FUNDACIÓN YPF

Módulo VII: Introducción a Redes Neuronales y Deep Learning.

Clase 26: Perceptrón simple y multicapa

FUNDACIÓN YPF

¿Ponemos a grabar el taller?

FUNDACIÓN YPF

¿QUÉ VAMOS A VER HOY?

- Entropía cruzada
- Propagation

- Perceptrón Multicapa

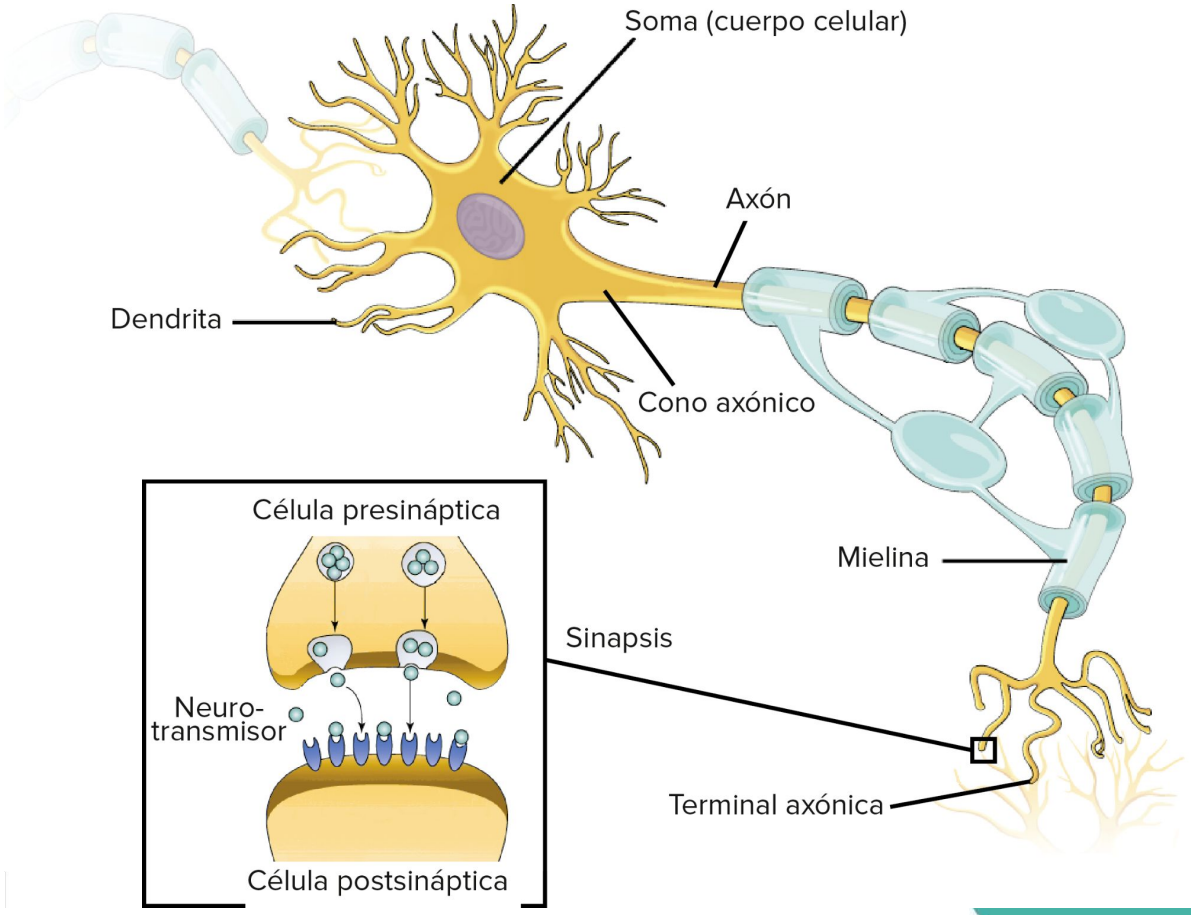
FUNDACIÓN YPF

Repasemos

FUNDACIÓN YPF

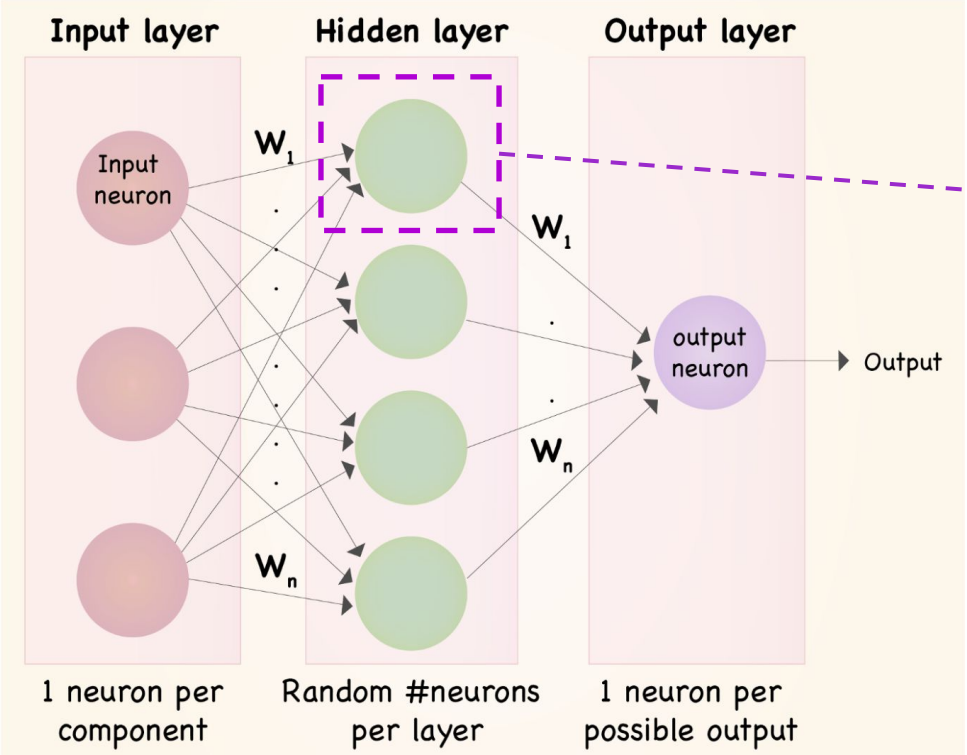
Redes Neuronales

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



FUNDACIÓN YPF

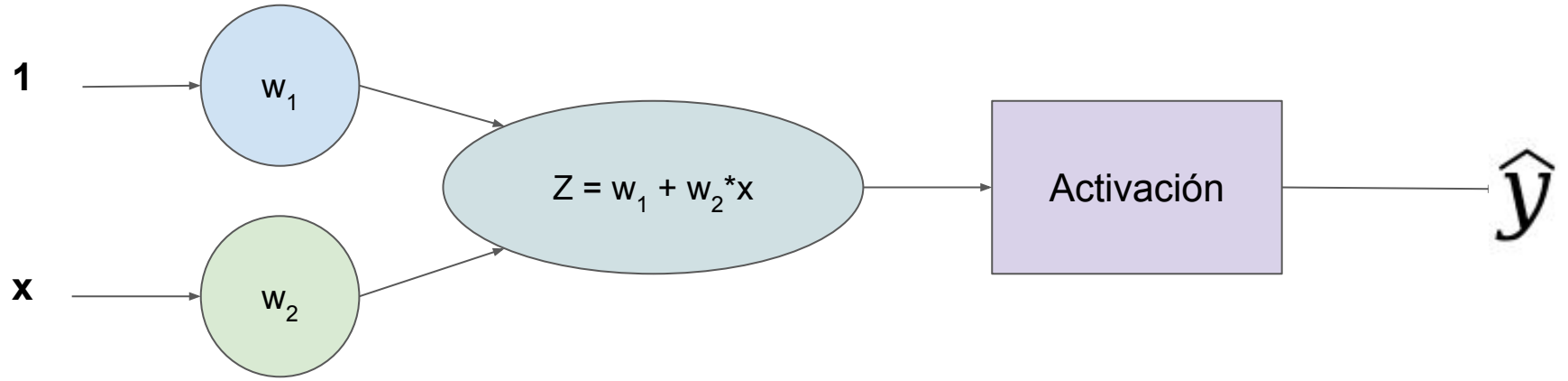
Redes Neuronales



Neurona: Unidades de procesamiento que intercambian datos o información

FUNDACIÓN YPF

Neurona básica



Perceptrón Simple

FUNDACIÓN YPF

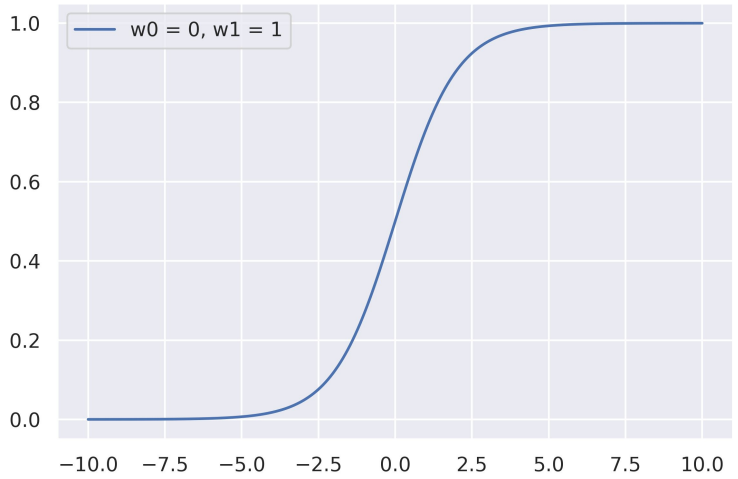
Descenso por gradiente

- 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

Función Sigmoidea

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

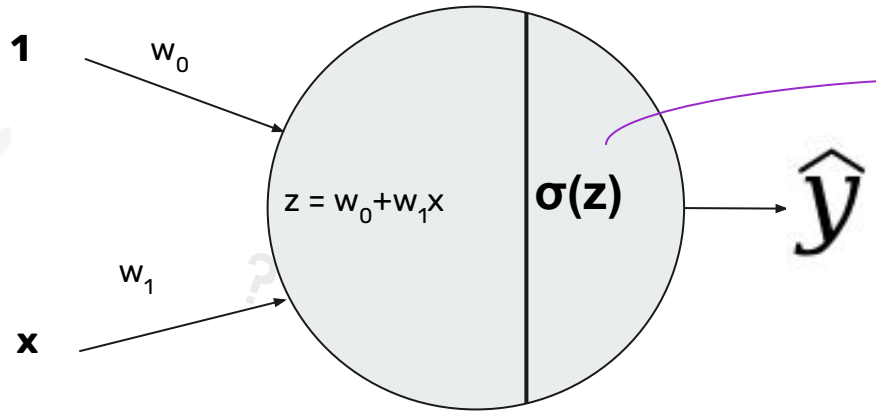


Transforma los valores introducidos a una escala (0,1), donde los valores altos tienen de manera asintótica a 1 y los valores muy bajos tienden de manera asintótica a 0.

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

Entropía cruzada

FUNDACIÓN YPF

Entropía cruzada

Pérdida para una instancia

$$L(\hat{y}, y) = -y * log(\hat{y}) - (1 - y) * log(1 - \hat{y})$$

Es una función de pérdida entre una etiqueta (y) y la probabilidad de pertenecer o no a esa etiqueta.

FUNDACIÓN YPF

Entropía cruzada

Pérdida para una instancia

$$L(\hat{y}, y) = -y * log(\hat{y}) - (1 - y) * log(1 - \hat{y})$$

Costo para todas las instancias

$$J(\overline{W}) = \frac{1}{n} \sum_{i=0}^{n-1} L(\widehat{y^{(i)}}, y^{(i)})$$

Es una función de pérdida entre una etiqueta (y) y la probabilidad de pertenecer o no a esa etiqueta.

FUNDACIÓN YPF

Entropía cruzada

Características:

- Difícil diferenciación y convergencia.
- Escala univariante.
- Simétrica.
- Es fácil de interpretar.

Es una función de pérdida entre una etiqueta (y) y la probabilidad de pertenecer o no a esa etiqueta.

FUNDACIÓN YPF

Propagation

FUNDACIÓN YPF

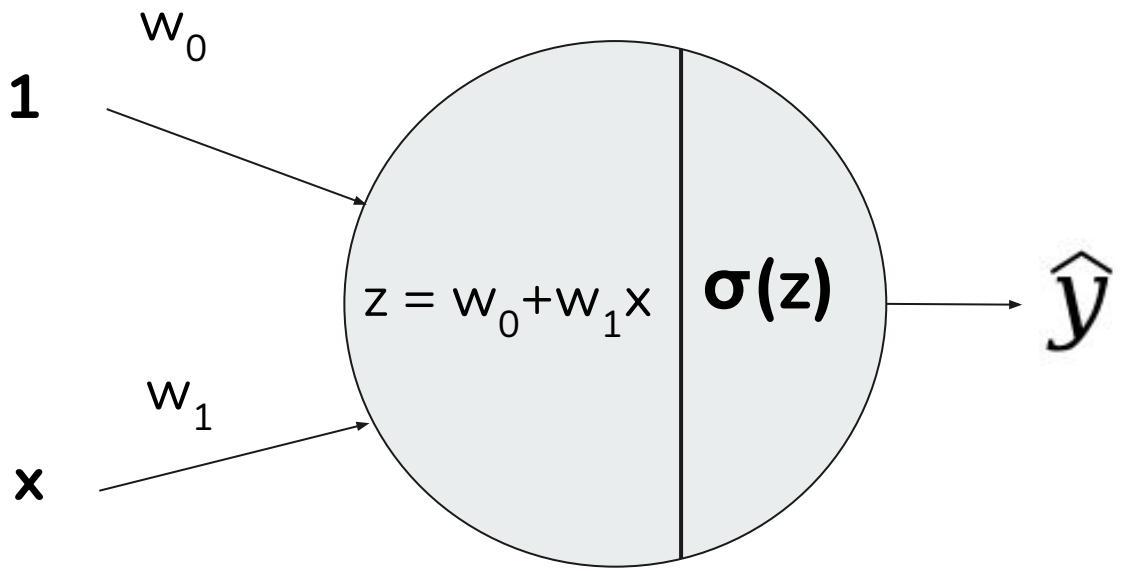
Propagation

- Descenso por gradiente calcula la derivada del costo y utiliza los valores para actualizar los parámetros. Este proceso se repite varias veces hasta que llega al mínimo, o sea, converge.
- En cada una de esas iteraciones, tiene que calcular el costo. El costo depende de las instancias de entrenamiento y de los parámetros que tengamos hasta ese momento.

Calcular el costo con las instancias de entrenamiento es lo que se conoce como **Forward Propagation.**

FUNDACIÓN YPF

Propagation

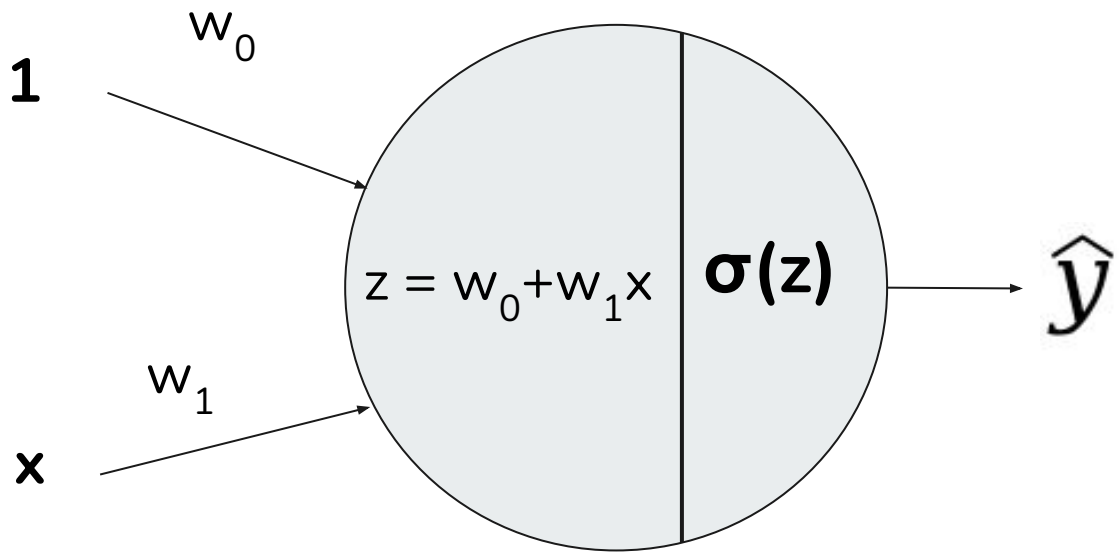


Entran una o varias instancias

Calculamos la predicción sobre esas instancias

FUNDACIÓN YPF

Propagation



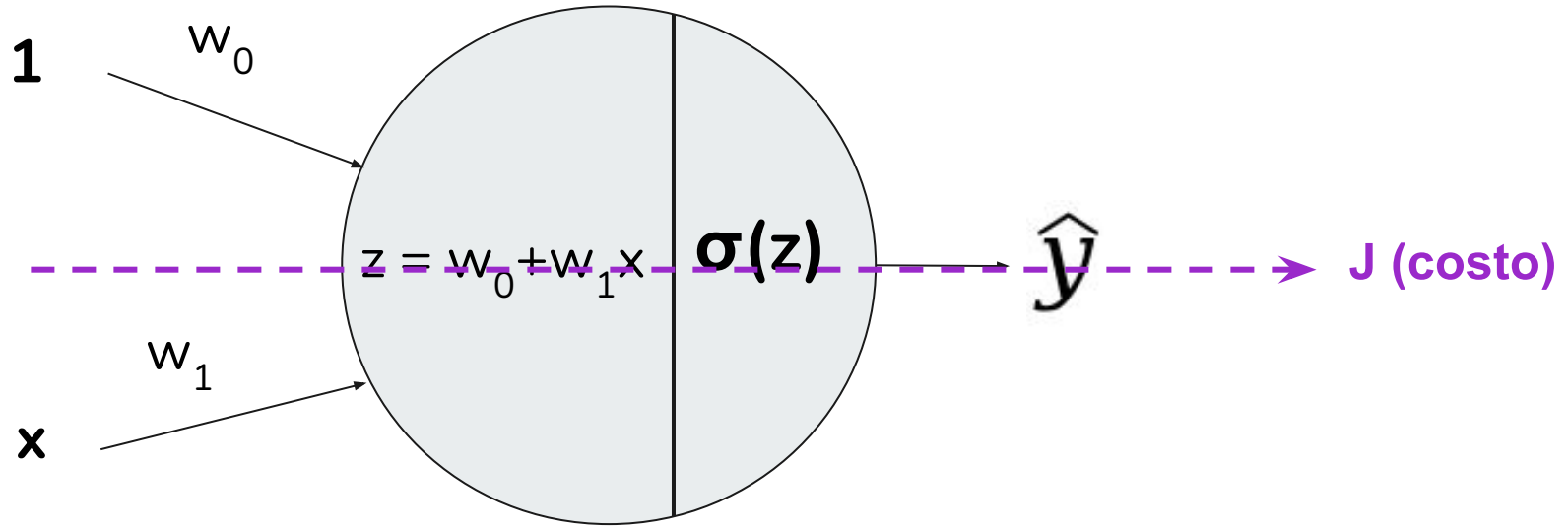
Entran una o varias instancias

Calculamos la predicción sobre esas instancias

Se calcula el costo de esas predicciones

FUNDACIÓN YPF

Propagation



Entran una o varias instancias

Calculamos la predicción sobre esas instancias

Se calcula el costo de esas predicciones

FUNDACIÓN YPF

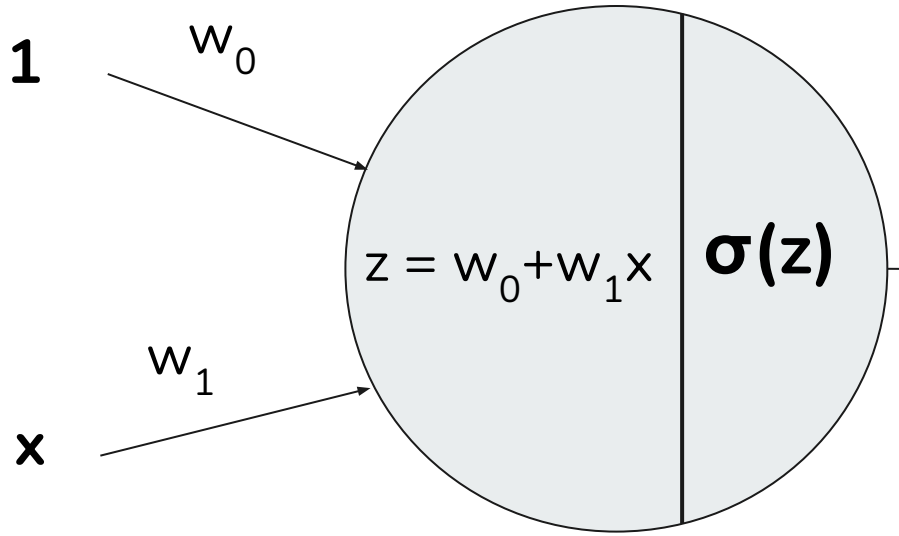
Propagation

- Con el costo calculado, queremos actualizar los valores de los parámetros.
- Para eso, tenemos que derivar el costo y propagar esa derivada hacia atrás, hasta llegar a los parámetros w0 y w1.

Calcular las derivadas y actualizar los parámetros “hacia atrás” se conoce como **Backpropagation**.

FUNDACIÓN YPF

Propagation



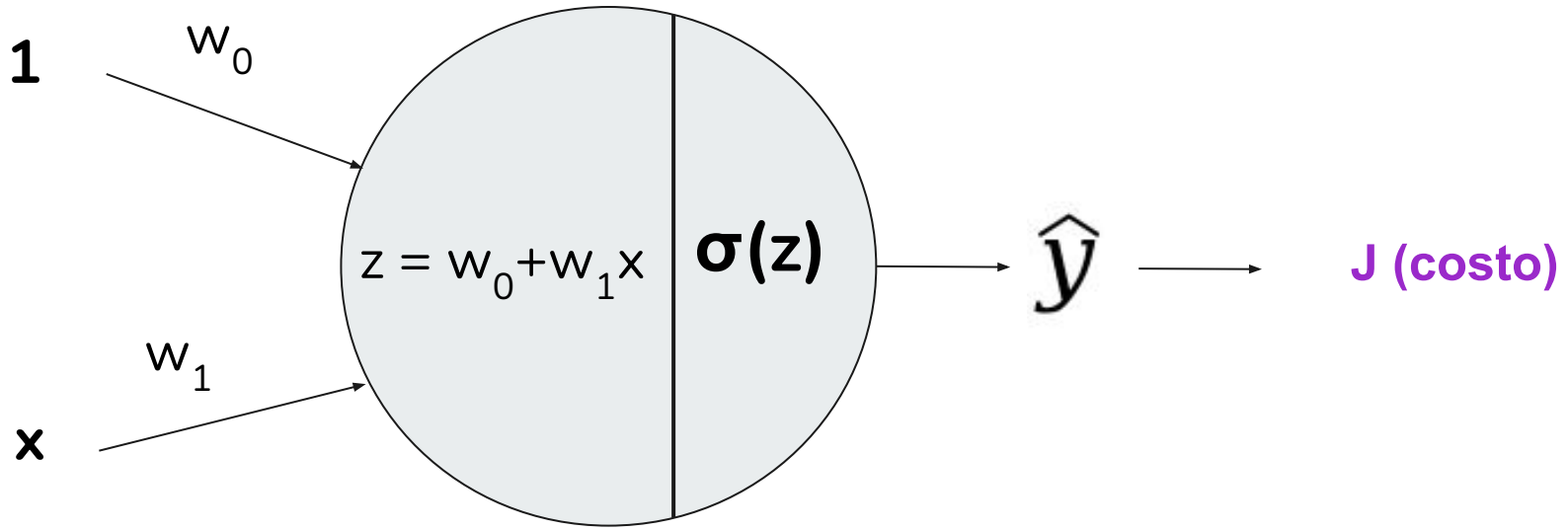
$\hat{y}$ → J (costo)

Se deriva la predicción con respecto z

Se deriva el costo con respecto a las predicciones

FUNDACIÓN YPF

Propagation



Se actualizan los parámetros

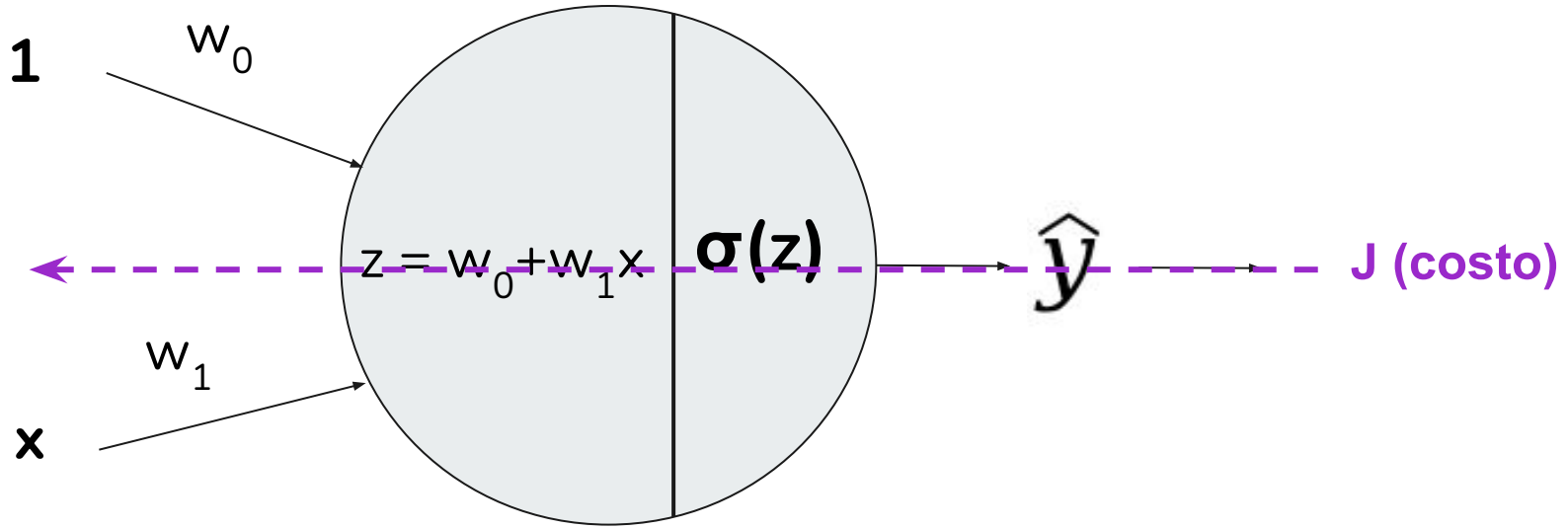
Se deriva z con respecto a w_0 y w_1

Se deriva la predicción con respecto z

Se deriva el costo con respecto a las predicciones

FUNDACIÓN YPF

Propagation



Se actualizan los parámetros

Se deriva z con respecto a w_0 y w_1

Se deriva la predicción con respecto z

Se deriva el costo con respecto a las predicciones

FUNDACIÓN YPF

Perceptrón simple…

El perceptrón simple es que solo encuentra fronteras lineales.

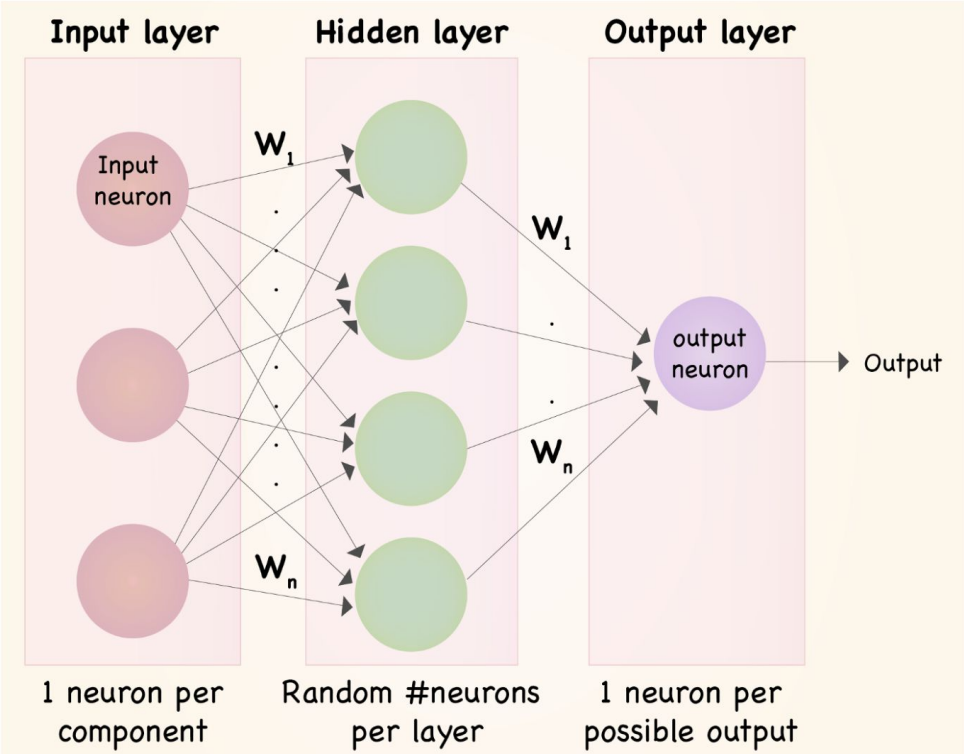
¿Qué hacemos entonces?

FUNDACIÓN YPF

Perceptrón multicapa

FUNDACIÓN YPF

Perceptrón Multicapa

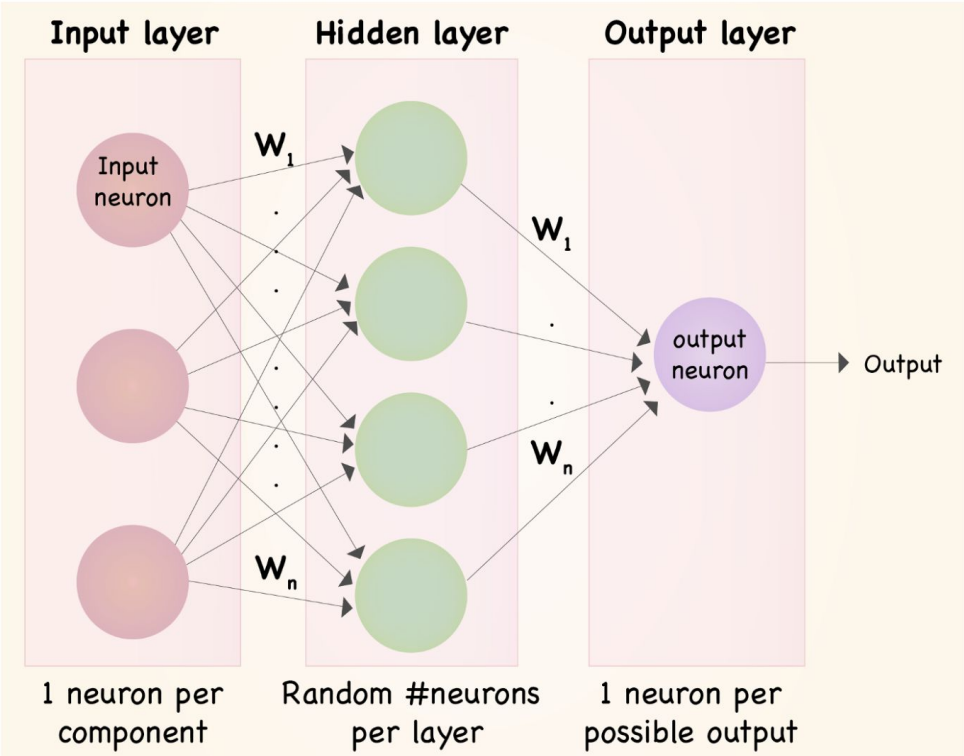


Cada neurona tiene sus propios pesos/parámetros.

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

FUNDACIÓN YPF

Perceptrón Multicapa

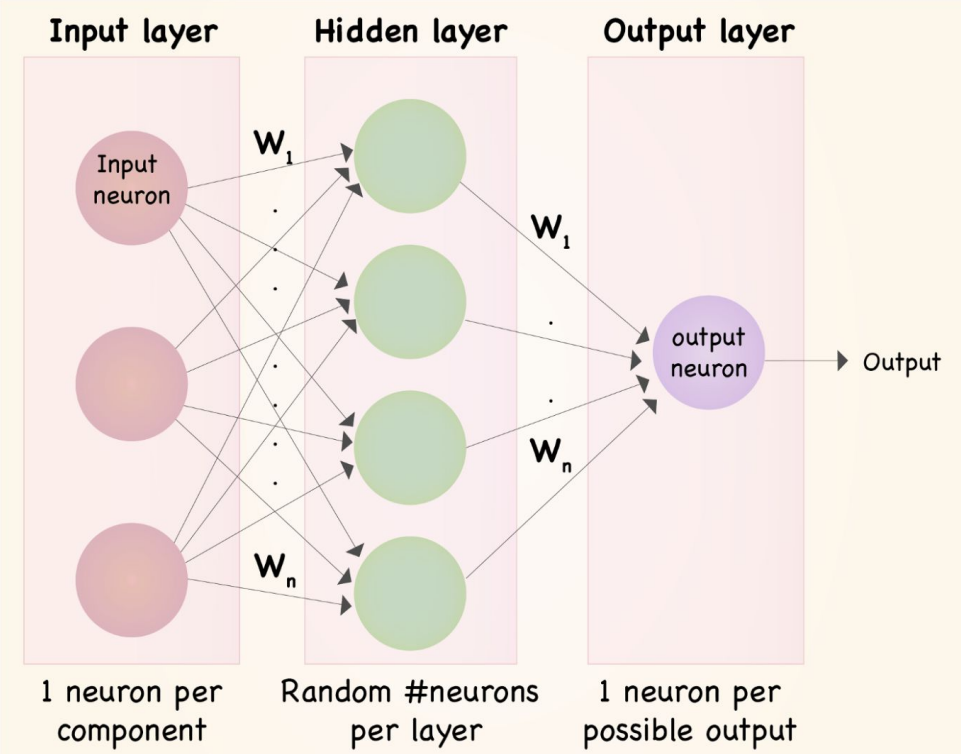


En las neuronas del perceptrón multicapa o red neuronal, se aplican los conceptos vistos:

- Descenso por gradiente
- Función de costo
- Forward Propagation
- Backpropagation

FUNDACIÓN YPF

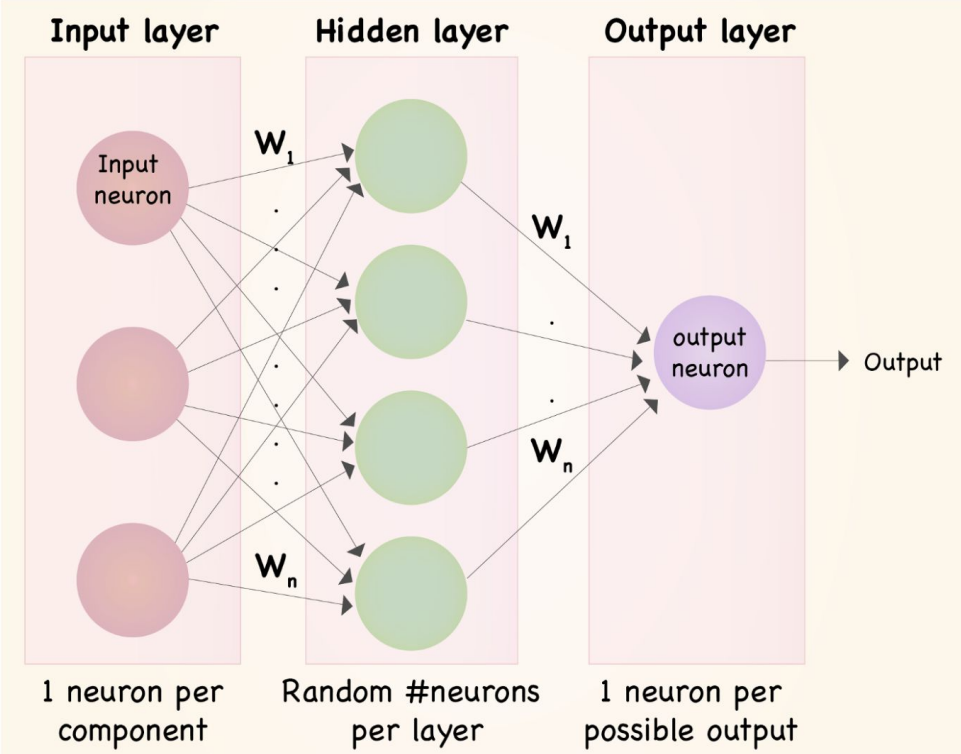
Perceptrón Multicapa



Como tenemos más neuronas, podemos hacer una **clasificación multiclase**: La cantidad de neuronas en la capa de salida tiene que ser igual a la cantidad de clases buscadas.

FUNDACIÓN YPF

Perceptrón Multicapa



Tenemos más opciones para las funciones de activación:

- Sigmoidea
- Tanh
- ReLu
- Leaky ReLu

FUNDACIÓN YPF

Descanso

Nos vemos en 10 minutos

FUNDACIÓN YPF

Repasamos en Kahoot

FUNDACIÓN YPF

¿DUDAS?

FUNDACIÓN YPF

¡Muchas gracias!