

Inteligencia Artificial

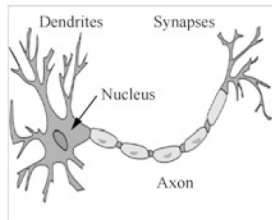
Instituto Nacional de Astrofísica, Óptica y Electrónica

Presents:

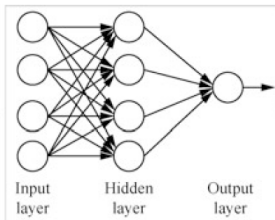
MSc. Mireya Lucia Hernández Jaimes

① Deep learning

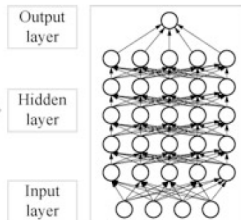
Deep Learning se basa en redes neuronales, llamados normalmente como redes neuronales multi capa (multi-layer NN). Su objetivo es simular la red neuronal del cerebro humano.



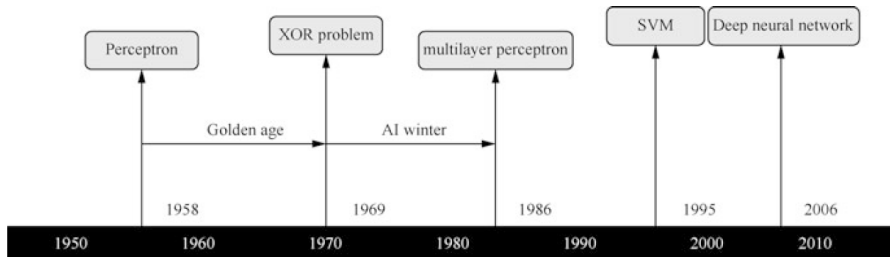
(a) Neuron in the Human Brain



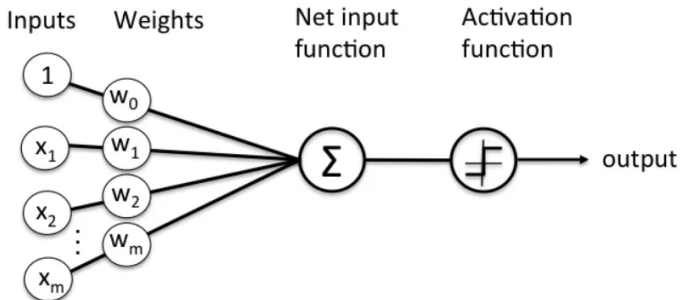
(b) Perceptron



(c) Deep neural network



Es la red neuronal más simple.

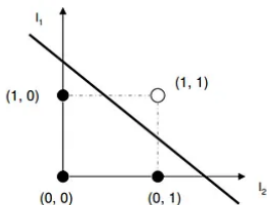


- Neuronas artificiales: Perceptrones
- Perceptrones: Entrada, pesos y la salida.
- Suma todas las características con su respectivo peso asociado.
- Función de activación: es la función matemática que le ayuda a comprender patrones complejos.

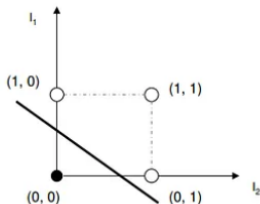
XOR problem

La función XOR no es una función que pueda ser línealmente separable por un solo hyperplano.

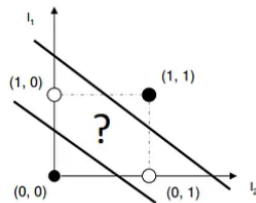
AND		
I_1	I_2	out
0	0	0
0	1	0
1	0	0
1	1	1



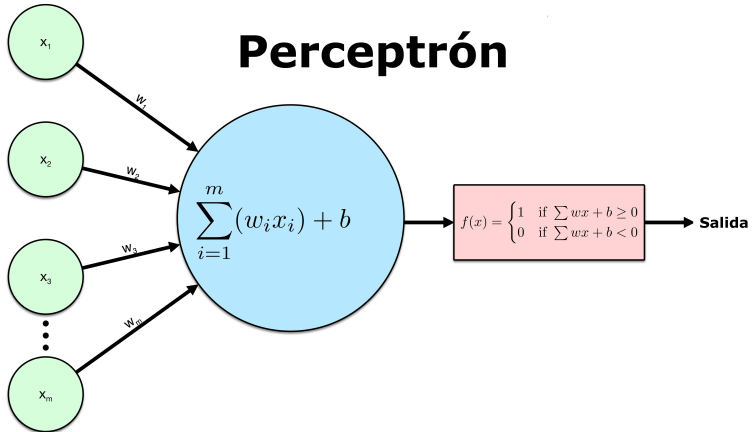
OR		
I_1	I_2	out
0	0	0
0	1	1
1	0	1
1	1	1



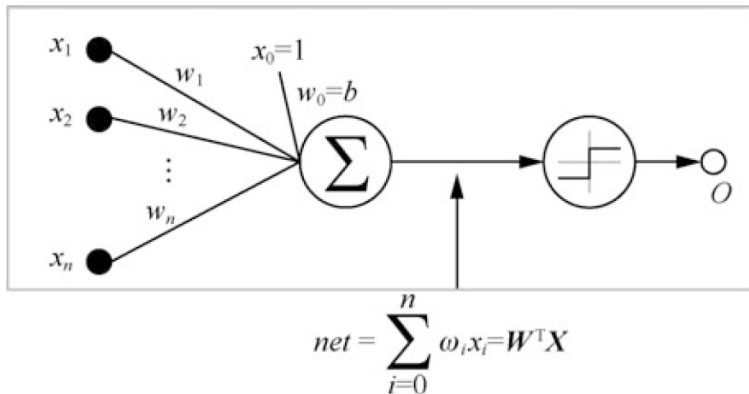
XOR		
I_1	I_2	out
0	0	0
0	1	1
1	0	1
1	1	0



Utilizando la función de activación umbral.



Utilizando la función de activación tipo escalón.

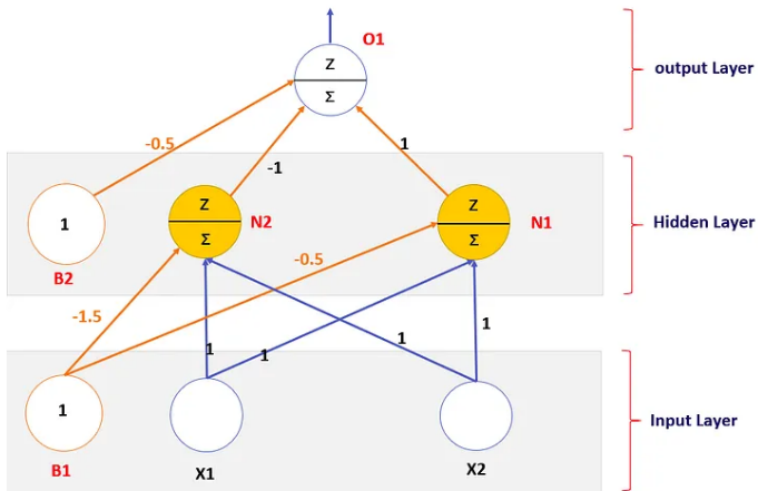


Asume el peso $W_1 = w_2 = 1$ y $b = -1.5$ para realizar las siguientes operaciones con el perceptrón de una sola capa utilizando la función de activación tipo escalón:

- AND
- OR
- XOR

Multi-layer perceptron

Resuelve la operación XOR con el siguiente perceptrón con una capa profunda:



Funciones de pérdida (Loss function)

Mientras se entrena una red neuronal profunda, se necesita utilizar una función de pérdida (o función de error). La métrica de error más común es la MSE.

$$J(\mathbf{w}) = \frac{1}{2n} \sum_{x \in X, d \in D} (t_d - o_d)^2$$

- n = tamaño de X
- D = neuronas en la capa de salida
- t = el valor real o esperado
- o = el valor obtenido por la red neuronal

El algoritmo del descenso del gradiente (gradient descent) se utiliza para permitir que la función de pérdida busque en la dirección del gradiente negativo, actualice iterativamente los parámetros y minimice la función de pérdida.

El descenso del gradiente de una función mutivariable $f(x_1, x_2, \dots, x_n)$ toma la siguiente forma:

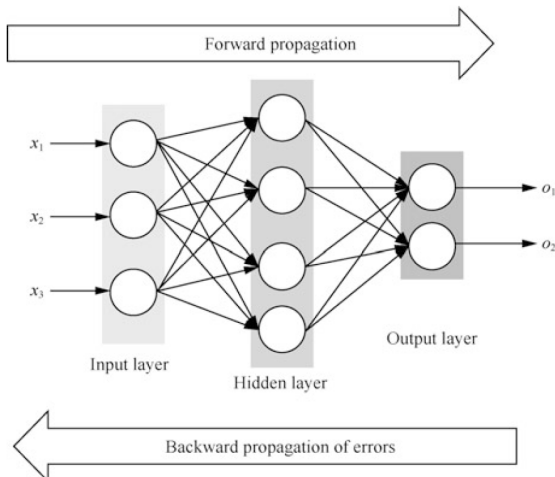
$$\nabla f(x_1, x_2, \dots, x_n) = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right]^T$$

El proceso de aplicar el algoritmo descenso del gradiente en la función de pérdida, para algoritmos de ML como Linear Regression, es más simple que para las redes neuronales. Es imposible encontrar los gradientes de una función de pérdida de todos los parametros que tiene una red neuronal multi capa en una sola fórmula. Como solución tenemos el algoritmo : Backpropagation, el cual nos permite lo siguiente:

- Actualizar los pesos asociados a las neuronas por capas
- Optimizar el tiempo de entrenamiento de los algoritmos.

Backpropagation algorithm

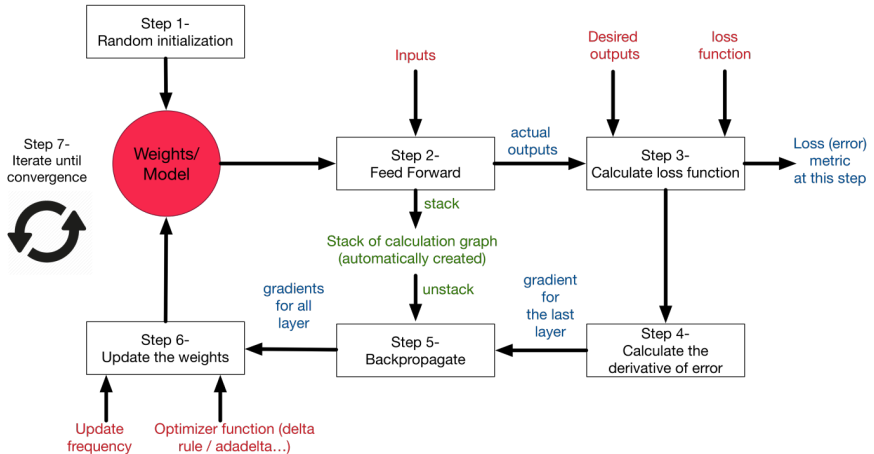
Durante el proceso de propagación hacia adelante, cada capa provocará un determinado error. A medida que estos errores se acumulan capa por capa, eventualmente se manifiestan en forma de una función de pérdida en la capa de salida.



$$J(\mathbf{w}) = \frac{1}{2m} \sum_{x \in X, d \in D} (t_d - f(\mathbf{w}_L o'_d))^2$$

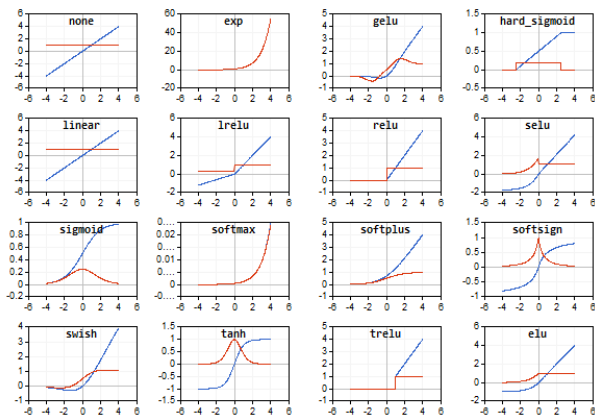
- O' = es la salida de la ultima capa (output layer)
- f = función de activación de la última capa (output layer)
- En esta formula, los valores de salida de la ultima capa (valores de las neuronas de la última capa) son solo relevantes para los pesos de una capa anterior.

Backpropagation algorithm



Funciones de activación

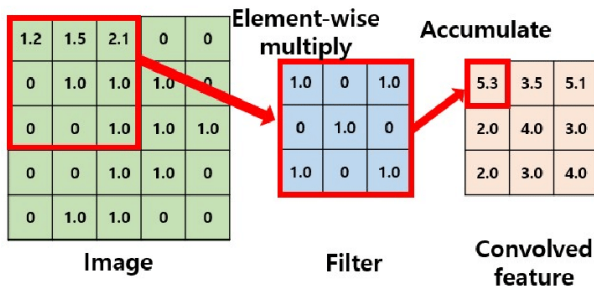
La función de activación juega un papel muy importante en el aprendizaje de modelos de redes neuronales y en la interpretación de funciones no lineales complejas. La función de activación agrega características no lineales a la red neuronal. Sin la función de activación, una red neuronal sólo puede representar una función lineal, sin importar cuántas capas tenga.



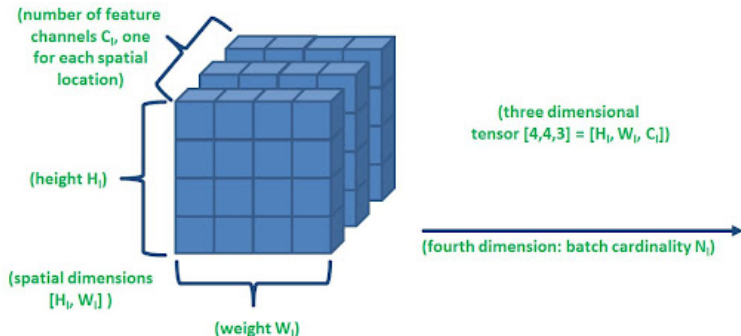
Consta de 3 capas principales:

- Capa de convolución (convolutional layers)
- Capa de agrupación (pooling layers)
- Capa completamente conectadas (fully connected layers)

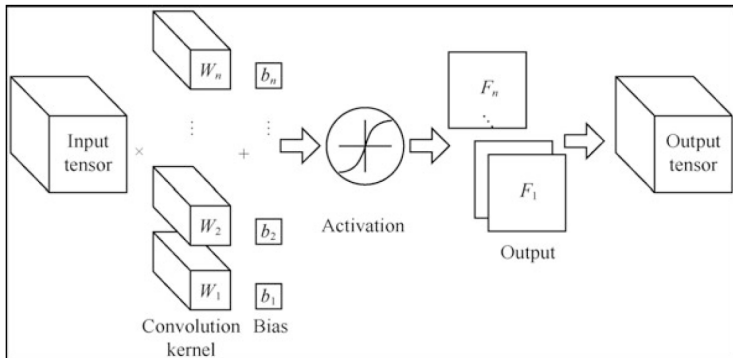
El nombre “red neuronal convolucional” deriva de una operación matemática llamada convolución. La convolución es la operación de realizar el producto interno para imagen (o mapa de características) y matriz de filtro (también llamada filtro y núcleo de convolución).



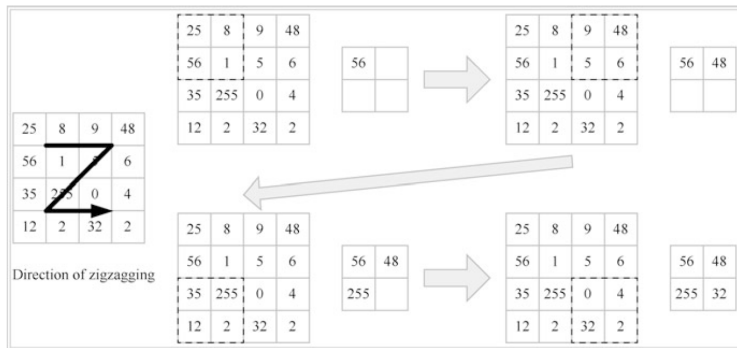
Una forma de representar los datos de entrada.



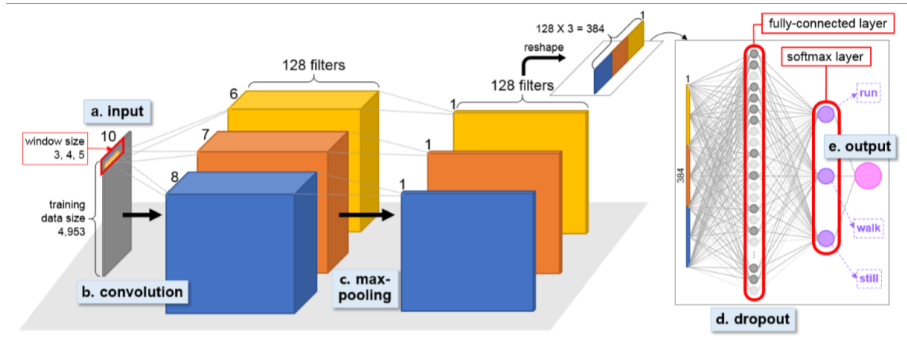
Capa de convolución: es mayormente estructurada como la convolución multi canal.

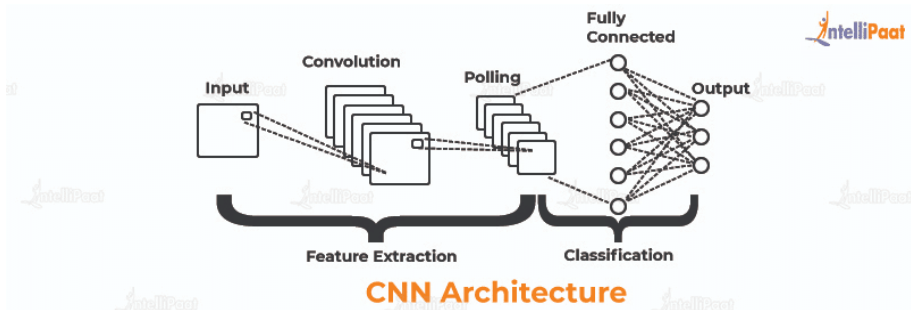


Capa de agrupación: nos sirve para reducir la dimensionalidad del espacio de características. Las estrategias más comunes es la capa de agrupación máxima y la capa de agrupación promedio.



Fully connected layer





RNN es un tipo de red neuronal que captura información dinámica en datos serializados conectando periódicamente nodos en la capa oculta, para clasificar los datos serializados. A diferencia del resto de redes neuronales feedforward, las redes neuronales recurrentes pueden mantener el estado de contexto de los datos serializados.