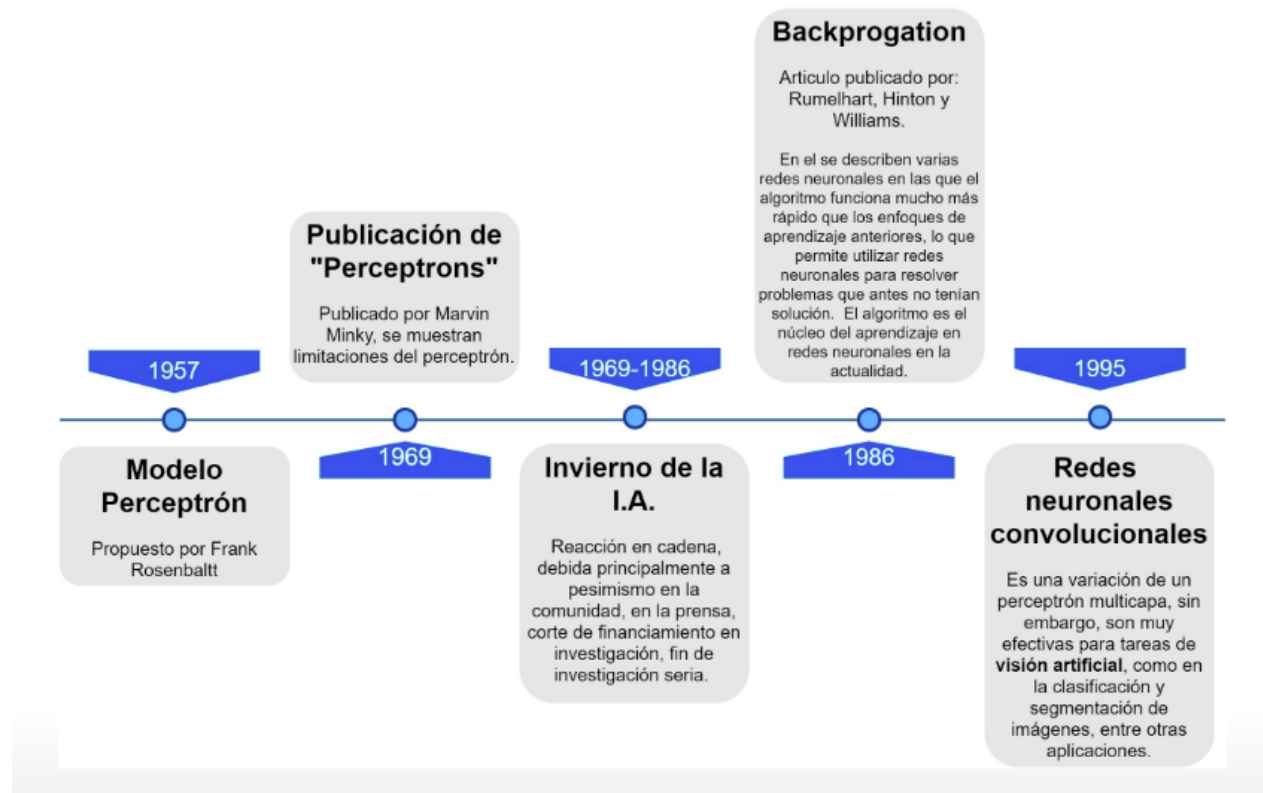


Entendiendo el perceptrón simple

Historia



Época actual

- 1- **Big Data:** Bases de datos más grandes, mayor capacidad de almacenamiento.
- 2- **Hardware:** GPUs, procesamiento en paralelo.
- 3- **Software:** Mejoras en técnicas, nuevos modelos, *Toolboxes*.



Las neuronas son simples procesadores de información, consisten en un cuerpo celular y cables que los conectan entre si.

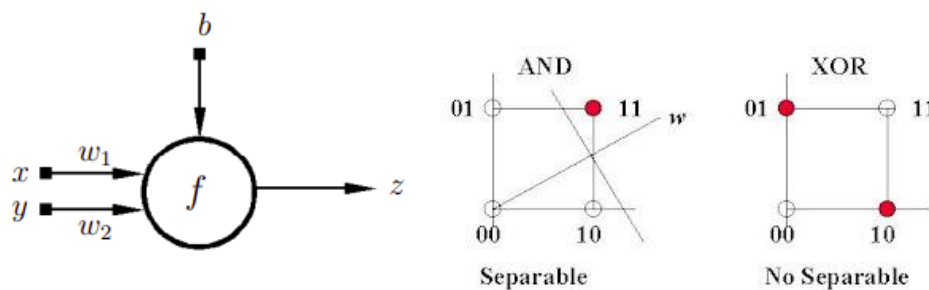
Partes de una neurona:

- Cuerpo celular (A)
- Dendritas (B)
- Axón (C)
- Sinapsis (D)



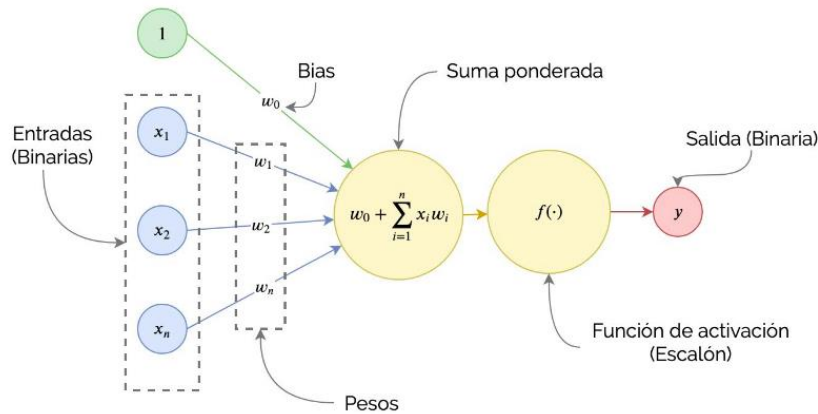
Perceptrón

El perceptrón es la forma más simple de una red neuronal utilizada para la clasificación de patrones que se dice que son linealmente separables (es decir, patrones que se encuentran en lados opuestos de un hiperplano). Básicamente, consiste en una sola neurona con pesos y sesgos sinápticos ajustables.



El algoritmo utilizado para ajustar los parámetros libres de esta red neuronal apareció por primera vez en un procedimiento de aprendizaje desarrollado por Rosenblatt (1958, 1962) para su modelo de cerebro perceptrón. Rosenblatt demostró que si los patrones (vectores) utilizados para entrenar el perceptrón se extraen de dos clases linealmente separables, entonces el algoritmo del perceptrón converge y posiciona la superficie de decisión en forma de hiperplano entre las dos clases. La prueba de convergencia del algoritmo se conoce como el **teorema de convergencia del perceptrón**.

El perceptrón construido alrededor de una sola neurona se limita a realizar la clasificación de patrones con solo dos clases (hipótesis). Al expandir la capa de salida (computación) del perceptrón para incluir más de una neurona, podemos realizar una clasificación correspondiente con más de dos clases. Sin embargo, las clases deben ser linealmente separables para que el perceptrón funcione correctamente.



El primer modelo matemático de una neurona artificial, creado con el fin de llevar a cabo tareas simples, fue presentado en el año 1943 en un trabajo conjunto entre el psiquiatra y neuroanatomista Warren McCulloch y el matemático Walter Pitts.

Sus componentes son:

- Las entradas x_n
- Los pesos sinápticos w_n correspondientes a cada entrada
- Un término aditivo w_0
- Una función de activación f
- Una salida y

Las entradas son el estímulo que la neurona artificial recibe del entorno que la rodea, y la salida es la respuesta a tal estímulo. La neurona se adapta al medio circundante y aprende modificando el valor de sus pesos sinápticos w_n y su término aditivo w_0 . Estos son conocidos como los parámetros libres del modelo, pues los mismos **pueden ser modificados y adaptados para realizar una tarea determinada**.

$$a = f\left(\sum_{i=0}^N w_i x_i\right)$$

La función de activación f es seleccionada de acuerdo a la tarea realizada por la neurona. Para calcular la salida, multiplicaremos la entrada con los pesos respectivos y la compararemos con un valor de umbral. La suma ponderada

de las entradas se pasa a través de una función de

$$f(s) = \begin{cases} 1, & \text{si } s \geq 0 \\ -1, & \text{si } s < 0 \end{cases}$$

paso/activación.

consideraremos como función de activación a la función signo definida por:

$$z = \begin{cases} 1, & \text{si } w_1 x + w_2 y + b \geq 0 \\ -1, & \text{si } w_1 x + w_2 y + b < 0 \end{cases}$$

Algoritmo del Perceptrón

El objetivo es encontrar el vector \mathbf{w} que pueda clasificar perfectamente las entradas positivas y negativas en nuestros datos

Inicializamos \mathbf{w} con algún vector aleatorio. Luego iteramos sobre todos los ejemplos en los datos, $(P \cup N)$ tanto ejemplos positivos como negativos.

```
P <-- entradas con etiqueta 1;  
N <-- entradas con etiqueta 0;
```

Ignacio David Vázquez Perez
218292866

```
Inicializar w aleatoriamente;  
mientras !converge haz  
    Elige aleatoriamente  $x \in P \cup N$ ;  
    si  $x \in P$  y  $w \cdot x < 0$  entonces  
         $w = w + x$ ;  
    termina  
    si  $x \in N$  y  $w \cdot x \geq 0$  entonces  
         $w = w - x$ ;  
    termina  
termina
```

Un solo perceptrón solo puede usarse para implementar **funciones separables linealmente** . Toma entradas reales y booleanas y les asocia un conjunto de **pesos** , junto con un **sesgo**. El algoritmo tampoco cuenta con un factor de aprendizaje o un metodo de optimización.

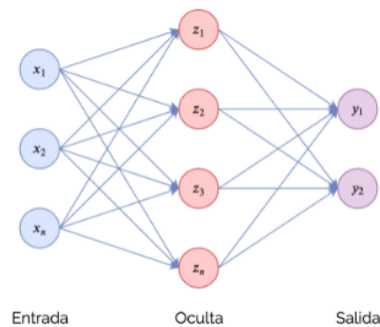
Perceptrón Multicapa

Una red neuronal, tanto la biológica como la artificial, consiste en un largo número de unidades simples llamadas **neuronas**, que reciben y transmiten señales unas con otras.

Capas

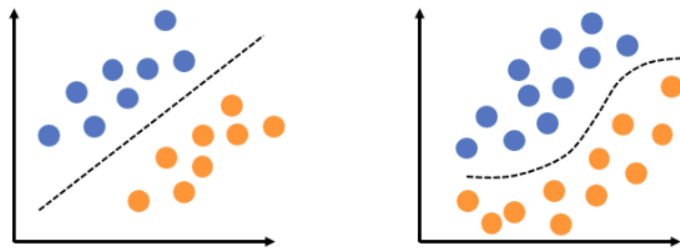
A menudo las arquitecturas de redes neuronales están compuestas por capas de neuronas que se comunican entre sí, principalmente las podemos dividir en:

- Capa de entrada
- Capas ocultas
- Capa de salida



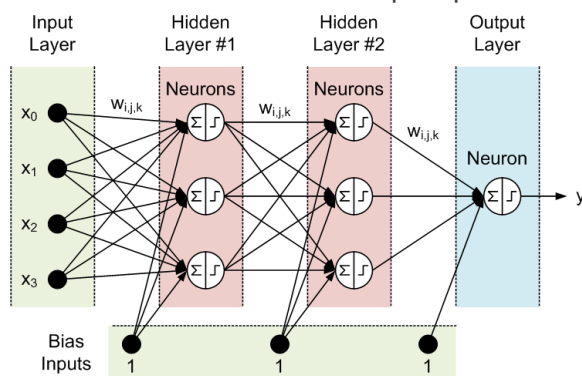
Perceptrón multicapa

- Es una red neuronal artificial (RNA) formada por al menos tres capas de nodos: una capa de entrada, una capa oculta y una capa de salida.
- Tiene capacidad para resolver problemas que no son linealmente separables, lo cual es la principal limitación del perceptrón simple.
- Esta red se ha convertido en una de las arquitecturas más utilizadas en la actualidad.



Para superar las limitaciones prácticas del perceptrón buscamos una estructura de red neuronal conocida como perceptrón multicapa. Los siguientes tres puntos destacan sus características:

- El modelo de cada neurona de la red incluye una función de activación no lineal que es diferenciable.
- La red contiene una o más capas que están ocultas para los nodos de entrada y salida.



- La red exhibe un alto grado de conectividad, cuyo alcance está determinado por los pesos sinápticos de la red.

El uso de neuronas ocultas hace que el proceso de aprendizaje sea más difícil de visualizar. En un sentido implícito, el proceso de aprendizaje debe decidir qué características del patrón de entrada deben representar las neuronas ocultas. Por lo tanto, el proceso de aprendizaje se hace

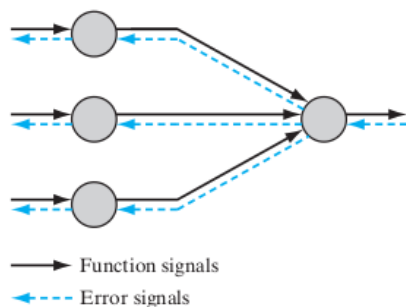
más difícil porque la búsqueda debe realizarse en un espacio mucho más grande de funciones posibles y debe elegirse entre representaciones alternativas del patrón de entrada.

La formación se desarrolla en dos fases:

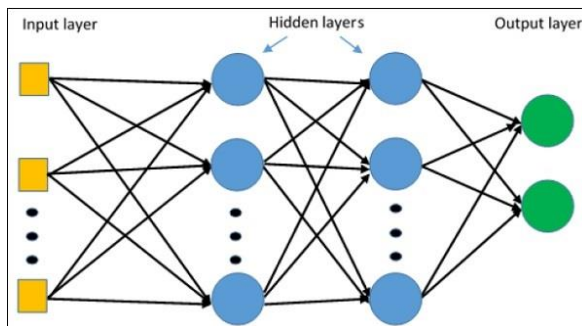
1. En la fase directa, los pesos sinápticos de la red se fijan y la señal de entrada se propaga a través de la red, capa por capa, hasta llegar a la salida. Así, en esta fase, los cambios se limitan a los potenciales de activación y las salidas de las neuronas de la red.

2. En la fase inversa, se produce una señal de error al comparar la salida de la red con una respuesta deseada. La señal de error resultante se propaga a través de la red, nuevamente capa por capa, pero esta vez la propagación se realiza en dirección hacia atrás. En esta segunda fase se realizan sucesivos ajustes a los pesos sinápticos de la red.

La red que se muestra aquí está completamente conectada. Esto significa que una neurona en cualquier capa de la red está conectada a todas las neuronas (nodos) en la capa anterior. El flujo de señales a través de la red progresa hacia adelante, de izquierda a derecha y capa por capa.



En esta red se identifican dos tipos de señales: **Señales de función**. Una señal de función es una señal de entrada (estímulo) que entra en el extremo de entrada de la red, se propaga (neurona por neurona) a través de la red y emerge en el extremo de salida de la red como una señal de salida. **Señales de error**. Una señal de error se origina en una neurona de salida de la red y se propaga hacia atrás (capa por capa) a través de la red.

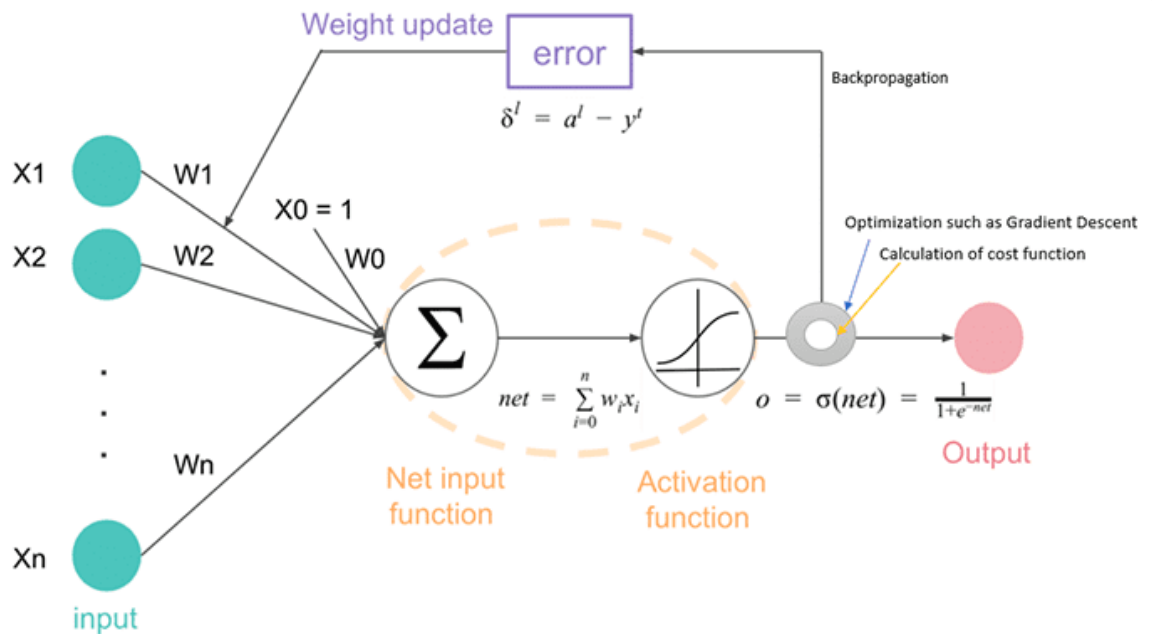


Las neuronas de salida constituyen la capa de salida de la red. Por lo tanto, las unidades ocultas no son parte de la salida o entrada de la red, de ahí su designación como "ocultas". La primera capa oculta se alimenta de la capa de entrada formada por unidades sensoriales (nodos fuente); las salidas resultantes de la primera capa oculta se aplican a su vez a la siguiente capa oculta; y así sucesivamente para

el resto de la red.

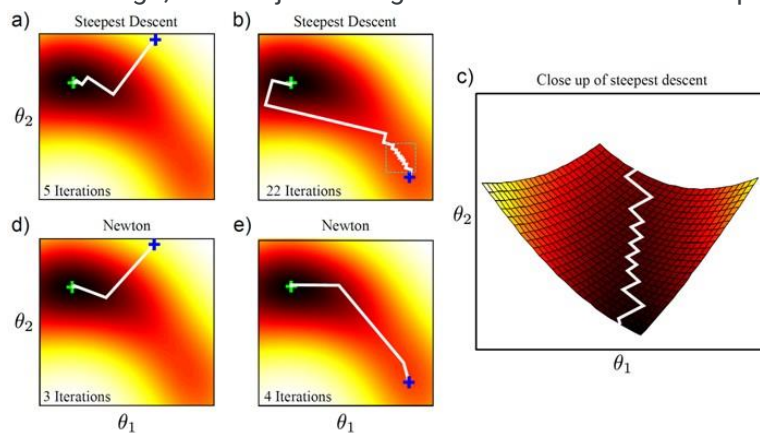
Cada neurona oculta o de salida de un perceptrón multicapa está diseñada para realizar dos cálculos:

1. el cálculo de la señal de función que aparece en la salida de cada neurona, que se expresa como una función no lineal continua de la señal de entrada y pesos sinápticos asociados con esa neurona;
2. el cálculo de una estimación del vector gradiente (es decir, los gradientes de la superficie de error con respecto a los pesos conectados a las entradas de una neurona), que se necesita para el paso hacia atrás a través de la red.



El algoritmo de retropropagación proporciona una "aproximación" a la trayectoria en peso espacio calculado por el método del descenso más empinado. Cuanto más pequeño hagamos el parámetro de tasa de aprendizaje, menores serán los cambios en los pesos sinápticos en la red de una iteración a la siguiente, y más suave será la trayectoria en el espacio de pesos.

Sin embargo, esta mejora se logra a costa de un ritmo de aprendizaje más lento.



Regla Delta

Problema de Optimización

$$L(y, \hat{y}) = \frac{1}{2}(y - \hat{y})^2$$

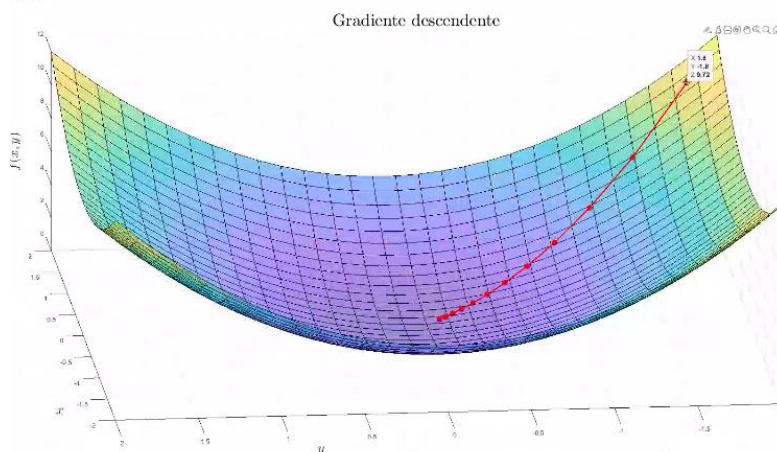
Algoritmo de Optimización

$$w \leftarrow w - \eta \nabla_w L(y, \hat{y})$$
$$b \leftarrow b - \eta \frac{\delta}{\delta b} L(y, \hat{y})$$

Ecuaciones de actualización

$$w \leftarrow w + \eta (y - \hat{y}) \left(\frac{d}{dz} \hat{y} \right) x$$
$$b \leftarrow b + \eta (y - \hat{y}) \left(\frac{d}{dz} \hat{y} \right)$$

Dr. Carlos Villaseñor Universidad de Guadalajara



En general, no se puede demostrar que el algoritmo de propagación hacia atrás converja, y hay o Criterios bien definidos para detener su funcionamiento. Más bien, hay algunos criterios razonables, cada uno con su propio mérito práctico, que pueden usarse para terminar los ajustes de peso. Para formular tal criterio, es lógico pensar en términos de las propiedades únicas de un mínimo local o global de la superficie de error. Después de cada iteración de aprendizaje, se prueba el rendimiento de generalización de la red.

El proceso de aprendizaje se detiene cuando el rendimiento de generalización es adecuado o cuando es evidente que el rendimiento de generalización ha alcanzado su punto máximo.

La propagación hacia atrás es una técnica específica para implementar el descenso de gradiente en el espacio de peso para un perceptrón multicapa. La idea básica es calcular eficientemente las derivadas parciales de una función de aproximación $F(w, x)$ realizada por la red con respecto a todos los elementos del vector de peso ajustable w para un valor dado del vector de entrada x .

Es una técnica computacionalmente eficiente para calcular los gradientes (es decir, derivadas de primer orden) de la función de costo $e(w)$, expresada como una función de los parámetros ajustables (pesos sinápticos y términos de sesgo) que caracterizan al perceptrón multicapa.