

Multilayer Perceptron

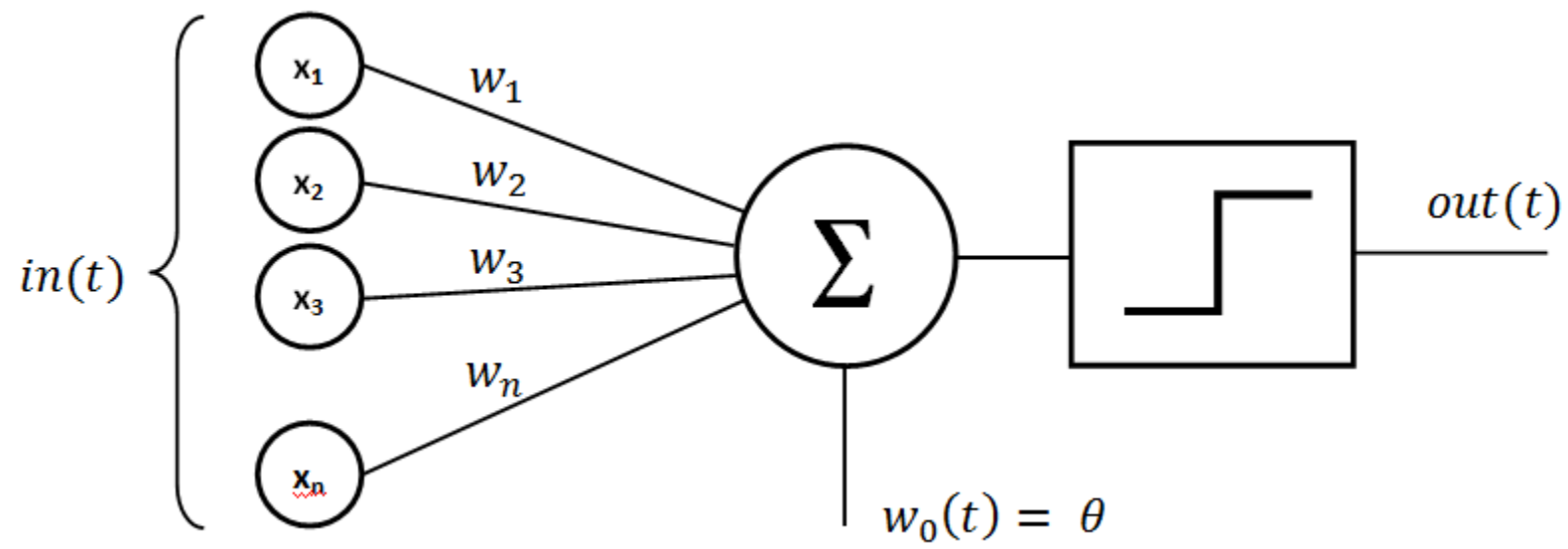
por Jose Naranjo

Recordemos

Un **perceptrón** es un algoritmo de aprendizaje supervisado para clasificaciones binarias y lineales que consiste de:

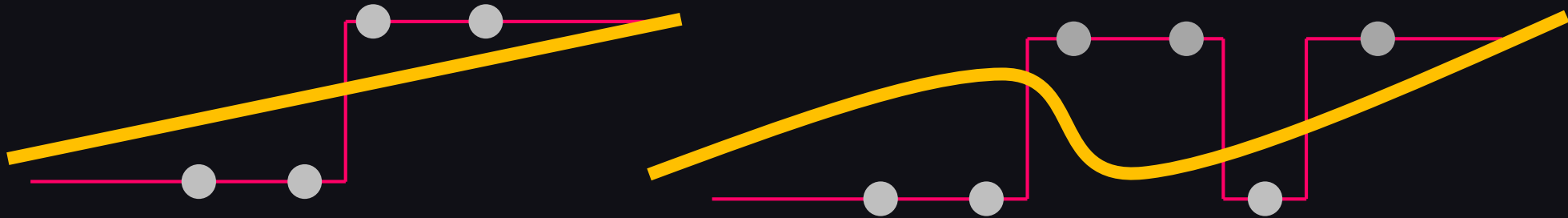
- entradas
- pesos
- un umbral ajustable o sesgo
- una suma ponderada
- una función de activación
- una salida

Si la suma es mayor al umbral, la salida es 1
(caso contrario, la salida es 0).

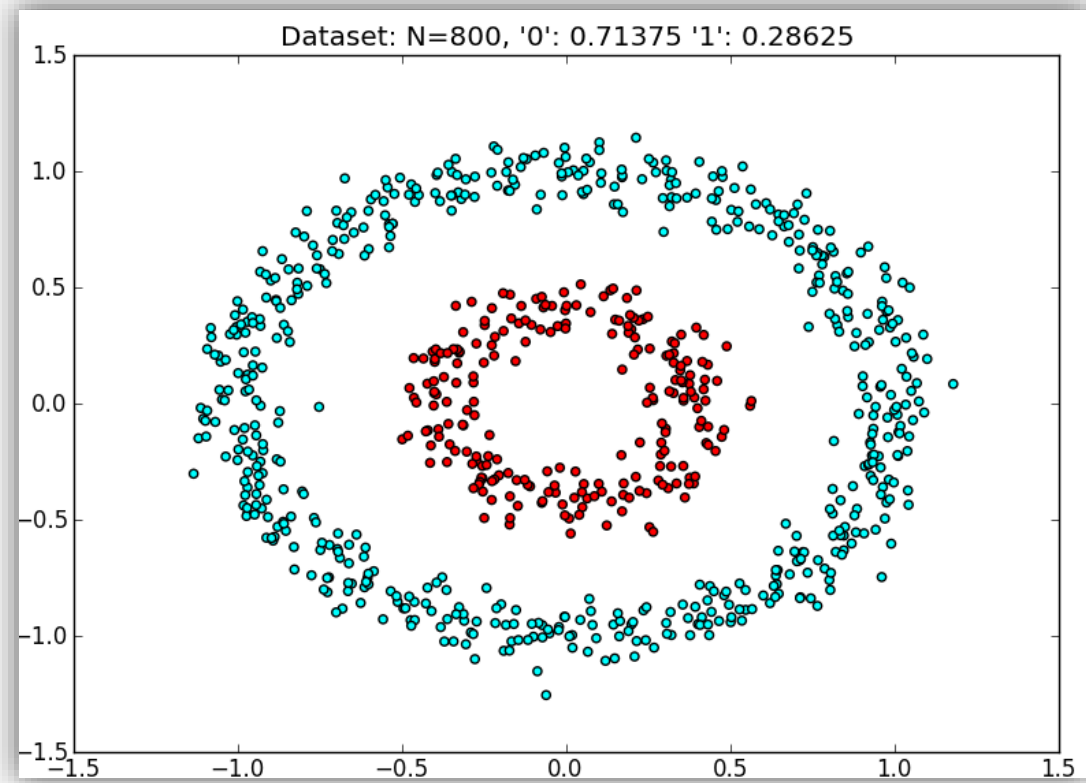


Lo que hace el perceptrón es particionar el espacio de características.

- binariza las salidas a través de una **función de activación**
- crea una frontera de decisión a través de un **hiperplano lineal**



¿Qué ocurre cuando la data no es linealmente separable?

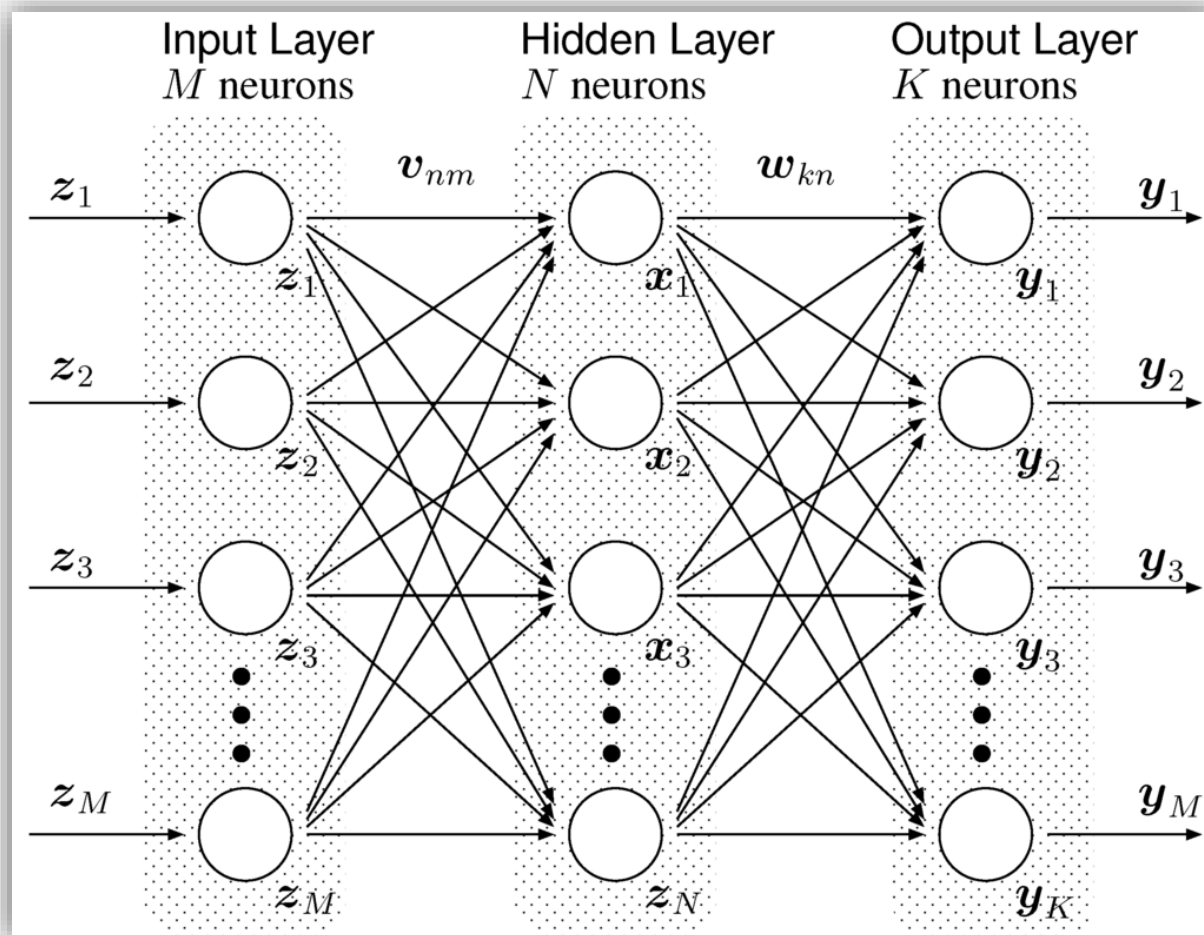


Perceptrón multicapa

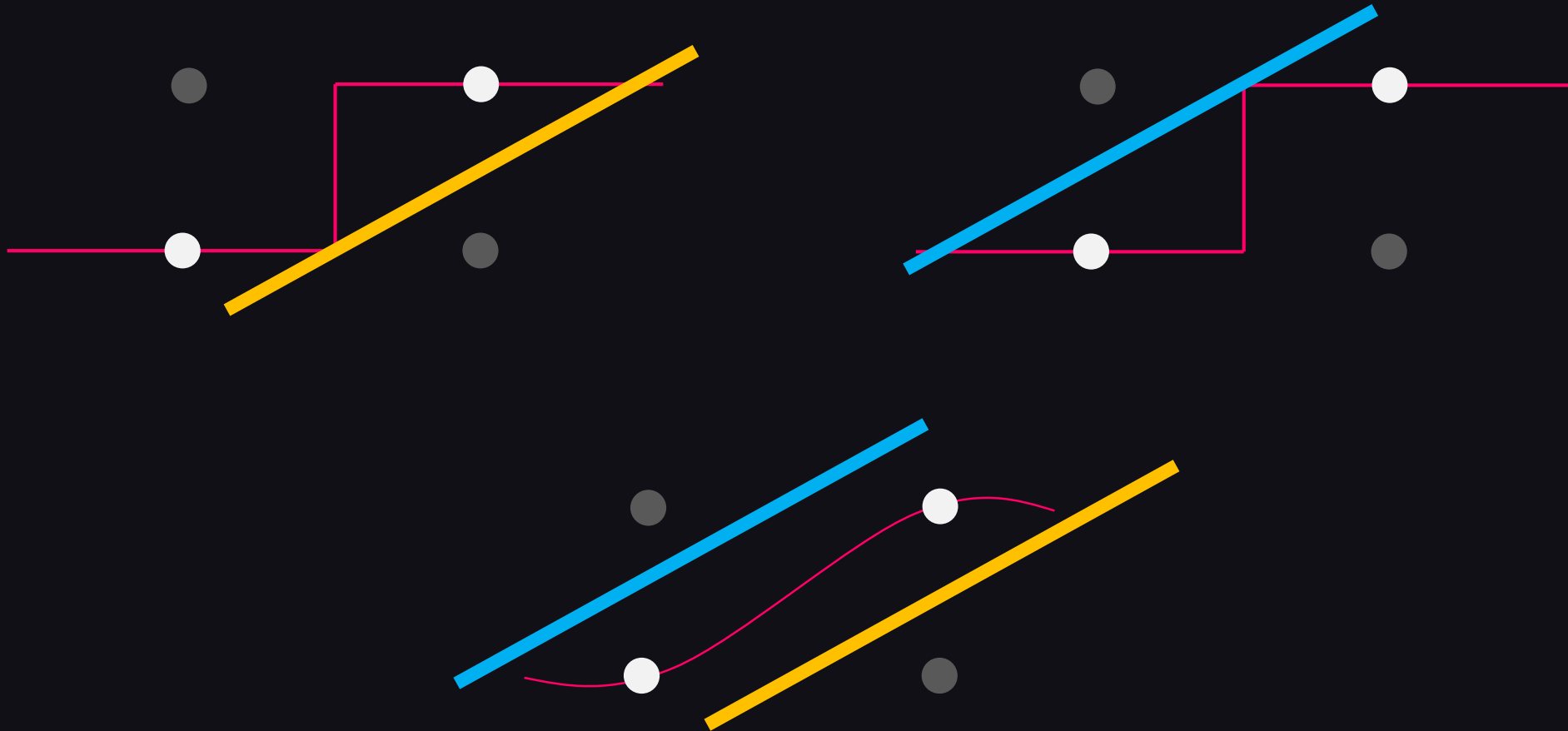
Un **perceptrón multicapa** (**Multilayer Perceptron o MLP**) es una **modificación del perceptrón estándar** que puede distinguir data que no es linealmente separable.

- Consiste de **múltiples capas** de nodos en un **grafo dirigido**, en el que cada capa se conecta con la siguiente de manera completa.
- Utiliza **propagación hacia atrás** (Backpropagation) para entrenar.

Todos los nodos (menos los de entrada) tienen **funciones de activación**:
i.e. cada nodo es un perceptrón



Cada nodo puede particionar el espacio de características de manera distinta, lo que permite separar data que no es separable linealmente.



Propagación hacia atrás

La **propagación hacia atrás (Backpropagation o BP)** es un **método de entrenamiento** que actualiza los pesos del sistema para reducir el error.

El error ($|\text{deseado} - \text{real}|$) se calcula a través de un método de optimización como descenso gradiente (Gradient Descent).

La **combinación de BP y Gradient Descent** se puede resumir en 2 fases:

1. **Propagación** (encontrar las diferencias entre las salidas objetivo y las salidas actuales de todas las neuronas)
2. **Actualización** (hallar el gradiente del peso multiplicando las diferencias y las activaciones y restándole un factor de aprendizaje)



Ventajas

- Tiene la capacidad de aprender modelos **no lineales**
- Puede aprender modelos en **tiempo real**
- Es **fácil de implementar**
- Sirve tanto para clasificación **como para regresión**

Desventajas

- **No es muy escalable** con respecto al número de características
- Puede haber más de un mínimo local lo que hace que a veces **se atore en el entrenamiento o diverja**
- Los pesos se inicializan aleatoriamente lo que puede dar **resultados muy variados** dependiendo de la distribución de la data
- **Tiene muchos parámetros:** número de capas ocultas, neuronas ocultas, iteraciones y tasa de aprendizaje.