FUNDACIÓN YPF

Módulo VII: Introducción a Redes Neuronales y Deep Learning.

Clase 27: Perceptrón multicapa (2)

FUNDACIÓN YPF

¿Ponemos a grabar el taller?

FUNDACIÓN YPF

¿QUÉ VAMOS A VER HOY?

- Repaso

- Funciones de activación
- Implementación de una red neuronal

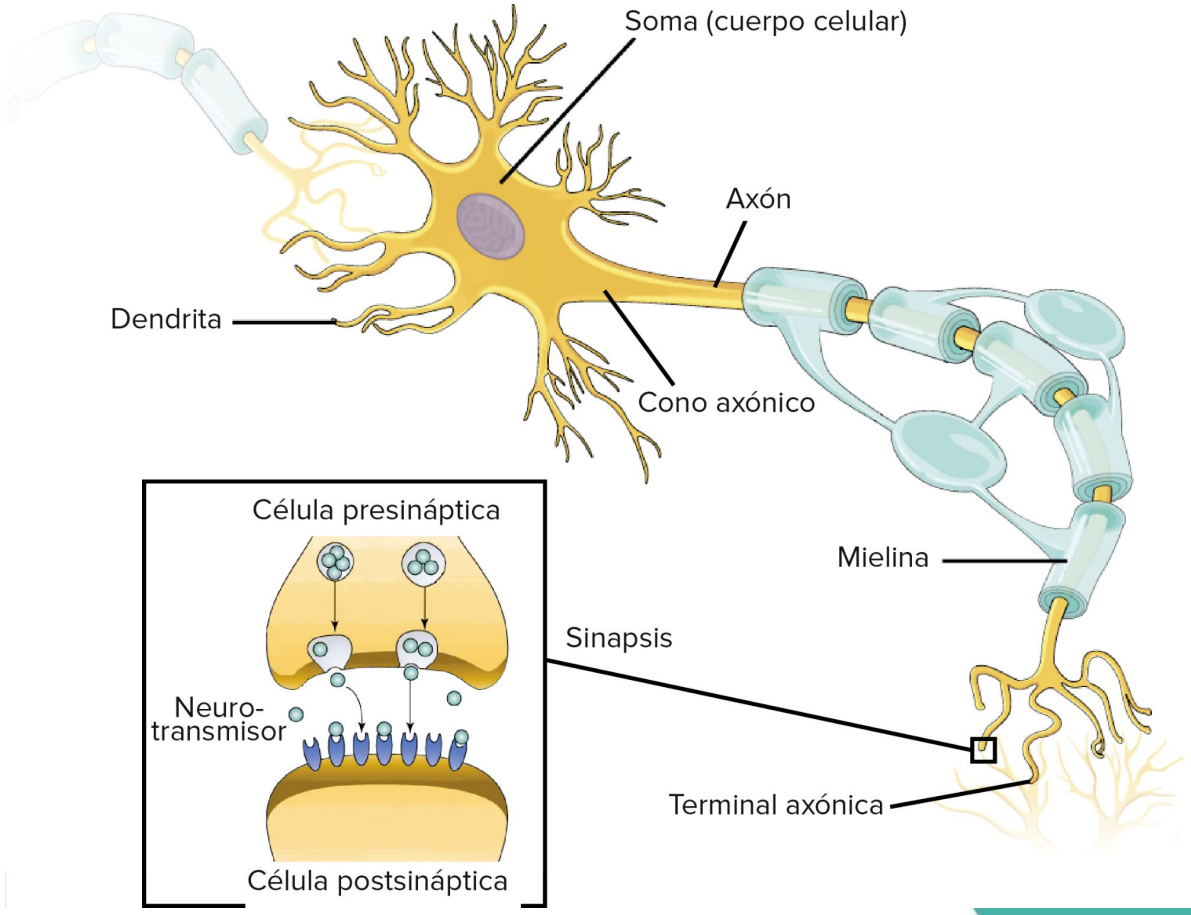
FUNDACIÓN YPF

Repasemos

FUNDACIÓN YPF

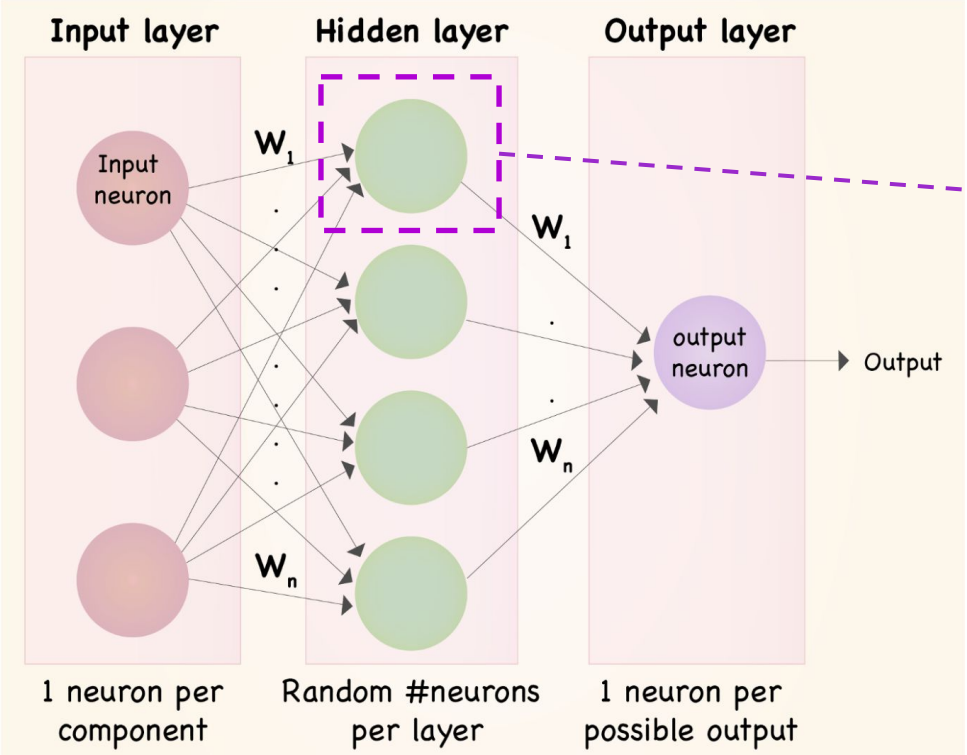
Redes Neuronales

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



FUNDACIÓN YPF

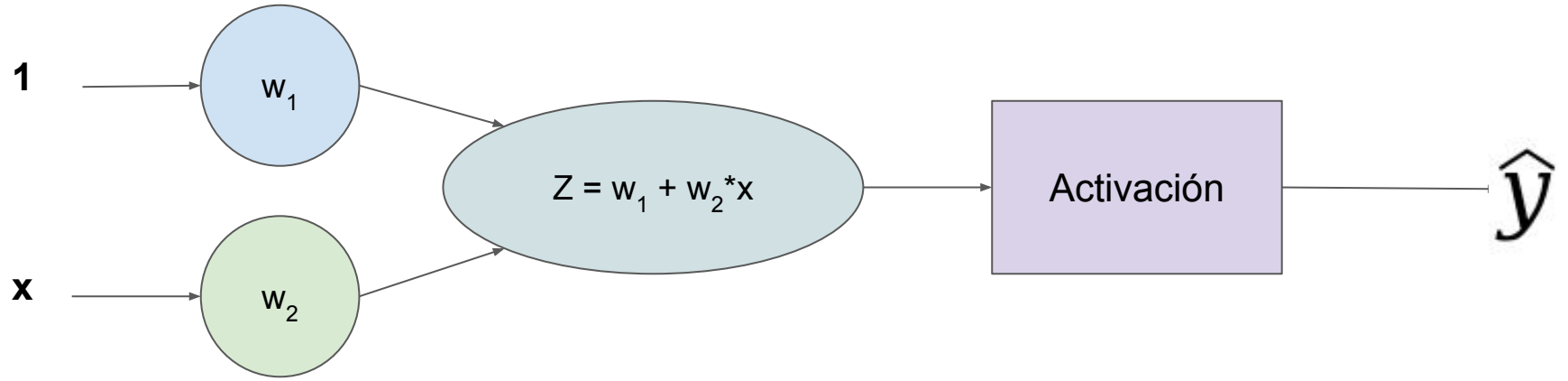
Redes Neuronales



Neurona: Unidades de procesamiento que intercambian datos o información

FUNDACIÓN YPF

Neurona básica

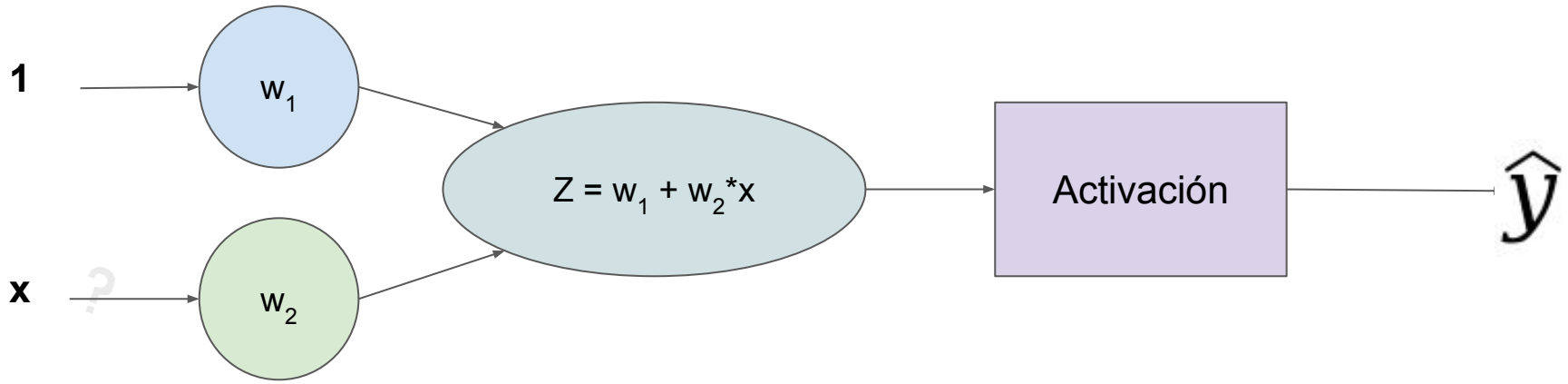


Perceptrón Simple

FUNDACIÓN YPF

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

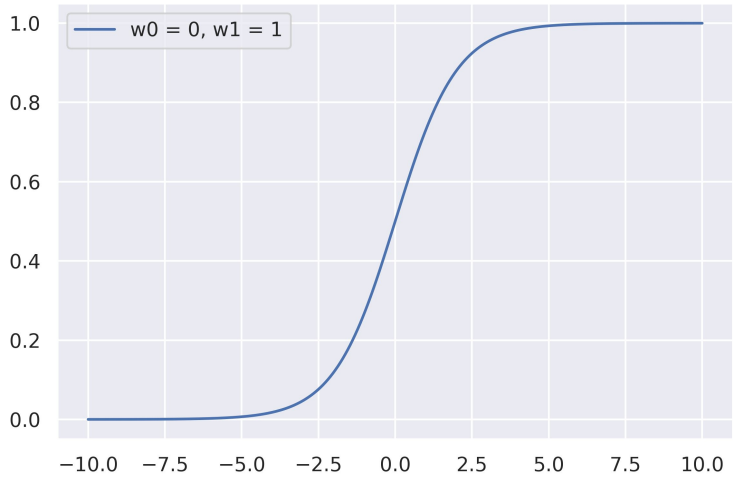
Descenso por gradiente

- 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

Entropía cruzada

Pérdida para una instancia

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

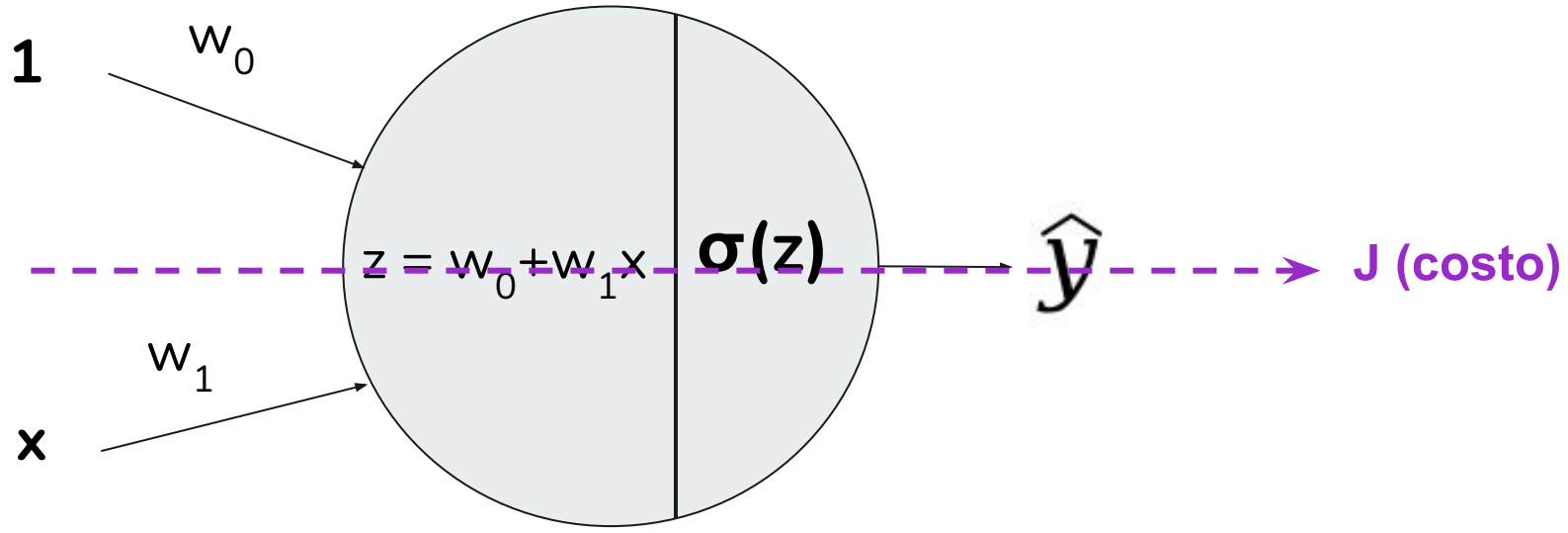
Costo para todas las instancias

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

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

FUNDACIÓN YPF

Forward Propagation



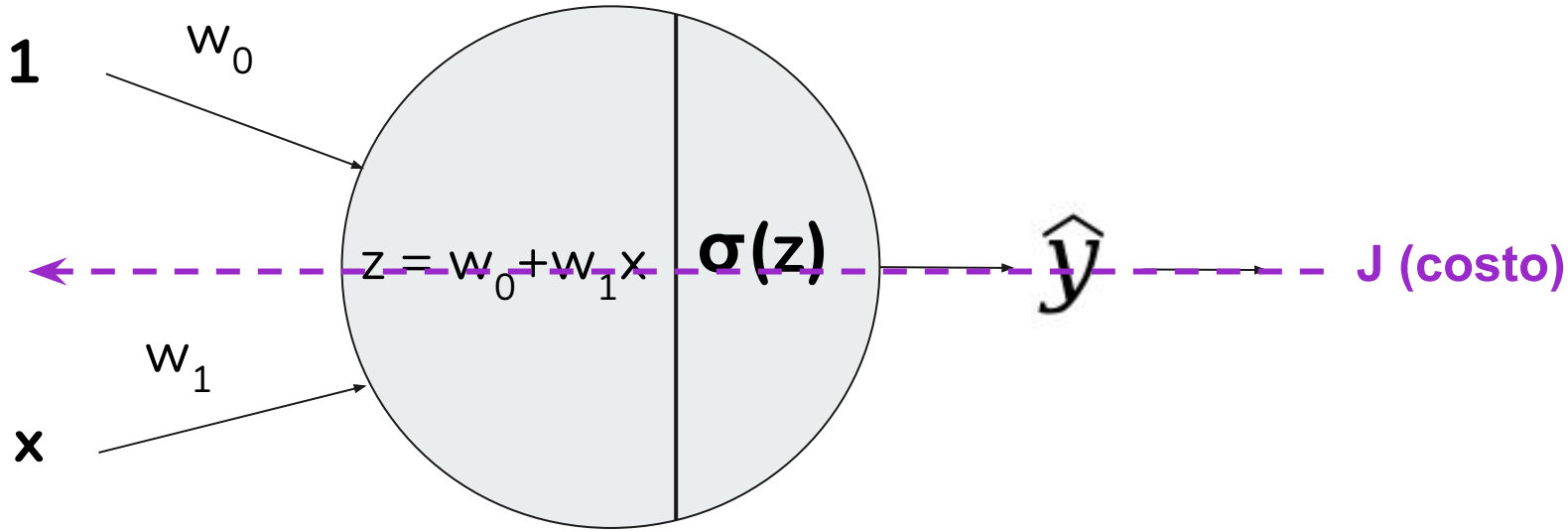
Entran una o varias instancias

Calculamos la predicción sobre esas instancias

Se calcula el costo de esas predicciones

FUNDACIÓN YPF

BACK Propagation



Se actualizan los parámetros

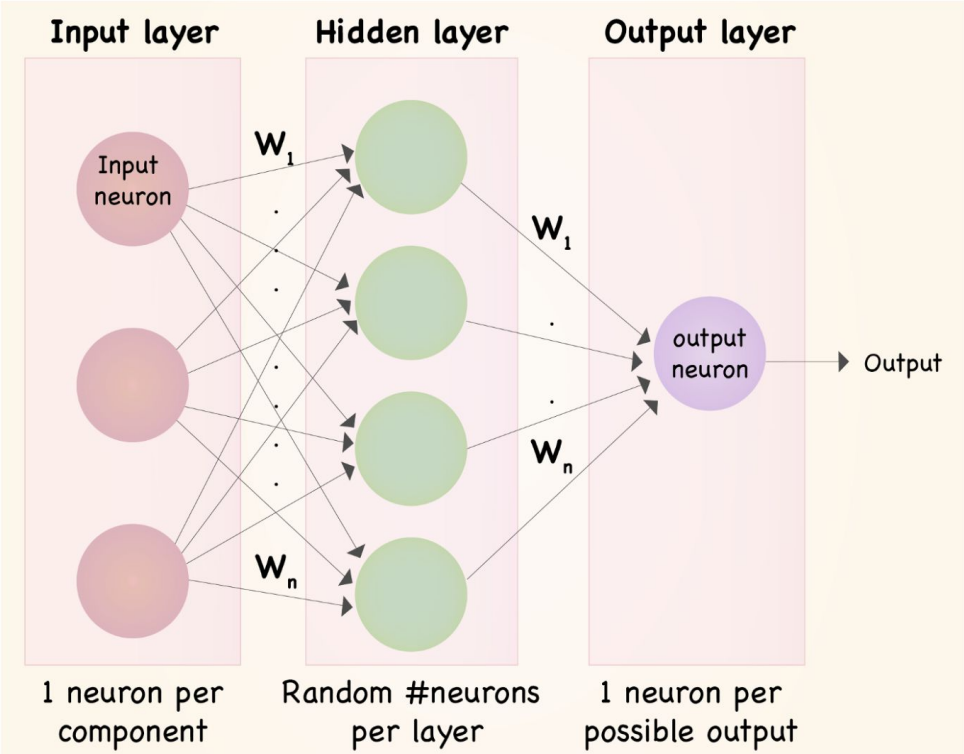
Se deriva z con respecto a w_0 y w_1

Se deriva la predicción con respecto z

Se deriva el costo con respecto a las predicciones

FUNDACIÓN YPF

Perceptrón Multicapa

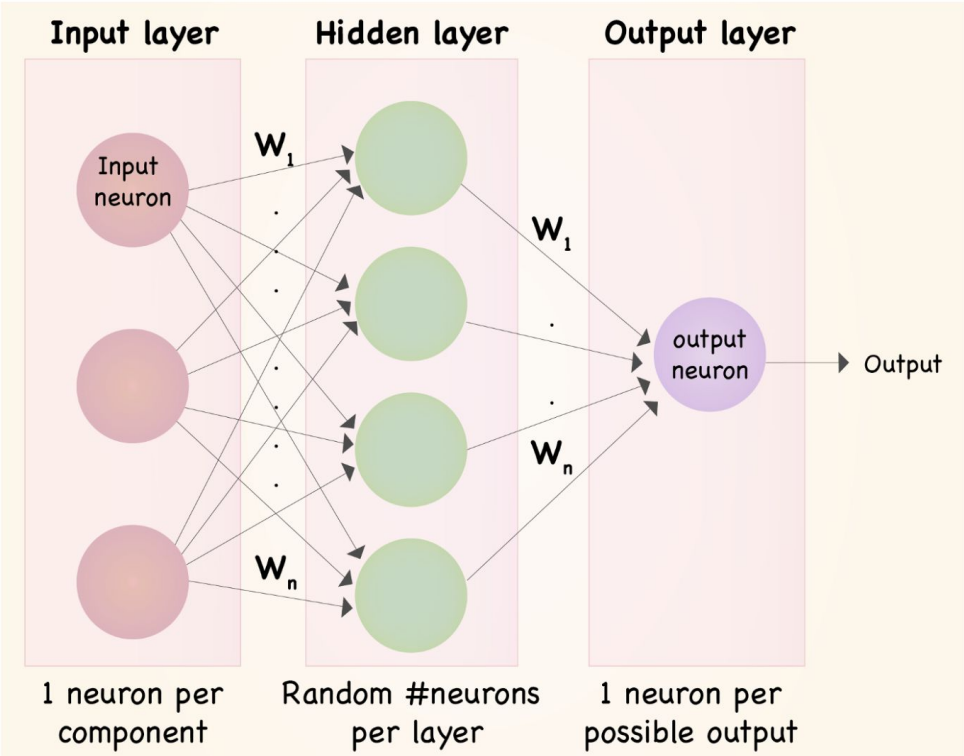


Cada neurona tiene sus propios pesos/parámetros.

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

FUNDACIÓN YPF

Perceptrón Multicapa

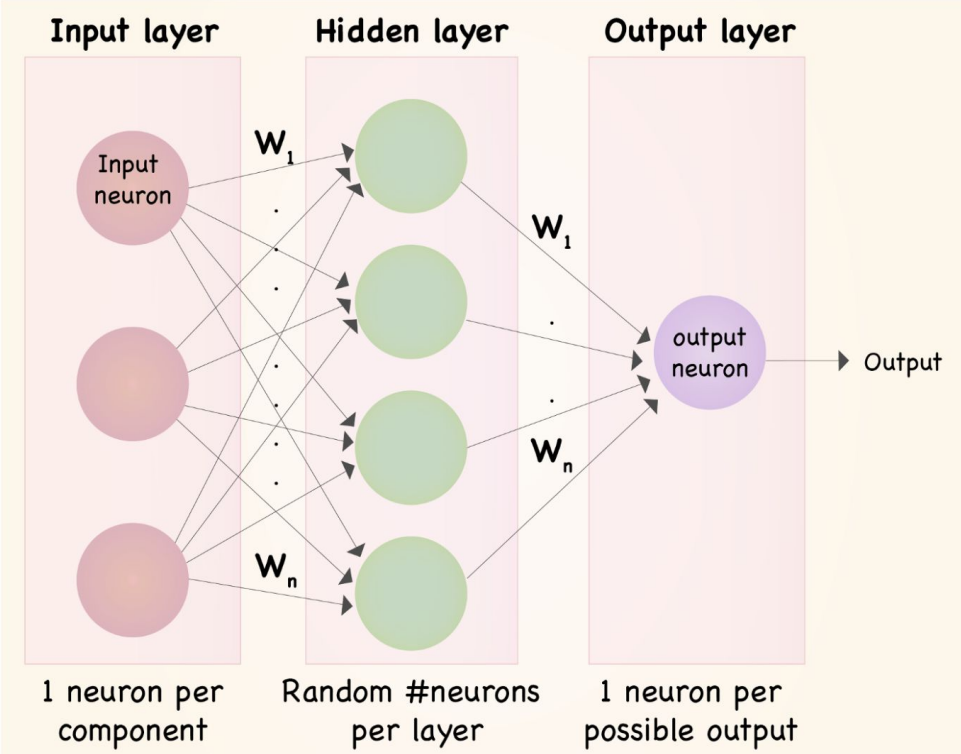


En las neuronas del perceptrón multicapa o red neuronal, se aplican los conceptos vistos:

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

FUNDACIÓN YPF

Perceptrón Multicapa



Tenemos más opciones para las funciones de activación:

- Sigmoidea
- Tanh
- ReLu
- Leaky ReLu

FUNDACIÓN YPF

Funciones de activación

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.

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

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

Tangente hiperbólica (Tanh)

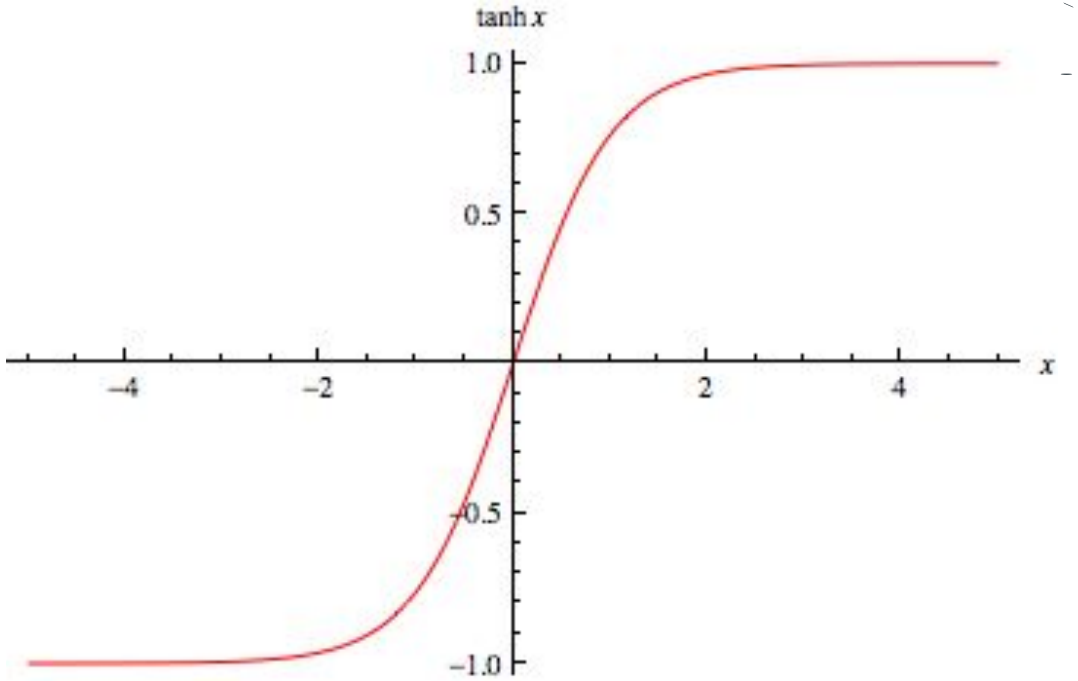
$$f(x) = \frac{2}{1 + e^{-2x}} - 1$$

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

FUNDACIÓN YPF

Tangente hiperbólica (Tanh)

$$f(x) = \frac{2}{1 + e^{-2x}} - 1$$



FUNDACIÓN YPF

Tangente hiperbólica (Tanh)

Características:

- Similar a la función sigmoide
- Sufre de *vanishing gradient*
- Lenta convergencia.
- Centrada en 0.
- Está acotada entre -1 y 1.
- Se utiliza para decidir entre una opción y la contraria.
- Tiene un buen rendimiento para **redes recurrentes.**

FUNDACIÓN YPF

Rectified Lineal Unit (ReLU)

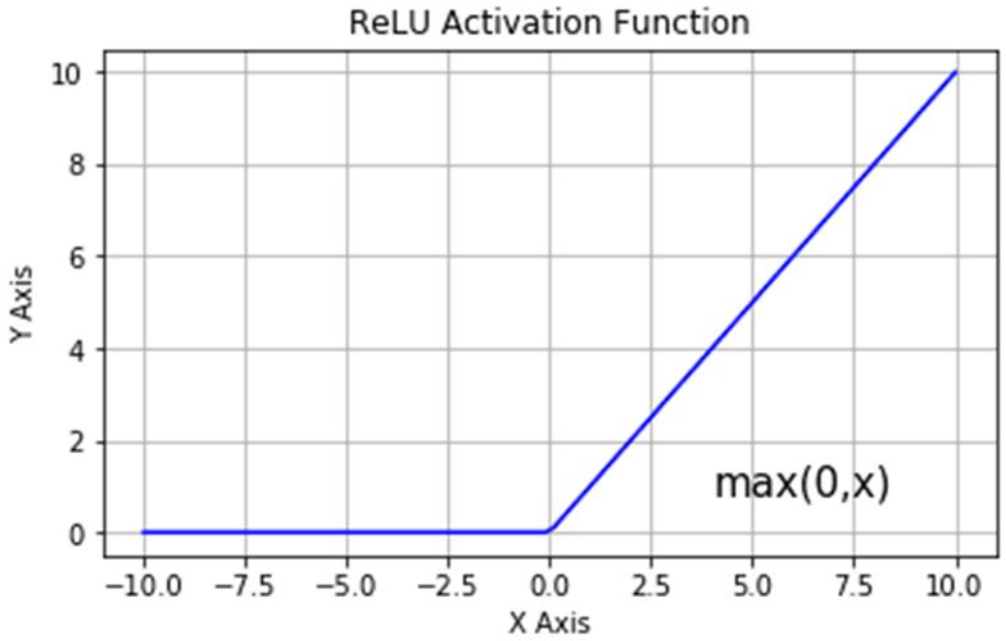
$$f(x) = \max(0, x) = \begin{cases} 0 \; for \; x < 0 \\ x \; for \; x \geq 0 \end{cases}$$

Transforma los valores introducidos anulando los valores negativos y dejando los positivos tal y como entran.

FUNDACIÓN YPF

Rectified Lineal Unit (ReLU)

$$f(x) = \max(0, x) = \begin{cases} 0 \; for \; x < 0 \\ x \; for \; x \geq 0 \end{cases}$$



FUNDACIÓN YPF

Rectified Lineal Unit (ReLU)

Características:

- Activación Sparse: solo se activa si son positivos.
- No está acotada.
- Si bien mejora el *vanishing gradient*, pueden desactivarse demasiadas neuronas
- Tiene un buen rendimiento para **imágenes y redes convolucionales.**

FUNDACIÓN YPF

Rectified Lineal Unit (ReLU)

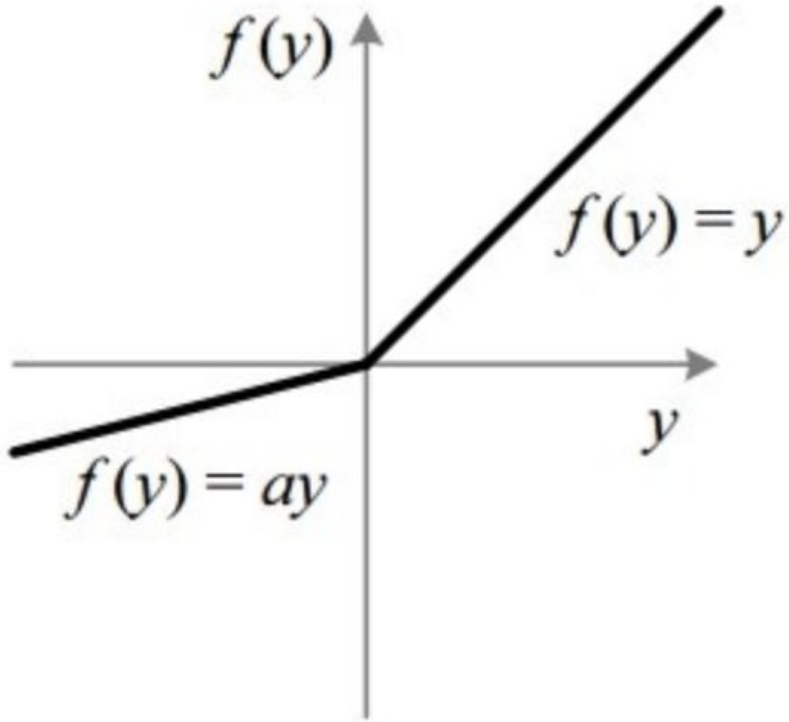
$$f(x) = \begin{cases} 0 & for\ x < 0 \\ a \cdot x & for\ x \geq 0 \end{cases}$$

Transforma los valores introducidos multiplicando los negativos por un coeficiente rectificativo y dejando los positivos según entran.

FUNDACIÓN YPF

Tangente hiperbólica (ReLU)

$$f(x) = \begin{cases} 0 & for\ x < 0 \\ a \cdot x\ for\ x \geq 0 \end{cases}$$



FUNDACIÓN YPF

Tangente hiperbólica (ReLU)

Características:

- Similar a la función ReLU.
- Penaliza los negativos mediante un coeficiente rectificador.
- No está acotada.
- Soluciona el problema de vanishing gradient
- Tiene un buen rendimiento para **imágenes y redes convolucionales.**

FUNDACIÓN YPF

Descanso

Nos vemos en 10 minutos

FUNDACIÓN YPF

Regularización

FUNDACIÓN YPF

Regularización

El objetivo de la regularización es **penalizar** parámetros/pesos muy grandes, ya que estos están asociados al overfitting.

- Regularización L2 y L1: agregan un término a la función de costo que penaliza los pesos grandes.
- **Dropout:** funciona como una capa que “apaga” neuronas de la capa anterior al azar.

FUNDACIÓN YPF

Drop Out

Funciona como una capa que “apaga” neuronas de la capa anterior al azar.

- Es una técnica muy utilizada.
- Al apagar neuronas, obliga a que ninguna se aprenda “de memoria” una muestra, sino que tengan que aprender entre todas.

FUNDACIÓN YPF

Repasamos en Kahoot

FUNDACIÓN YPF

¿DUDAS?

FUNDACIÓN YPF

¡Muchas gracias!