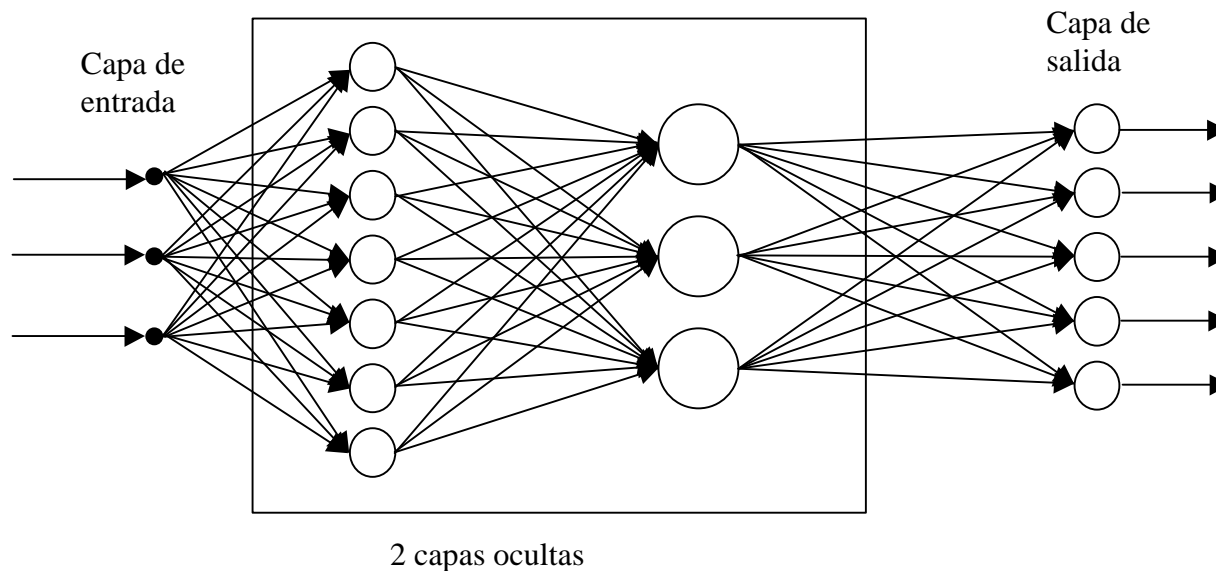


Redes Unidireccionales (Feedforward)

- ✓ El problema genérico más simple que resuelven las redes neuronales es la síntesis de sistemas que realizan una determinada asociación de datos entrada/salida.
- ✓ Los pesos sinápticos de las neuronas se ajustan de tal manera que ante cada entrada la red responda de una manera preestablecida.
- ✓ Asociación \Leftrightarrow Problema de aproximación de funciones:
 - ✗ Datos de entrada y salida se asimilan con vectores multidimensionales.
 - ✗ Se pretende que el valor real de la salida generada por la red cuando se estimule con un cierto dato de entrada, se parezca lo más posible al correspondiente dato de salida.
 - ✗ La ventaja de usar redes neuronales para este problema de aproximación de funciones es que la red ofrece una gran capacidad de generalización.
- ✓ Asociación \Leftrightarrow Problema de clasificación de datos:
 - ✗ La salida de la red ante cada dato de entrada debe decidir la clase a la que pertenece ese dato de entrada.
 - ✗ Interesa la capacidad de discriminar el dato que ha servido de entrada a la red.
 - ✗ Reconocimiento de patrones.
 - ✗ La ventaja de usar redes neuronales radica en su gran capacidad para sintetizar funciones discriminantes no lineales.

- ✓ El modelo más utilizado para estas redes unidireccionales (feedforward) es el perceptrón.
 - ✗ Red multicapa.
 - ✗ Las neuronas de cada capa reciben información de las de la capa precedente y la transmiten a la capa siguiente.
 - ✗ Las neuronas no están interconectadas dentro de cada capa de la red.

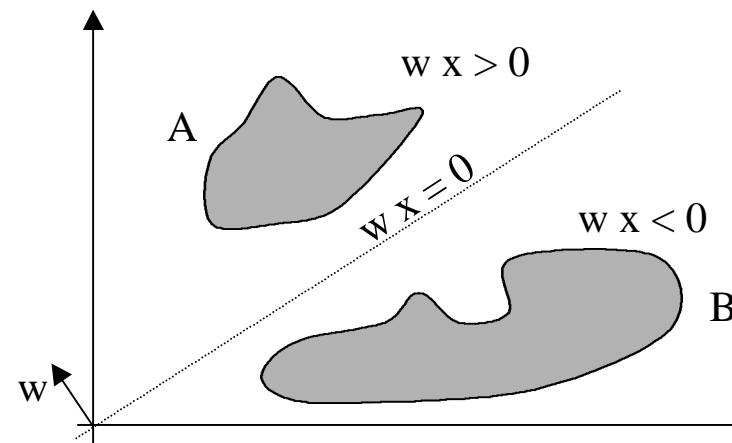


- ✗ El perceptrón más común incluye sólo una capa oculta.

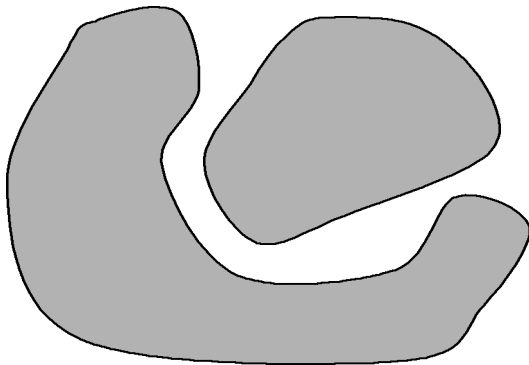
El Perceptrón Simple

- ✓ Red formada por una sola capa de neuronas desconectadas entre sí, que reciben todas ellas las mismas entradas.
- ✓ Estudiaremos el perceptrón monocapa que contiene una sola neurona.
- ✓ Se usa como función de transferencia de la neurona:
$$y=f(x) = \text{sgn}(w \cdot x), \text{ siendo } \text{sgn}(t) = \begin{cases} +1 & \text{si } t > 0 \\ 0 & \text{si } t = 0 \\ -1 & \text{si } t < 0 \end{cases}$$
- ✓ Con este modelo el perceptrón acepta como entrada un vector de \mathbb{R}^n , y emite como salida uno de los valores +1, 0 y -1.
- ✓ El perceptrón puede ser usado como función discriminante.

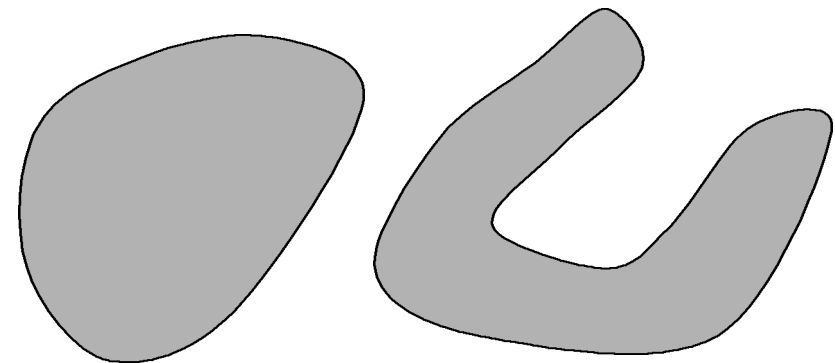
La utilización del perceptrón simple como función discriminante equivale a utilizar formas lineales como funciones discriminantes.



- ✓ Matemáticamente el aprendizaje del perceptrón consiste en la determinación de un vector w que satisfaga las condiciones previas.
- ✓ Para que la función lineal $w \cdot x$ actúe como función discriminante de dos conjuntos A y B , cada uno de estos conjuntos debe estar contenido en un semiespacio diferente, de los dos definidos por el hiperplano.



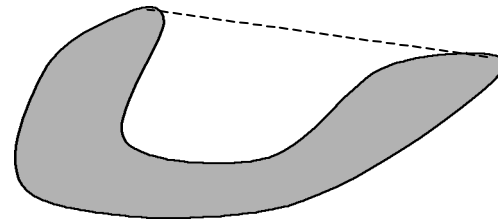
Conjuntos no separables linealmente



Conjuntos convexo y no convexo

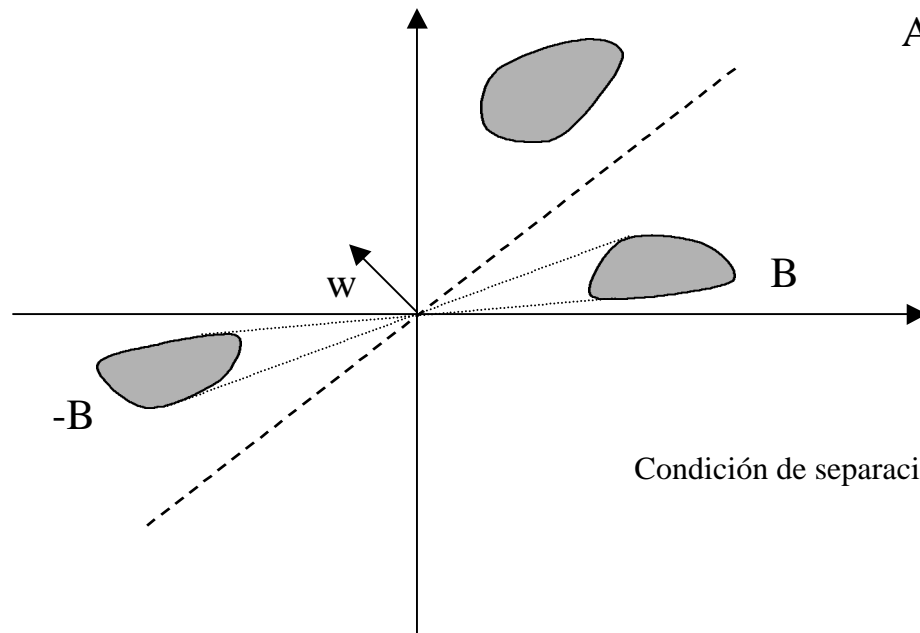
- ✓ Un conjunto C del espacio R^n es convexo si para cada par de puntos de C todos los puntos del segmento que los enlaza también pertenecen al conjunto C .
- ✓ Dos conjuntos convexos disjuntos pueden ser separados linealmente.

- ✓ Para que dos conjuntos no convexos sean linealmente separables es necesario y suficiente que sus envolventes convexas sean conjuntos disjuntos.



Envolvente conexa de un conjunto

- ✓ Para el caso del perceptrón, la separación ha de realizarse además mediante un hiperplano que pase por el origen.



A

X: Unión de puntos de A y -B

Separabilidad lineal \Leftrightarrow El origen no se encuentra en X

$$x \in X, \quad w \cdot x > 0$$

Condición de separación desde el origen

El Algoritmo del Perceptrón

- ✓ Dados p vectores x_i , con $i=1, \dots, p$ de \mathbb{R}^n , tales que su envolvente convexa no contiene al origen, se trata de encontrar un vector w que de producto escalar positivo con cada uno de ellos, es decir tal que

$$w \cdot x_i > 0, \quad i=1, \dots, p$$

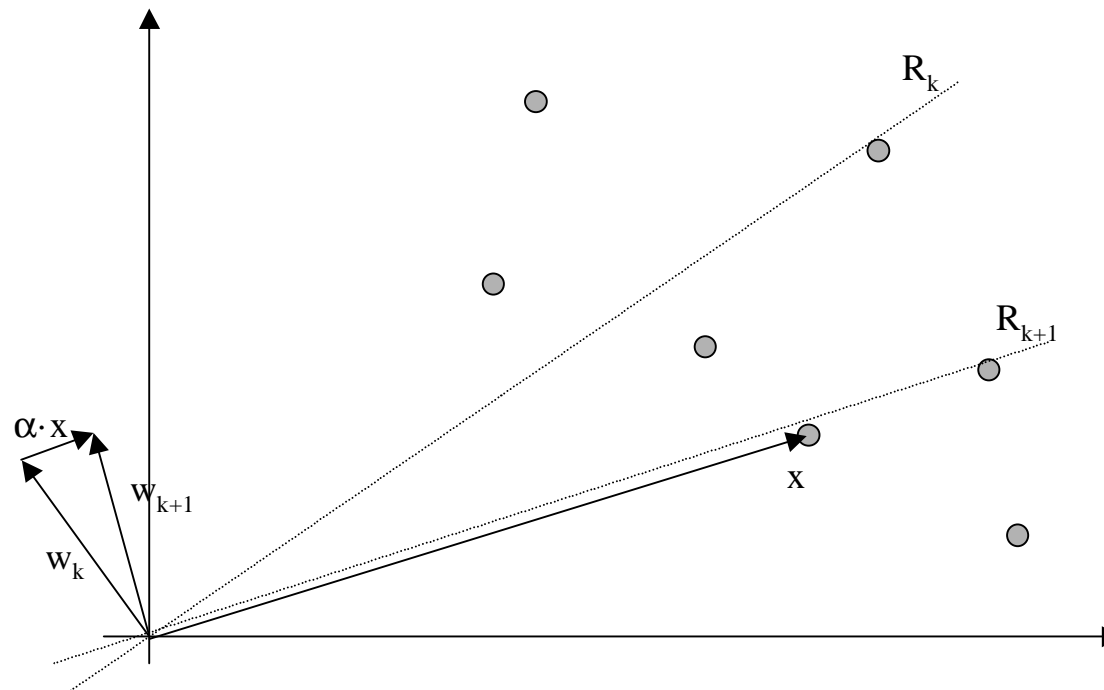
- ✓ Se puede encontrar la solución mediante el siguiente algoritmo iterativo:
- ✗ Elijase arbitrariamente un vector inicial w_0 .
 - ✗ En cada etapa k se dispone de un vector w_k y se calcula su producto escalar con un cierto vector $x_{i(k)}$ del sistema de vectores datos del problema.
 - ✗ Si el producto escalar anterior es positivo, el vector de pesos permanece invariable, es decir se toma

$$w_{k+1} = w_k$$

En caso contrario, cuando ese producto escalar es no positivo, se elige como nuevo vector de pesos:

$$w_{k+1} = w_k + \alpha x_{i(k)}$$

donde el parámetro α , que controla la tasa de adaptación, se elige como un número real tal que $0 < \alpha < 1$.



✓ Criterio de aprendizaje neuronal de Hebb:

- ✘ Cuando una neurona tiene un nivel positivo de actividad y se siente estimulada por otra neurona, tiende a producirse un refuerzo en la conexión que las enlaza, mientras que tiende a disminuir su conexión si se siente inhibida.

El Adaptador Lineal

- ✓ El algoritmo del perceptrón proporciona una solución en un número finito de etapas en el supuesto de que tal sistema admita solución. Si no existe solución el sistema no es convergente.
- ✓ Una forma diferente de encontrar solución para:

$$w \cdot x_i > 0, \quad i=1, \dots, p$$

es seleccionar un vector **b** de componentes positivas y buscar una solución del sistema de ecuaciones

$$w \cdot x_i = b_i, \quad i=1, \dots, p$$

- ✗ Puede ocurrir que para el vector **b** elegido este sistema no tenga solución y sin embargo el sistema original sí lo tenga.
- ✗ Si el sistema de desigualdades es compatible es seguro la existencia de algún vector **b**.
- ✗ Si el número 'p' de puntos es muy grande pueden ser impracticables los métodos ordinarios de solución de sistemas lineales. También puede ocurrir el caso de no disponer al comienzo de todos estos puntos.
- ✗ Por ésto es conveniente elegir un método adaptativo, en el que se van incorporando secuencialmente las ecuaciones y se va buscando satisfacer las nuevas ecuaciones a la vez que se sigue en lo posible cerca de una solución de las antiguas.

- ✓ Si w_k es solución de las primeras k ecuaciones, al añadir una ecuación más probablemente no será solución de todas ellas.
- ✓ Un método para adaptar esta solución consiste en proyectar ortogonalmente el punto w_k sobre el hiperplano $w \cdot x_{k+1} = b_{k+1}$.

$$w = w_k + \mathbf{I} x_{k+1}$$

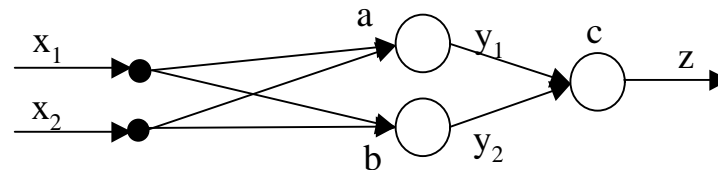
$$w_k \cdot x_{k+1} + \mathbf{I} \|x_{k+1}\|^2 = b_{k+1}$$

$$w_{k+1} = w_k + \frac{b_{k+1} - w_k \cdot x_{k+1}}{\|x_{k+1}\|^2} x_{k+1}$$

- ✗ Si se aplica reiteradamente este esquema a un sistema compatible y determinado de ecuaciones lineales, la sucesión de vectores generada converge hacia la solución del sistema.
- ✓ Cuando se aplica este algoritmo tomando como vector b aquel cuyas componentes son todas 1, la fórmula de adaptación es la llamada regla delta de Widrow-Hoff para el aprendizaje del perceptrón.

Limitaciones del Perceptrón Simple

- ✓ El perceptrón monocapa sólo puede sintetizar clasificadores lineales, es decir, discriminar conjuntos linealmente separables.
- ✓ Muchas funciones booleanas no son sintetizables con el perceptrón monocapa (XOR).
- ✓ Puede conseguirse una red neuronal que realiza la función XOR si se emplean dos capas de neuronas:



- ✓ Se puede demostrar que cualquier función booleana de n variables se puede sintetizar con una red neuronal de dos capas. Sin embargo no es una tarea trivial deducir los pesos de las neuronas.
- ✓ La razón de la dificultad de programar el aprendizaje es que la red bicapa no puede detectar la influencia que tienen las modificaciones de los pesos de las neuronas de la primera capa en la salida de las neuronas de la segunda capa (credit assignment problem).

El Algoritmo de Retropropagación del Error

- ✓ Aprendizaje de una red multicapa cuyas neuronas tienen una función de transferencia derivable. (función signo \rightarrow función sigmoide).
- ✓ Algoritmo 'backpropagation' para una red con una capa oculta.
 - ✗ Capa de entrada: vector $x \in R^n$; Capa oculta: p neuronas; Capa de salida: 1 neurona.
 - ✗ v_i : Vector de pesos de la i -ésima neurona de la capa oculta.
 - ✗ w : Vector de pesos de la neurona de salida.
 - ✗ Si f es la función de transferencia de las neuronas, la ecuación funcional de la red neuronal:

$$z = f\left(\sum_{i=1}^p w_i f(v_i \cdot x)\right)$$

- ✗ Para ajustar los pesos de todas las neuronas mediante el algoritmo de retropropagación, primero se ajusta el peso de la neurona de salida:

$$w(k+1) = w(k) + I_k (z_{k+1} - j(x_{k+1}; u(k))) f'(b_{k+1}) y_{k+1}$$

- ✗ El ajuste de los pesos de la capa oculta será:

$$v_i(k+1) = v_i(k) + I_k (z_{k+1} - j(x_{k+1}; u_r(k))) f'(b_{k+1}) f'(a_m) w_m(k+1) x_k$$

donde en $u_r(k)$ ya se recogen los valores actualizados de w_k .

- ✓ Todos los cálculos previos son válidos cuando la función f de transferencia de las neuronas es una función cualquiera derivable. Las expresiones se simplifican notablemente si se utiliza la función sigmoide.

$$f(t) = \text{th}(\mathbf{at}) = \frac{e^{at} - e^{-at}}{e^{at} + e^{-at}}$$

- ✓ El algoritmo de retropropagación es una implementación del método del gradiente y adolece de todos los inconvenientes que tiene este método.
- ✓ Existe la posibilidad de que el proceso de aprendizaje no converja a los pesos óptimos.
- ✓ Ampliando el número de neuronas de las capas ocultas de la red se consigue una mejor codificación de los datos de entrada y se reduce la presencia de mínimos locales.
- ✓ Es necesario adoptar algún criterio numérico para terminar el proceso iterativo puesto que los pesos pueden oscilar alrededor de la solución óptima.
 - ✗ Posibles criterios de finalización:
 - Módulo del gradiente.
 - Observación de los valores de la función objetivo.
 - Método de validación cruzada.

Posibilidades del Perceptrón Multicapa

- ✓ Síntesis de funciones lógicas booleanas.
 - ✗ El perceptrón simple sólo puede implementar una pequeña fracción de funciones lógicas.
 - ✗ Un perceptrón con una sola capa ya puede representar cualquier función lógica.
- ✓ Problemas de clasificación.
 - ✗ Para el perceptrón simple la función de decisión es lineal.
 - ✗ En un perceptrón de dos capas el contorno de decisión puede ser la frontera de cualquier conjunto convexo.
 - ✗ Para contornos de decisión más generales basta con emplear un perceptrón de tres capas, es decir, con sólo dos capas ocultas.
- ✓ Aproximación de funciones no lineales.
 - ✗ El perceptrón simple puede ajustar tan finamente como se quiera cualquier función continua.
- ✓ Limitaciones
 - ✗ No es posible en general determinar a priori el tamaño de la red que mejor resuelve un problema en concreto. Con una red demasiado simple puede que el problema no tenga una solución adecuada, pero una red demasiado grande complica el proceso de aprendizaje.
 - ✗ Ejemplo: Ajuste de datos con un polinomio.