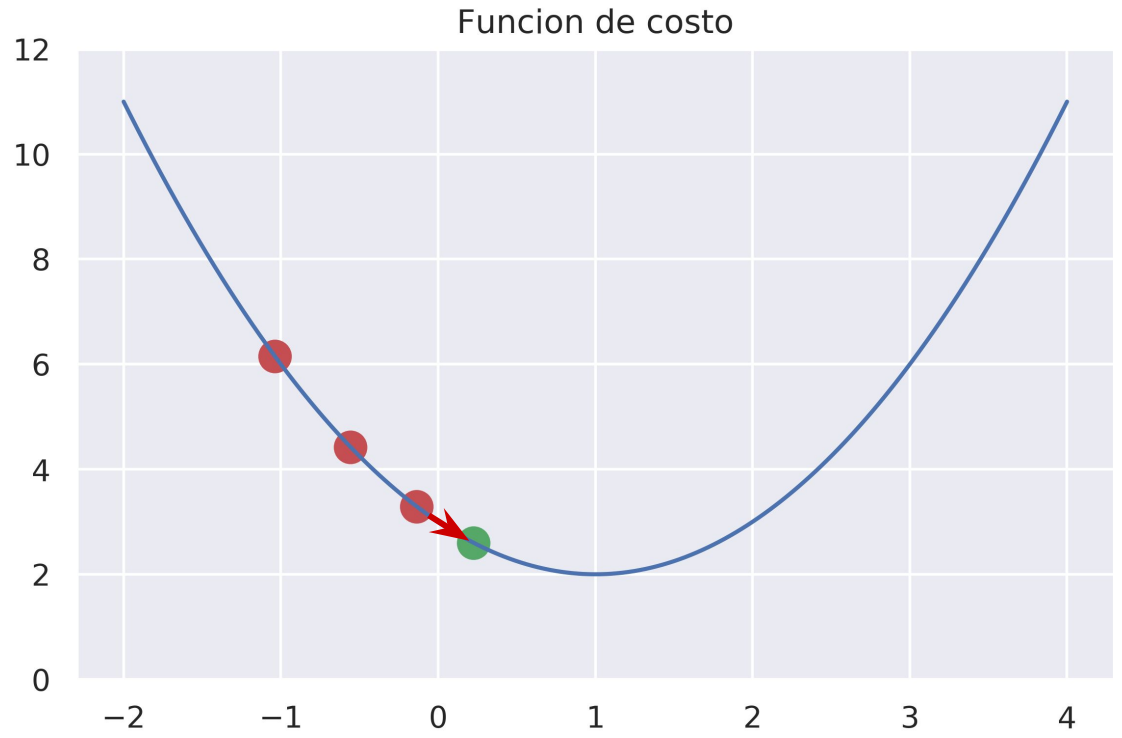
Y repetimos hasta converger: **dirección de decrecimiento en el punto**



FUNDACIÓN YPF

Descenso por gradiente

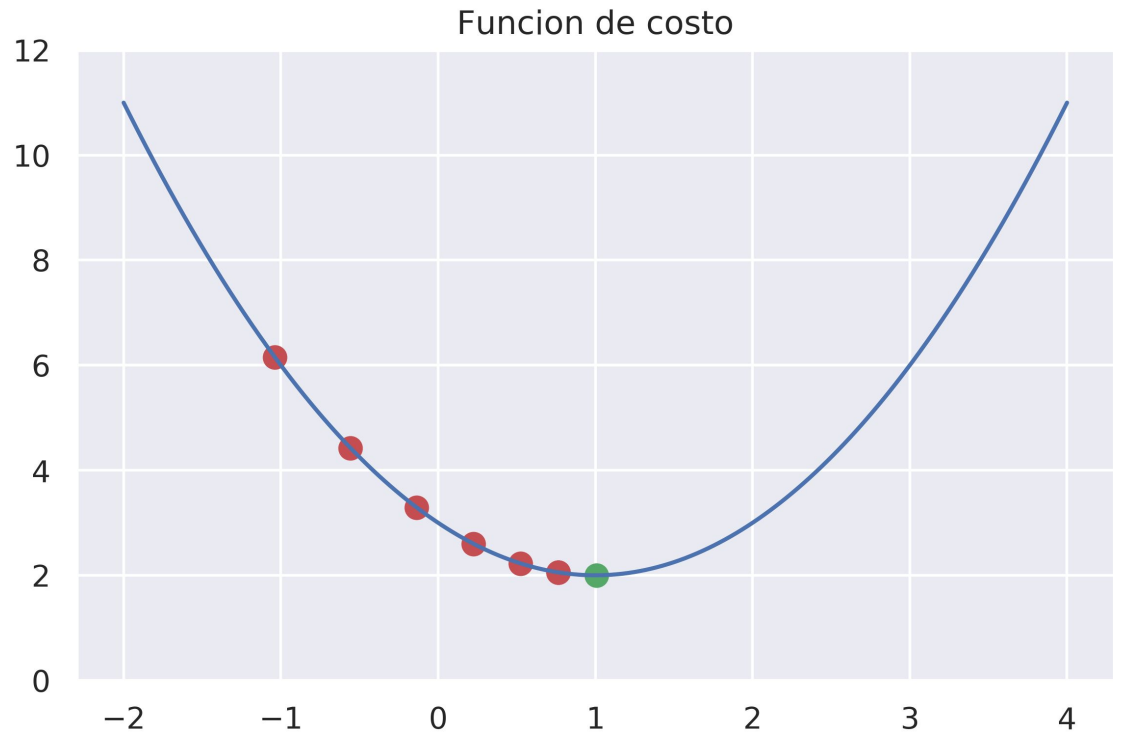
Y repetimos hasta converger: **actualizamos** los valores de los parámetros.



FUNDACIÓN YPF

Descenso por gradiente

Encontramos el mínimo de la función



FUNDACIÓN YPF

Descenso por gradiente

- En cualquier algoritmo de machine learning, es necesaria una **función de costo/pérdida** que depende del problema (clasificación, regresión, etc).
- La función de costo es una función de los parámetros de la red neuronal.
- Los mejores parámetros de la red son aquellos que **minimicen la función de costo.**
- Es imposible explorar el espacio de parámetros exhaustivamente. Por lo cual, se utiliza una técnica que lo haga eficientemente: **Descenso por Gradiente**.

FUNDACIÓN YPF

Descanso

Nos vemos en 10 minutos

FUNDACIÓN YPF

Repasamos en Kahoot

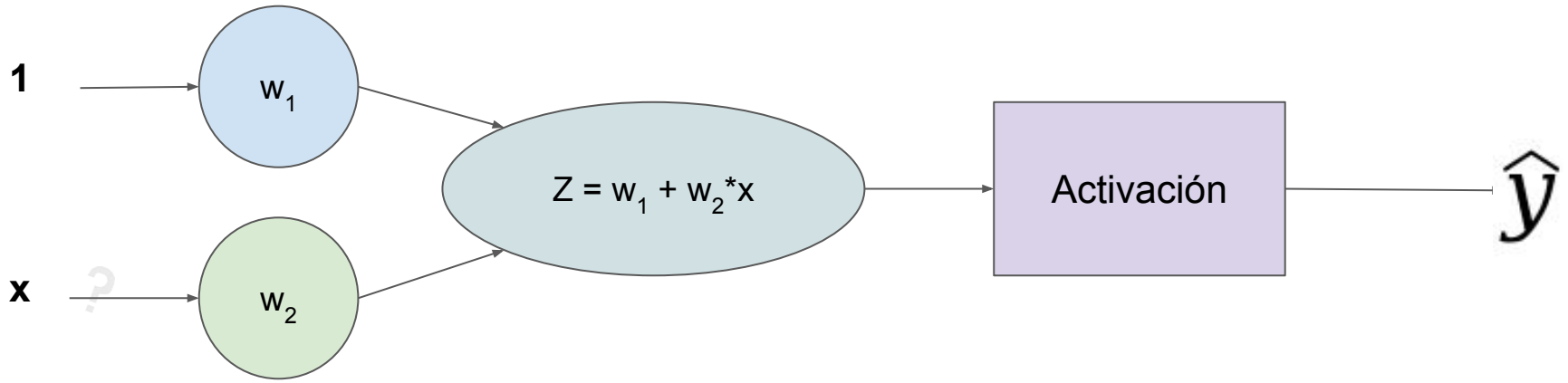
FUNDACIÓN YPF

Perceptrón simple

FUNDACIÓN YPF

Perceptrón simple

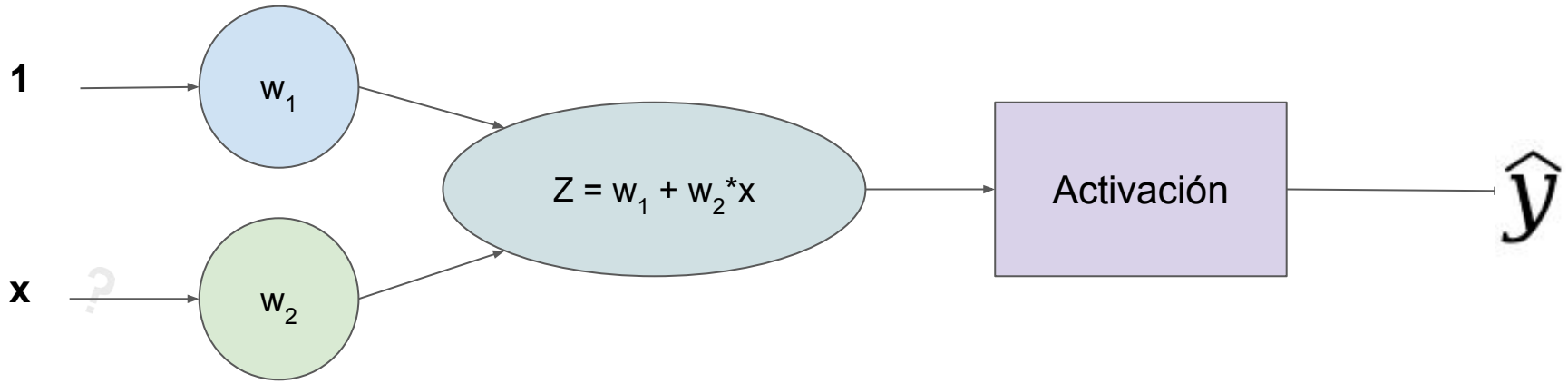
Necesitamos una función que, dado los features, devuelva probabilidades entre 0 y 1



FUNDACIÓN YPF

Perceptrón simple

Necesitamos una función que, dado los features, devuelva probabilidades entre 0 y 1



Entradas: Features + Bias

Pesos: Lo que buscamos en el entrenamiento

Unión sumadora: combina los features con los pesos

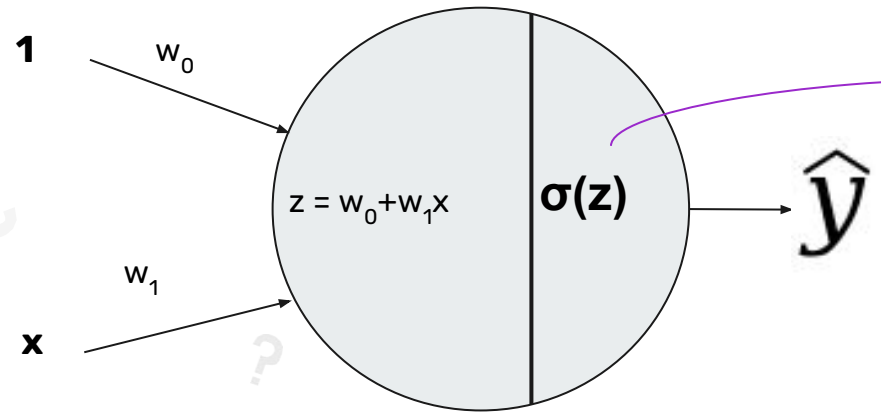
Activación: si no está, es una función lineal

Salida: entre 0 y 1, probabilidad del evento

FUNDACIÓN YPF

Perceptrón simple

Necesitamos una función que, dado los features, devuelva probabilidades entre 0 y 1



Activación:

- Sin la activación, es una función lineal
- Se debe introducir una función que *sature* la entrada en 0 o en 1 dependiendo del resultado de la unión sumadora

***Función de activación*:**

- Sigmoidea

FUNDACIÓN YPF

Función Sigmoidea

Transforma los valores introducidos a una escala (0,1), donde los valores altos tienen de manera asintótica a 1 y los valores muy bajos tienden de manera asintótica a 0.

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

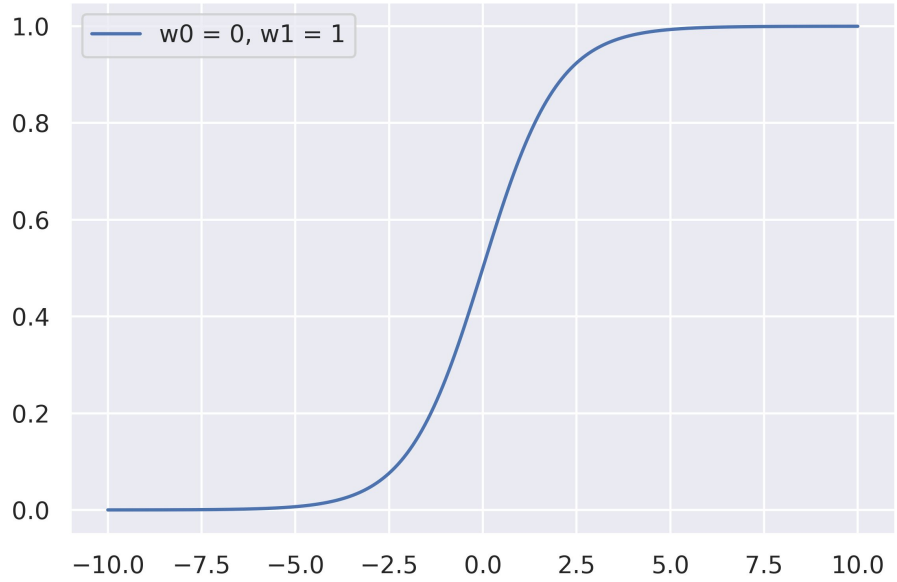
FUNDACIÓN YPF

Función Sigmoidea

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

$$z = w_0 + w_1 x$$

$$y(x) = \frac{1}{1 + e^{-(w_0 + w_1 x)}}$$



FUNDACIÓN YPF

Función Sigmoidea

Características:

- Sufre de *vanishing gradient*
- Lenta convergencia.
- No está centrada en el cero.
- Está acotada entre 0 y 1.
- Tiene un excelente rendimiento para la **capa de salida**.

FUNDACIÓN YPF

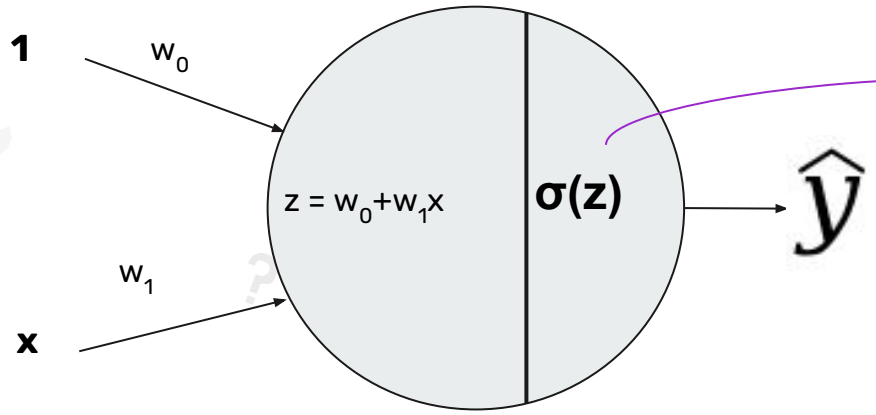
Vanishing Gradient

Cuando los pesos de la red neuronal reciben una actualización proporcional a la derivada parcial de la función de error con respecto al peso actual en cada iteración de entrenamiento, el gradiente se irá desvaneciendo a valores muy pequeños, impidiendo eficazmente el peso de cambiar su valor. En el caso peor, esto puede impedir que la red neuronal continúe su entrenamiento.

FUNDACIÓN YPF

Perceptrón simple

Necesitamos una función que, dado los features, devuelva probabilidades entre 0 y 1



Activación:

- Sin la activación, es una función lineal
- Se debe introducir una función que *sature* la entrada en 0 o en 1 dependiendo del resultado de la unión sumadora

Debemos encontrar los pesos ***w0 y w1*** apropiados, para ello necesitamos una **función de costo**.

FUNDACIÓN YPF

CUARTA Pre- Entrega

En esta clase, deberán realizar la cuarta pre-entrega
La misma incluirá los desafíos vistos, en relación a los modelos de aprendizaje no supervisado.

- Deberán desarrollar un modelo de clustering, desde el análisis exploratorio hasta la mejora
- Elegir algún algoritmo y aplicarlo a su proyecto.
- Será importante que hagan una revisión y tomen decisiones, para aplicar lo visto con un criterio que deberán explicitar.

Presentarán lo trabajado entregando el link a su Github en el foro del aula virtual

FUNDACIÓN YPF

¿DUDAS?

FUNDACIÓN YPF

¡Muchas gracias!