Tema 12. Conocimiento en aprendizaje

Razonamiento y Representación del Conocimiento

Introducción

- Aprendizaje automático:
  - Supervisado
  - No supervisado
  - Por refuerzo
  - Genético

Introducción

- Representación del conocimiento
  - Proceso automático
  - ¿Sabemos cómo se representa el conocimiento aprendido? → Principal problema
  - ¿Podemos inferir relaciones causales entre el resultado del entrenamiento y el conocimiento adquirido?

Introducción

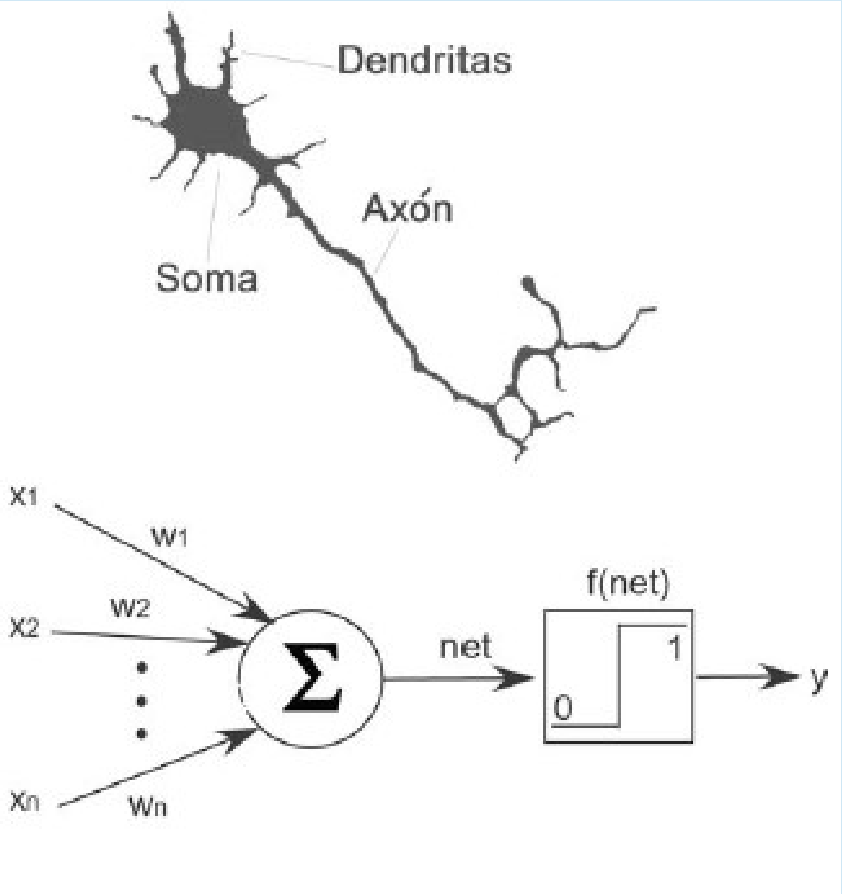
- Muchos métodos de aprendizaje automático
  - Podemos centrarnos en uno de los más generales y utilizados:
    - Redes neuronales

Redes Neuronales

- ¿Qué son?
  - Sistema conexionista
  - Busca imitar el funcionamiento del cerebro de los seres vivos
  - Fundamento: reforzar o debilitar las conexiones de los elementos de la red para obtener el resultado deseado

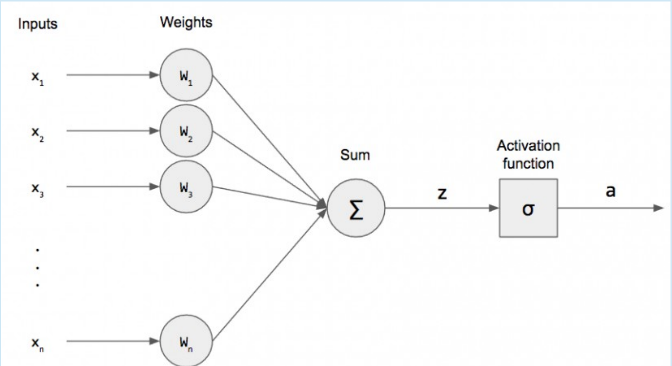
Redes neuronales

- **Neurona: modelo biológico**
  - Entradas: dendritas
  - Integración: en el soma. Dispositivos “todo-o-nada” (solo se dispara salida si las entradas superan un nivel (umbral)
  - Salidas: el axón transporta la señal de salida a otras neuronas. Conecta con sus dendritas a través de **sinapsis**
- **Neurona: modelo computacional**
  - Entradas: Números reales
  - Integración: suma ponderada (net) por los **pesos sinápticos** seguida de una función de activación f(net)
  - Salida: resultado y=f(net)



Perceptrón simple

- Unidad de procesamiento muy simple
  - Integrar las entradas (suma) → net
- Cada entrada está ponderada por un **peso** → suma ponderada de las entradas de la neurona
  - Las entradas son valores reales
- Salida: función de activación f(net) devuelve 1 o 0 dependiendo de si se supera un cierto umbral Θ (sesgo, *bias*) → función escalón
- Aprendizaje: modificar los pesos para adecuar la respuesta de la red a las salidas esperadas



$$net = \sum_{i=1}^{N} w_i x_i, \quad y = f(net) = \begin{cases} 1 & si\; net > \Theta \\ 0 & otro\; caso \end{cases}$$

Feed Forward Network

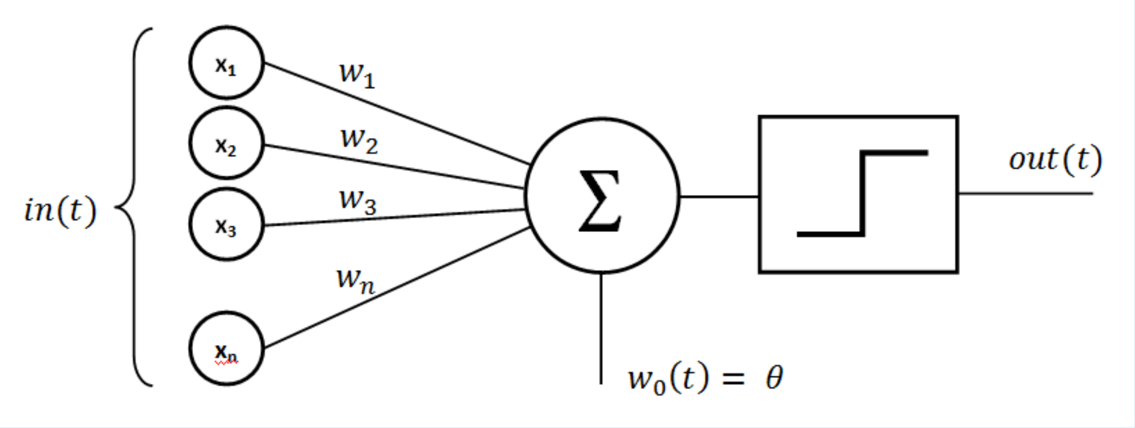
Perceptrón simple

- Simplificando el perceptrón
- Podemos simplificar el perceptrón si integramos el umbral Θ (sesgo, bias) dentro de la integración de la neurona

$$net = \sum_{i=1}^{N} (w_i x_i) - \Theta ,$$

$$net = \sum_{i=0}^{N} w_i x_i , con\, x_0 = -1$$

$$y = f(net) = \begin{cases} 1 & si\, net > 0 \\ 0 & otro\, caso \end{cases}$$



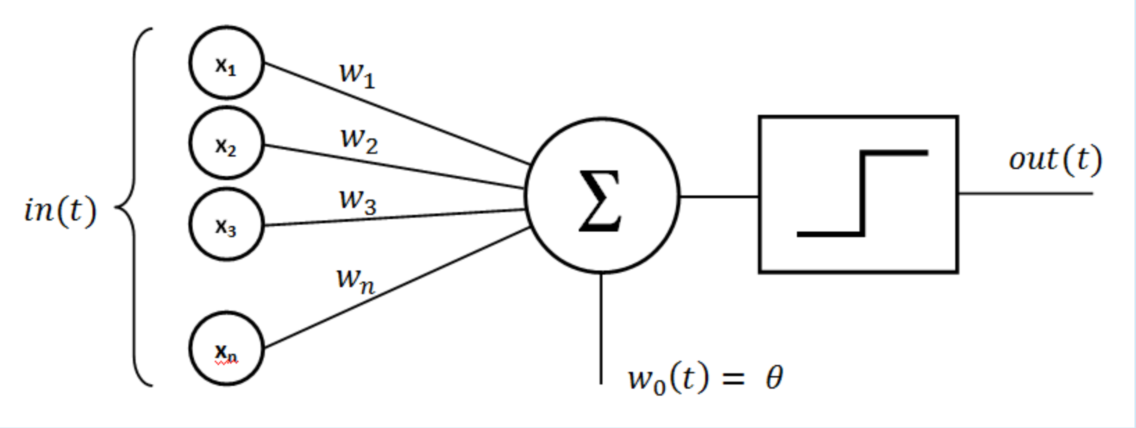
Perceptrón simple

- El conocimiento del perceptrón se encuentra repartido entre los pesos w_i
- Modificando los pesos, podemos modificar la salida del perceptrón para un mismo vector de entrada

$$net = \sum_{i=1}^{N} (w_i x_i) - \Theta ,$$

$$net = \sum_{i=0}^{N} w_i x_i , con\, x_0 = -1$$

$$y = f(net) = \begin{cases} 1 & si\, net > 0 \\ 0 & otro\, caso \end{cases}$$



Perceptrón simple

- Interpretación geométrica

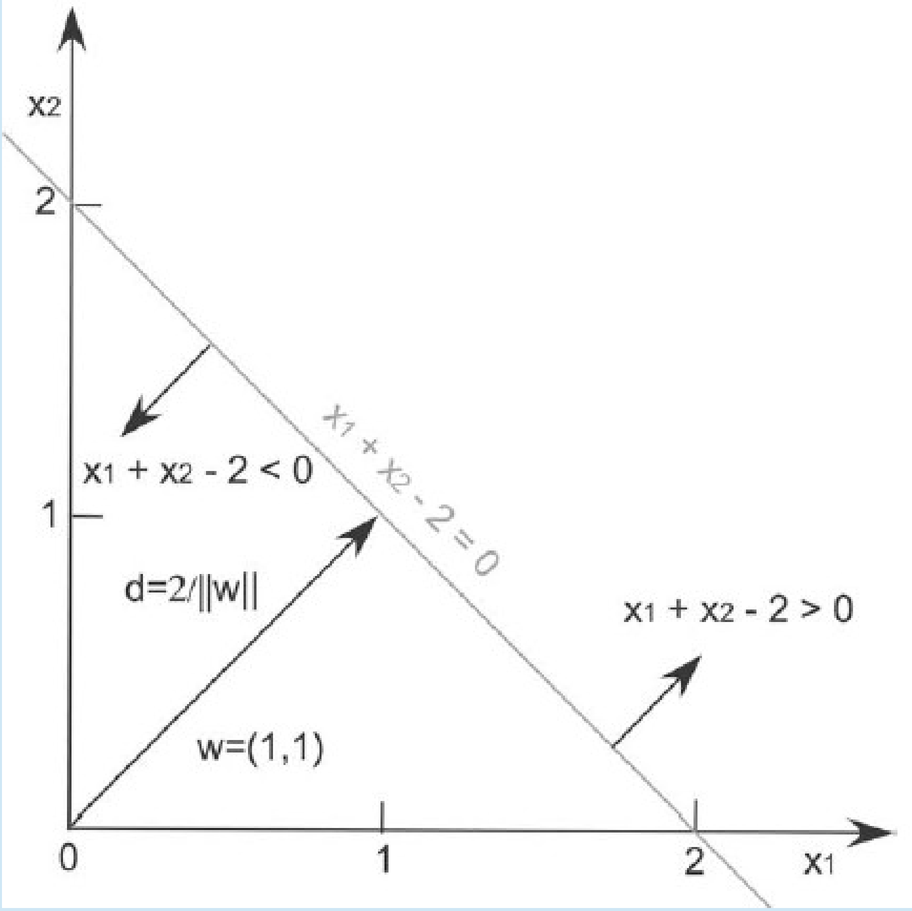
$$y = f\left(\sum_{i=0}^{N} w_i x_i\right) = \begin{cases} 1 & si\, net > 0 \\ 0 & otro\, caso \end{cases}$$

  - Analizando esta ecuación:
    - N=2 → net = $w_1x_1 + w_2x_2 - w_0$
    - N=3 → net = $w_1x_1 + w_2x_2 + w_3x_3 - w_0$
    - N=k → net = $w_1x_1 + \ldots + w_kx_k - w_0$
  - ¿Qué es lo que tenemos?

Perceptrón simple

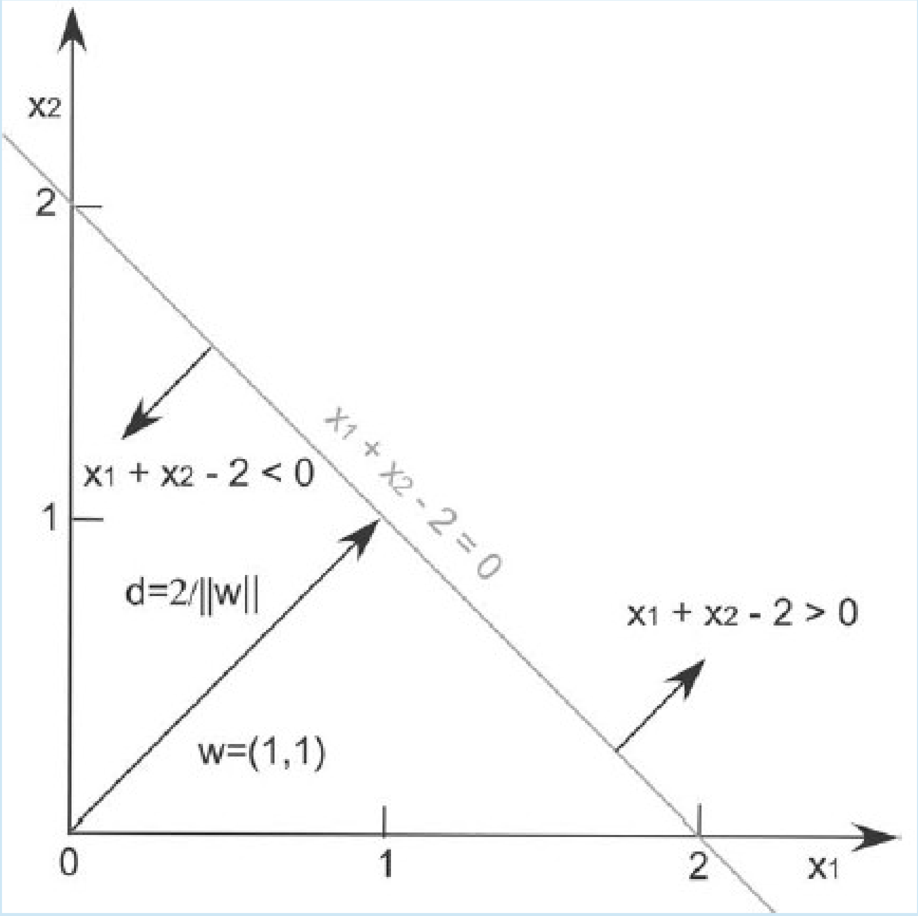
- Interpretación geométrica
  - La neurona define un hiperplano de forma que los ejemplos etiquetados con y=1 caen al lado positivo y los etiquetados con y=0 al lado negativo

$$\sum_{i=1}^{N} (w_i x_i) - \Theta = 0$$



Perceptrón simple

- Uso del perceptrón
  - Paso Feed Forward
  - Los datos son pasados a las entradas del perceptrón para que éste produzca una salida
  - Ejercicio:
    - (2, 2)
    - (7, -3)
    - (1, -1)



Entrenar el perceptrón simple

- Buscamos dar valor a los pesos del perceptrón para que sea capaz de reconocer la clase a la que pertenece un conjunto de ejemplos de entrenamiento
- Esos ejemplos están etiquetados:
  - Cada ejemplo tiene:
    - Un vector de características → datos de entrada
    - La clase a la que pertenece (y_d) → salida deseada

Entrenar el perceptrón simple

- Para un ejemplo del cto. De entrenamiento
  - ¿Podemos encontrar una configuración de pesos del perceptrón para que la salida sea igual a la clase del ejemplo?

Entrenar el perceptrón simple

- Para un ejemplo del cto. De entrenamiento
  - ¿Podemos encontrar una configuración de pesos del perceptrón para que la salida sea igual a la clase del ejemplo?

SÍ

Siempre

Entrenar el perceptrón simple

- Y con esos pesos, ¿podremos clasificar correctamente el resto de ejemplos?
  - Seguramente no
- Entonces, ¿qué hacemos con el perceptrón y sus pesos’
  - Buscar el conjunto de pesos que clasifique bien la mayoría de los ejemplos posibles

Entrenar el perceptrón simple

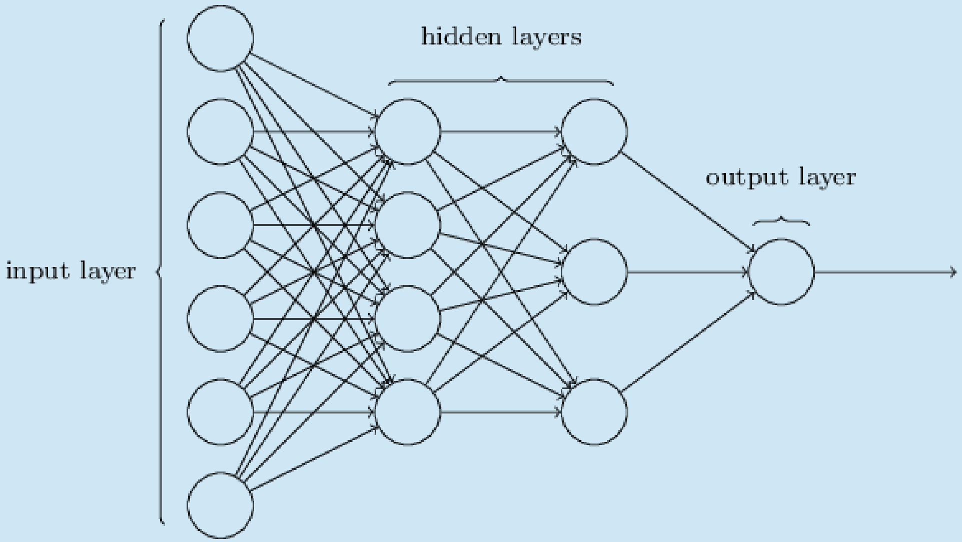
- ¿Podemos asegurar que el perceptrón simple acabará aprendiendo a clasificar todos los ejemplos de entrenamiento?
  - No, no podemos asegurarlo
  - Depende de la distribución espacial de los datos del conjunto de entrenamiento
    - Conjuntos linealmente separables

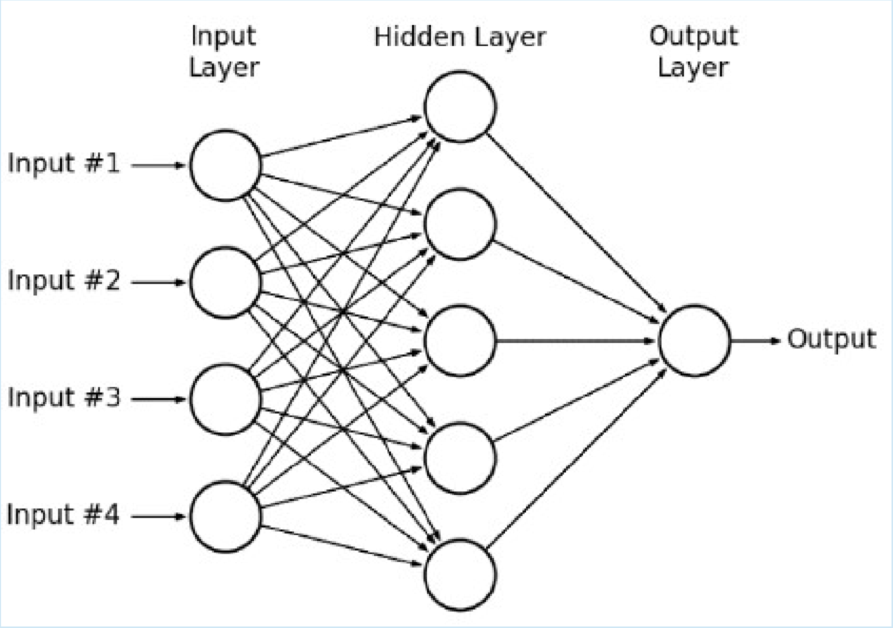
Uniendo perceptrones

- Una red neuronal está compuesta de varias neuronas :)
- Al igual que en el modelo biológico, podemos enlazar la salida de una neurona con la entrada de una o varias neuronas diferentes
- Si organizamos las neuronas por capas y conectamos la salida de las neuronas de una capa con la entrada de todas las neuronas de la siguiente capa tenemos el perceptrón multicapa

Perceptrón multicapa

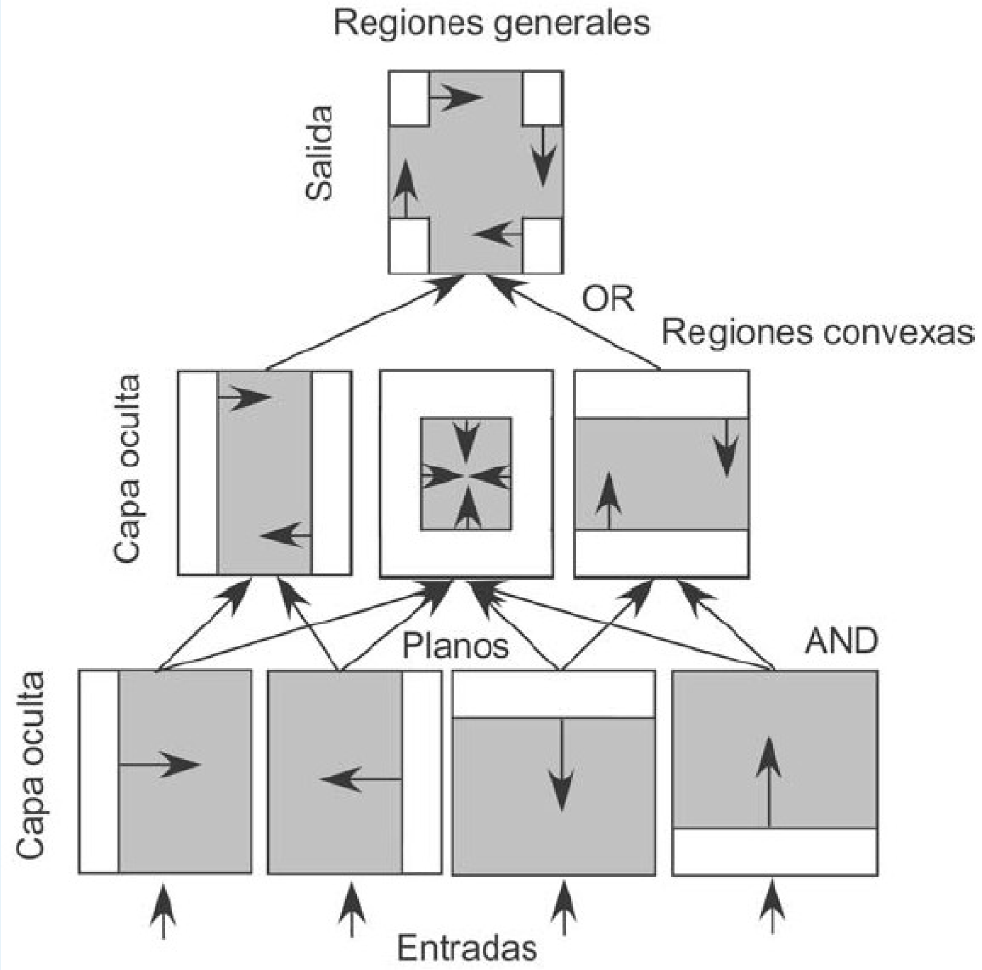
- Neuronas agrupadas por capas
- Hoy se conoce como red *fully connected*





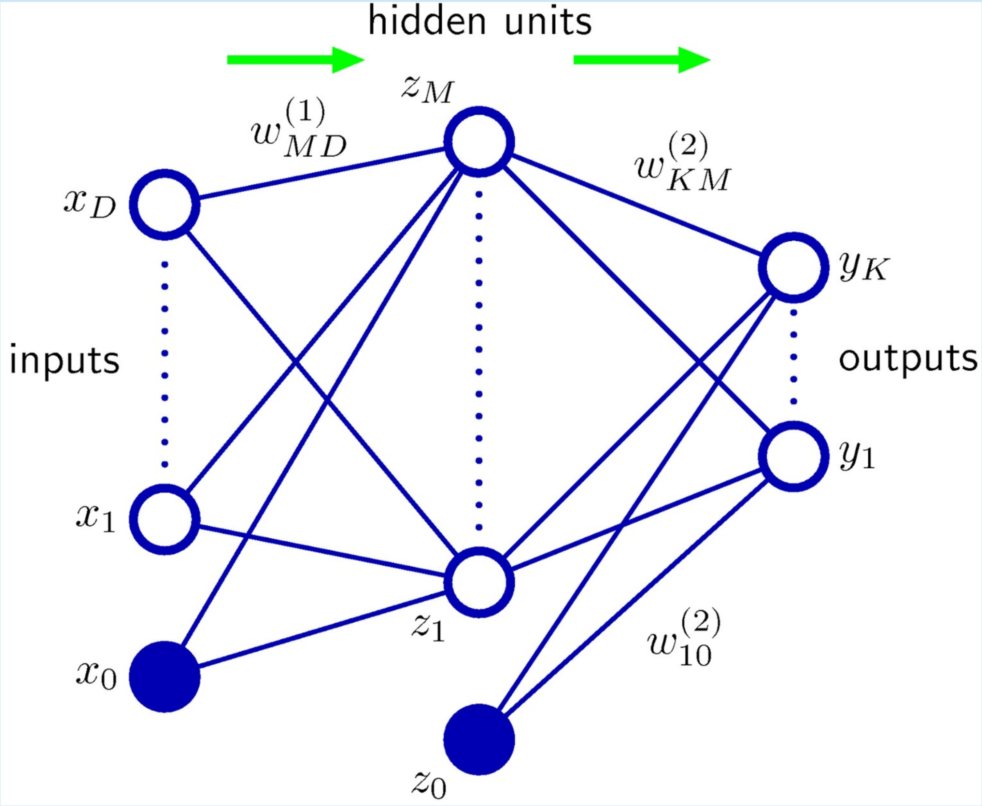
Perceptrón multicapa

- Interpretación geométrica
  - Problemas con regiones de decisión más complejas exigen distintas estrategias de separación
  - Dichas estrategias las proporcionan las capas ocultas



Perceptrón multicapa

- Interpretación



$$y_k(x,w)=\sigma\left(\sum_{j=0}^{M} w_{kj}^{(2)} h\left(\sum_{i=0}^{D} w_{ji}^{(1)} x_i\right)\right)$$