



PUCP

PROCESAMIENTO DE SEÑALES E IMÁGENES DIGITALES

IEE239

INGENIERÍA MECATRÓNICA

Facultad de Ciencias e Ingeniería



REDES NEURONALES CONVOLUCIONALES

- Proceso de Visión Humano
 - Aprendemos a reconocer objetos.
- Neurociencia:
 - El primer uso fue para extraer números en cheques bancarios.



Tomado de: <https://www.acws.cl/las-5-necesidades-de-los-perros/>

REDES NEURONALES

- Sistema que simula la toma de decisiones de una persona y que sirve para:
 - Reconocimiento de caracteres
 - Selección y clasificación de objetos
 - Conducción autónoma
 - Entre otras.
- El principio de las redes neuronales es la interacción de muchos datos (entradas) que trabajan de manera conjunta para la toma de una decisión (salida).

REDES NEURONALES

- Una red neuronal es una combinación de elementos de **entrada**, con pesos específicos, que se suman y cuyo resultado (**salida**) pasa por una **función de activación**.

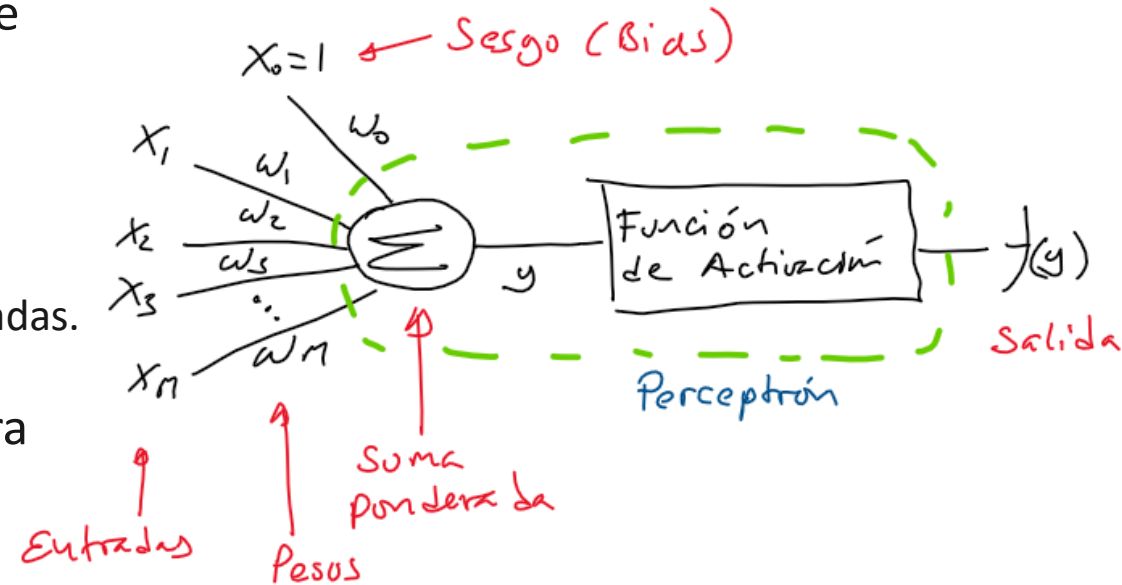
$$y = \sum_{i=1}^m x_i w_i + x_0$$

- x_0 es el sesgo (*bias*) o tendencia y m es el número de entradas.
- La **función de activación** se aplica para que la neurona aprenda patrones complejos de datos, pudiendo separa categorías.

- Ejemplo:

$$f(y) = \begin{cases} 1 & y \geq 0 \\ 0 & y < 0 \end{cases}$$

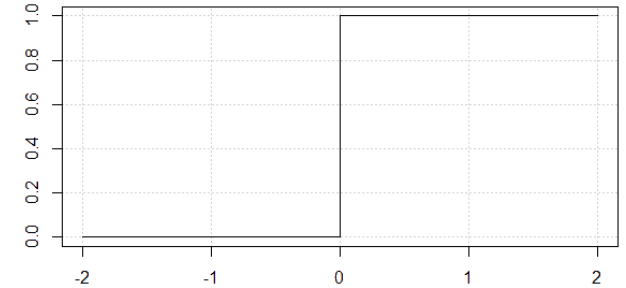
- Para esta función de activación, la **salida** tendrá un valor verdadero o falso.



FUNCIONES DE ACTIVACIÓN

- **Escalón:** Esta función produce dos valores únicos en la salida, 0 ó 1.

$$f(y) = \begin{cases} 1 & y \geq 0 \\ 0 & y < 0 \end{cases}$$



<https://www.alexisalulema.com/es/2022/09/23/funciones-de-activacion-en-tensorflow/>

- Ejemplo 1: Compuerta NOT
- Ejemplo 2: Compuerta AND
- Ejemplo 3: Compuerta XOR

FUNCIONES DE ACTIVACIÓN

- **Sigmoide:** Función continua, con cambio graduales entre 0 y 1.

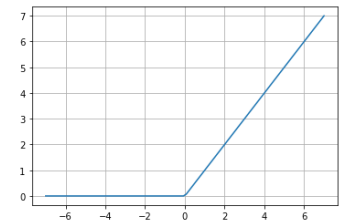
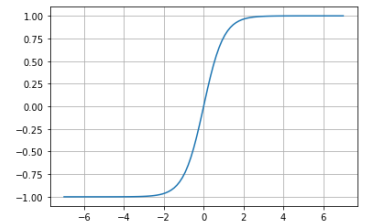
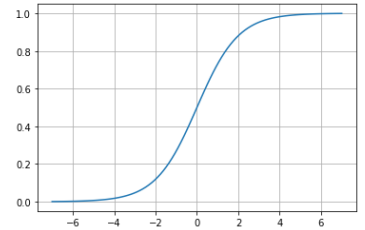
$$f(y) = \frac{1}{1 + e^{-y}}$$

- **Tangente Hiperbólica:** Función similar a la sigmoide, pero con el rango de salida entre -1 y 1.

$$f(y) = \tanh(y) = \frac{e^y - e^{-y}}{e^y + e^{-y}}$$

- **ReLU (*Rectified Linear Unit*):** Función muy popular. Esta función de activación acelera la convergencia de los gradientes descendentes.

$$f(y) = \max(0, y)$$



<https://www.alexisalulema.com/es/2022/09/23/funciones-de-activacion-en-tensorflow/>

REDES NEURONALES

- Un problema es la asignación de los pesos a la red neuronal para que cumpla con las especificaciones dadas.
 - A la búsqueda de los pesos correctos se le llama aprendizaje.
- Una forma de aprender es la búsqueda del error.
 - Se denomina error a la diferencia entre la salida obtenida (O) y la deseada.

$$E = f(y) - O$$

- Para la búsqueda de los pesos se aplica la regla de aprendizaje

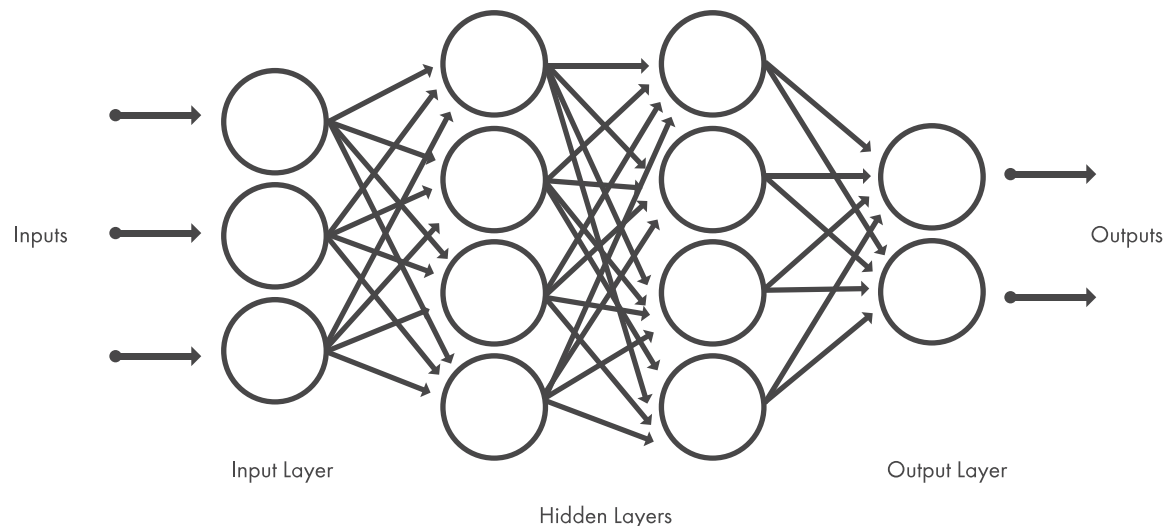
$$w_i \leftarrow w_i + \alpha E x_i$$

- Donde el Factor de Aprendizaje, α , tiene valores entre $0 < \alpha < 1$

x_1	x_2	$f(y)$	O
0	0	0	
0	1	1	
1	0	1	
1	1	1	

REDES NEURONALES

- Una red neuronal combina diferentes capas interconectadas.
- Dentro de una red neuronal se realizan sumas, multiplicaciones y transformaciones no lineales.
- Al desarrollar una red neuronal, se tiene que establecer cuántas capas y neuronas se deben de tener para que la red pueda clasificar correctamente.



<https://la.mathworks.com/discovery/neural-network.html>

<https://playground.tensorflow.org/>

RETROPROPAGACIÓN

(BACKPROPAGATION)

- Retropropagación es uno de los métodos para encontrar los pesos que se ajusten mejor a la red y así pueda tomar una decisión.
- Este algoritmo usa el error cuadrático medio, el cual tiene como objetivo minimizar su valor.

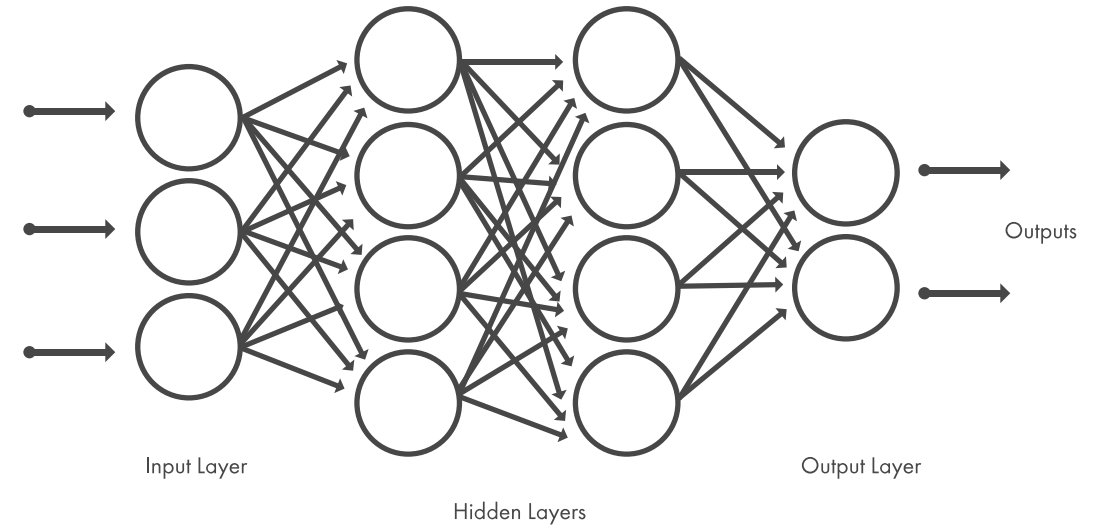
$$E = \frac{1}{2} \sum_{k=1}^n (f(y_k) - o_k)^2$$

- Donde n es el número de neuronas de salida.
- La red necesita un conjunto de datos (entradas y salidas) que serán utilizados para que la red aprenda.
 - A este conjunto de datos se le denomina “conjunto de entrenamiento”

RETROPROPAGACIÓN

(BACKPROPAGATION)

- El entrenamiento se inicia propagando una entrada y obteniendo una salida (salida obtenida) con los pesos iniciales.
- Se halla el error y luego se corrigen los pesos aplicando la propagación hacia atrás (retropropagación).
- Con los pesos corregidos, se hace uso de la siguiente entrada y salida para continuar corrigiendo los pesos.
- El error irá disminuyendo en cada iteración.
- Para corregir los pesos, se hace uso del descenso de la gradiente.



<https://la.mathworks.com/discovery/neural-network.html>

DESCENSO DE LA GRADIENTE



$$w \leftarrow w + \alpha \nabla E(w_i)$$

- Donde α es el factor de aprendizaje y $\nabla E(w)$ es la gradiente de la función del error
- La función de activación debe ser derivable. Con esto, para aplicar la retropropagación, la función de activación escalón no sirve.
 - Se utiliza la función de activación sigmoide. Esto porque la función sigmoide es derivable.

RETROPROPAGACIÓN

(BACKPROPAGATION)

- Tenemos la red:

$$y = x_0 w_0 + x_1 w_1 + x_2 w_2$$

- Para la salida:

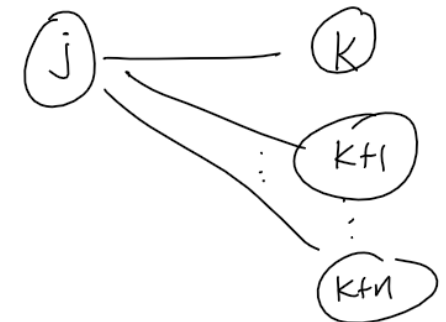
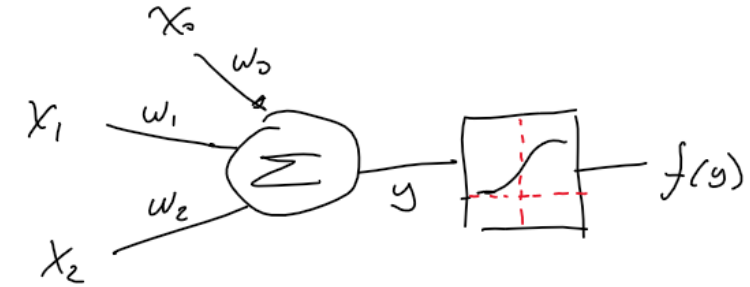
$$f(y) = \frac{1}{1 + e^{-y}} \rightarrow f'(y) = f(y)(1 - f(y))$$

- La gradiente del error ($E = \frac{1}{2} (f(y_k) - O_k)^2$)

$$\frac{\partial E}{\partial w_i} = x_i O_k (1 - O_k) (f(y_k) - O_k) = \Delta_k x_i = \nabla E(w_i)$$

- Error de la capa oculta:

$$\Delta_j = O_j (1 - O_j) \sum_{k=1}^n w_{jk} \Delta_k$$



RETROPROPAGACIÓN

(BACKPROPAGATION)

- Para la siguiente red neuronal:

- Tenemos para la capa de salida:

$$\Delta_5 = O_5(1 - O_5)(f(y_5) - O_5)$$

- $w_{50} \leftarrow w_{50} + \alpha \Delta_5(1)$

- $w_{35} \leftarrow w_{35} + \alpha \Delta_5 O_3$

- $w_{45} \leftarrow w_{45} + \alpha \Delta_5 O_4$

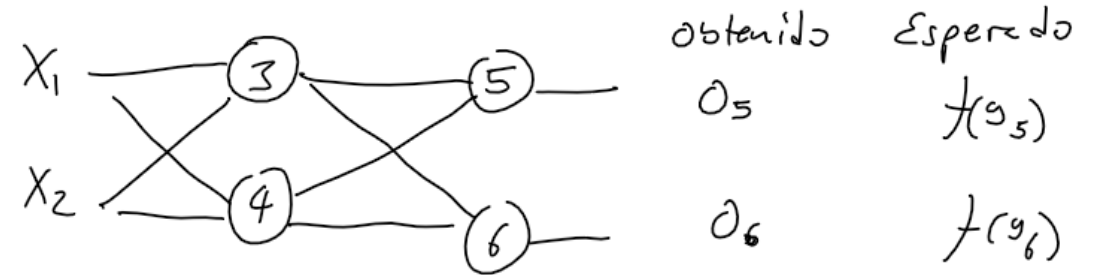
- Para la capa oculta:

$$\Delta_3 = O_3(1 - O_3)(w_{35}\Delta_5 + w_{36}\Delta_6)$$

- $w_{30} \leftarrow w_{30} + \alpha \Delta_3(1)$

- $w_{13} \leftarrow w_{13} + \alpha \Delta_3 x_1$

- $w_{23} \leftarrow w_{23} + \alpha \Delta_3 x_2$

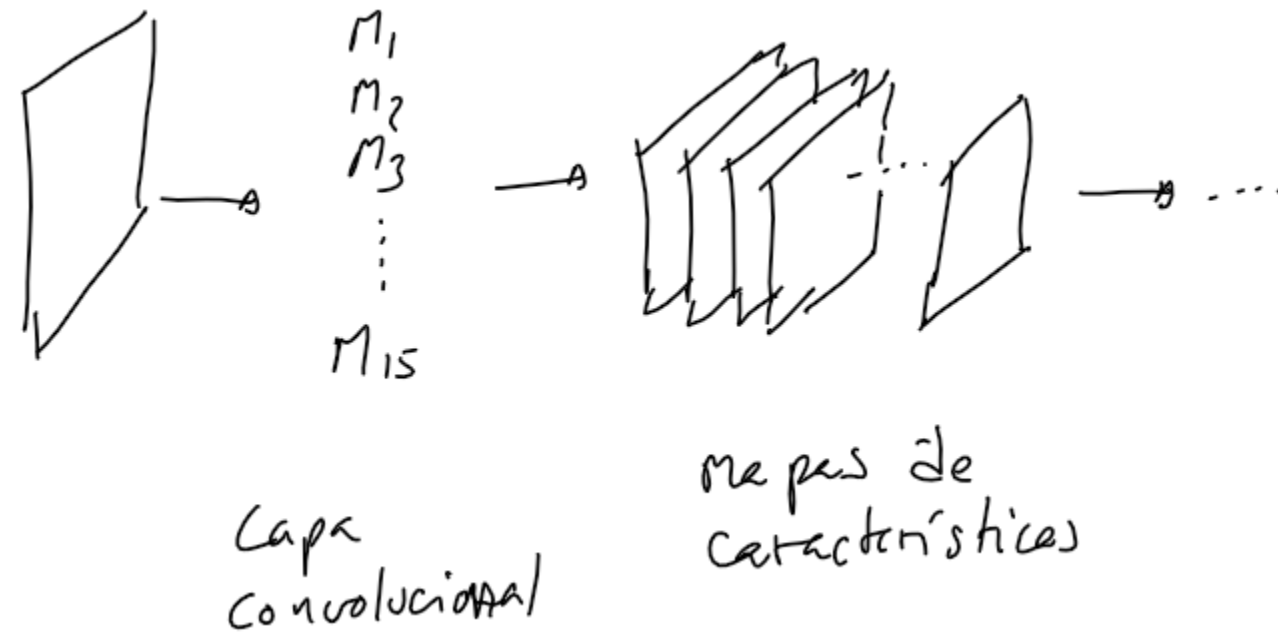


RED NEURONAL CONVOLUCIONAL

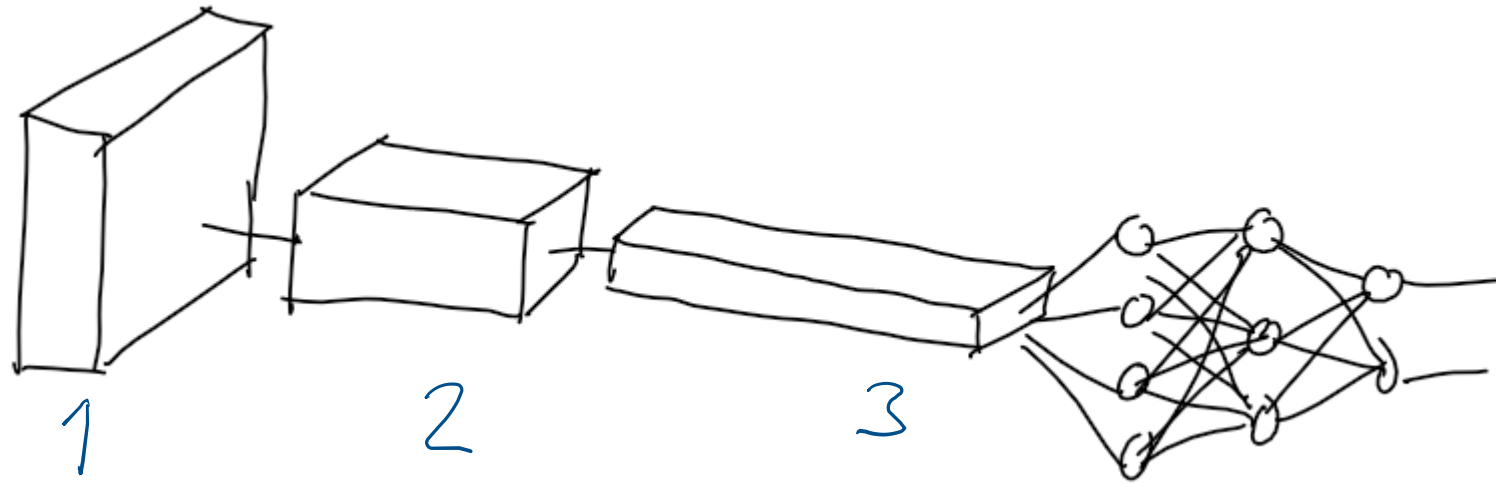
- En la red neuronal convolucional se establece que el sistema le dará el valor al kernel de convolución, que irá aprendiendo por sí mismo.
- Aprender estos filtros para aprender a detectar patrones. Cada imagen generada se le llama **mapa de características**.
- Estos mapas de características pasará a ser entradas de otras capas.

RED NEURONAL CONVOLUCIONAL

- Cada paso por la capa convolucional pasa a procesar mayor área de información de imagen.
 - El sistema hace detecciones sobre detecciones por capas anteriores.
- Se va reduciendo las imágenes cada vez que se vaya pasando a la siguiente capa convolucional.



RED NEURONAL CONVOLUCIONAL



1. El tamaño de la imagen se va reduciendo.
2. El número de mapas característicos va aumentando.
3. Se llega a un punto donde se han detectado todos los patrones necesario para pasar a una red neuronal.