APRENDIZAJE AUTOMÁTICO

Hasta aquí, hemos considerado el problema de memoria asociativa a través de redes recurrentes, en el cual hemos construido de inicio reglas, como la de Hopfield o sus modificaciones. Estas reglas "NO SE APRENDIERON", se "CONSTRUYERON".

A partir de hoy analizaremos un nuevo paradigma, y veremos una nueva arquitectura.

Nota: el modelo de Hopfield también se puede "aprender", aunque no veremos aquí este algoritmo.

Como vimos, no es trivial encontrar una regla para un problema tan complejo como el problema de memoria asociativa. La idea en el caso de "aprendizaje" es encontrar un algoritmo "adecuado" para resolver un problema dado.

El modelo de Hopfield es un caso de sistema "auto-asociativo", que identifica muchas configuraciones con una única configuración. Aquí el input y el output tiene la misma forma y el mismo tamaño. Pero hay muchísimos problemas en los cuales el input es de un tipo y el output de otro.

En general, más allá de las redes neuronales pero también incluyéndolas, podemos decir que hay 3 tipos de algoritmos de aprendizaje. En el caso de redes, se trata de encontrar los acoplamientos W_{ij} en forma automática

APRENDIZAJE SUPERVISADO

En este caso disponemos de datos de entrada que ha sido ETIQUETADOS previamente, y nos permiten entonces encontrar el conjunto de las sinapsis que resuelven bien el problema para los ejemplos etiquetados, a los cuales llamamos "CONJUNTO DE ENTRENAMIENTO"

Ejemplo: clasificar correo electrónico entrante como spam o no spam.

Algunos algoritmos desarrollados para aprendizaje supervisados son:

- K-Nearest Neighbors
- Regresión lineal
- Regresión logística
- Support Vector Machine
- Clasificador bayesiano
- Arboles de decisión
- Random forest
- Redes Neuronales
- Redes Neuronales profundas.

APRENDIZAJE NO SUPERVISADO

En este caso no disponemos datos previamente etiquetados y el algoritmo debe clasificar o procesar la información por si solo.

Este algoritmo se usa generalmente para clasificar datos en pocas categorías.

Algunos algoritmos desarrollados para aprendizaje no supervisado son

- K-means
- Análisis de la componente principal
- Detección de anomalías.
- Redes neuronales

APRENDIZAJE POR REFUERZO

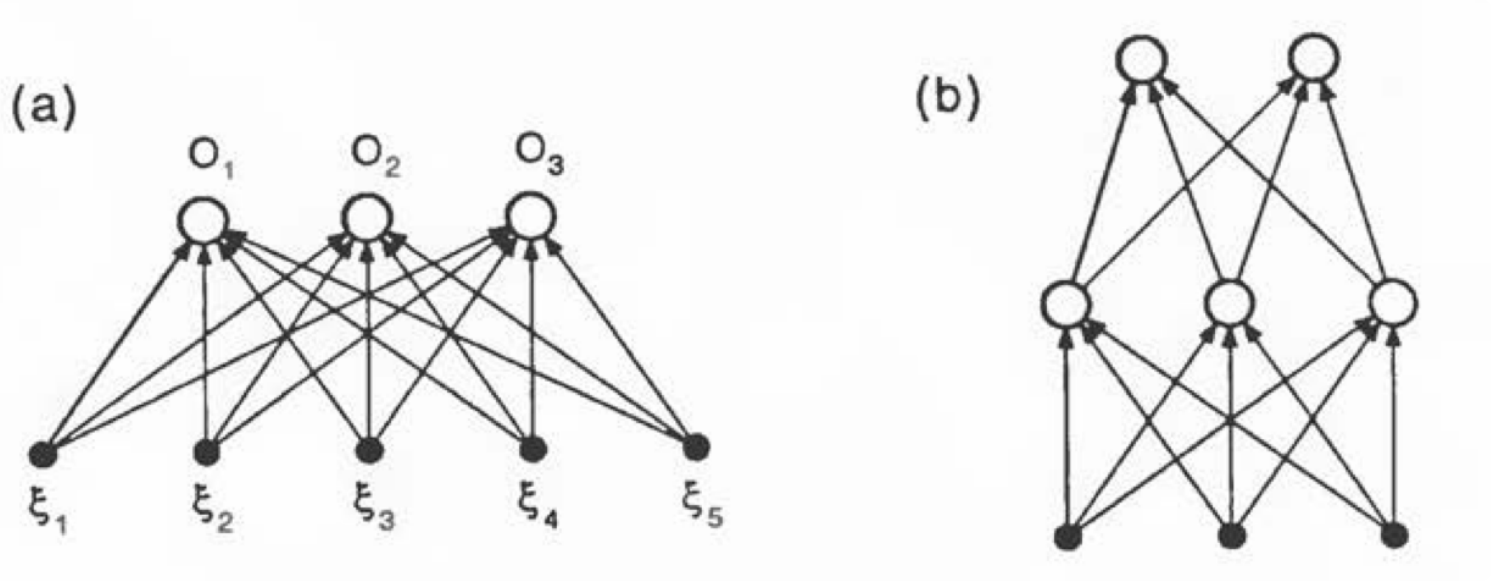
El algoritmo es un "agente autónomo" que explora un "espacio" desconocido y decide que acciones realizar mediante prueba y error. Recibirá premios y castigos. Crea la mejor estrategia posible para obtener la mayor recompensa. Esta estrategia definirá acciones a tomar ante cada situación que se enfrente.

- MDP: Markov Decision Process

REDES FEED-FORWARD

PERCEPTRON

1958: Frank Rosenblatt (Cornell Aeronautical Laboratory)



Las neuronas se organizan en capas o comodas. La información fluye de la entrada a la salida.

La salida es una capa

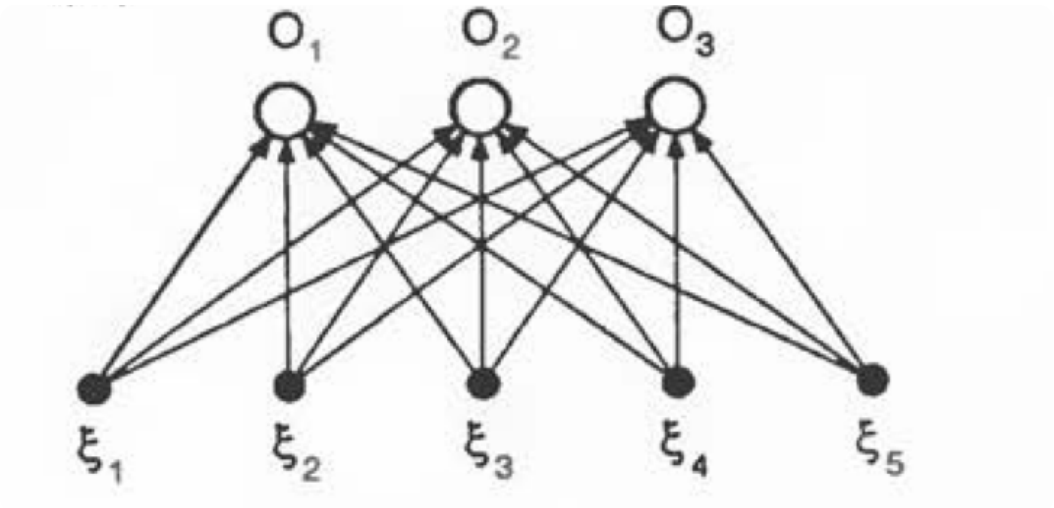
La entrada NO ES una capa.

En el ejemplo (a) hay 1 capa.

En el ejemplo (b) hay 2 capas.

- No se admiten conexiones entre neuronas de la misma capa
- No se admiten conexiones de una capa a otra previa
- Solo se admiten conexiones entre una capa y la siguiente.
- Las conexiones son asimétricas.

EL PERCEPTRON DE 1 CAPA



El vector

$$\vec{\xi} = (\xi_1, \xi_2, \xi_3, \ldots \xi_N)$$

representa la entrada. Las variables ξ_i pueden ser discretas o continuas.

El vector

$$\vec{O} = (O_1, O_2, \ldots O_M)$$

representa la salida. Tenemos $N \times M$ acoplamientos

$$W_{ij} \qquad i = 1, 2, \ldots, M \quad j = 1, 2, \ldots, N$$

Tenemos que encontrar un conjunto $\{W_{ij}\}$ tal que ante una entrada dada

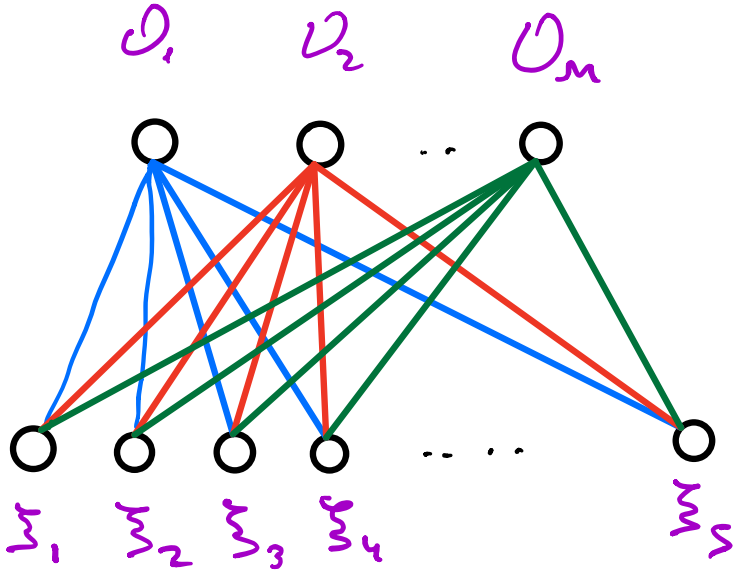
$$\vec{\xi}^{\mu} = (\xi^{\mu}_1, \xi^{\mu}_2, \ldots, \xi^{\mu}_N)$$

nos lleve a una salida determinada

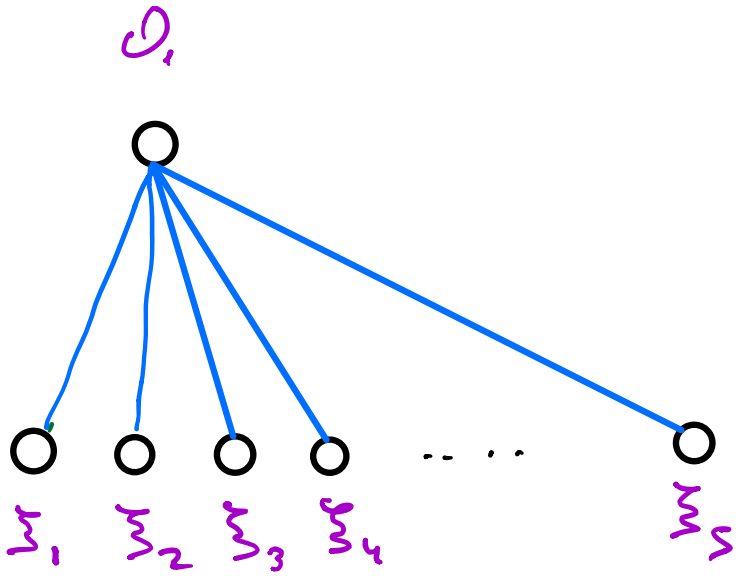
$$\vec{O}^{\mu} = (O_1^{\mu}, O_2^{\mu}, \ldots, O_M^{\mu})$$

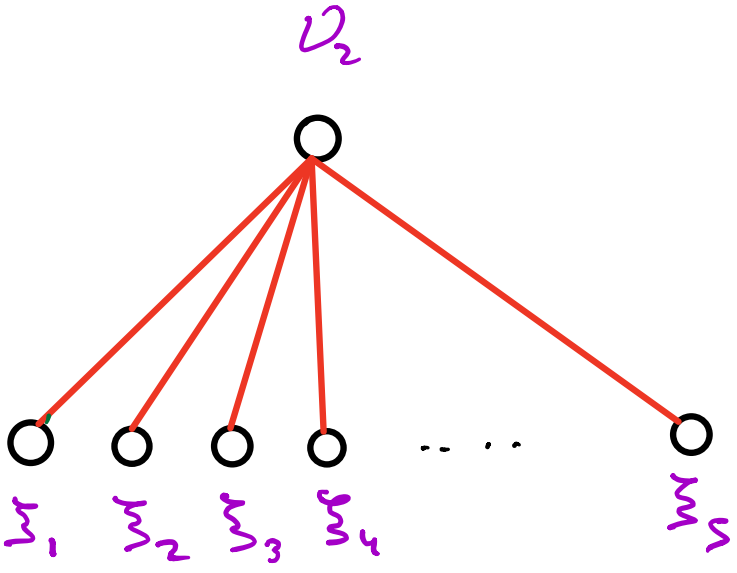
EL PERCEPTRON SIMPLE

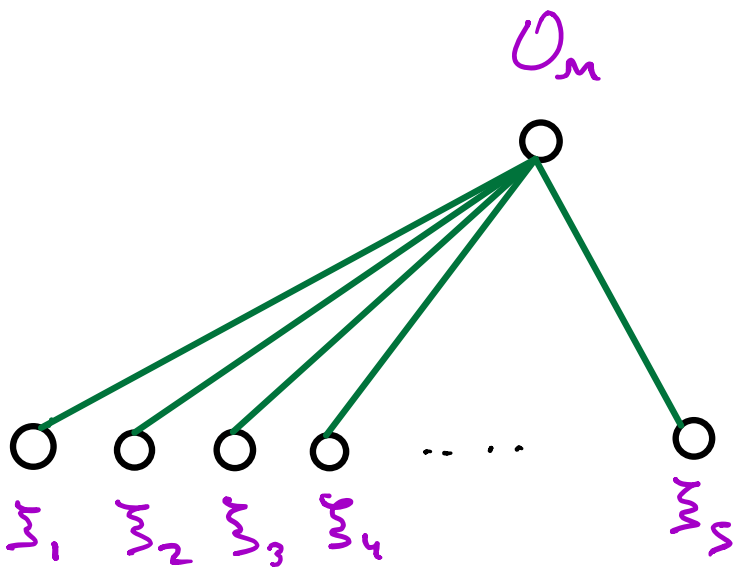
Podemos pensar un perceptron de 1 capa con M perceptrones con 1 salida cada una



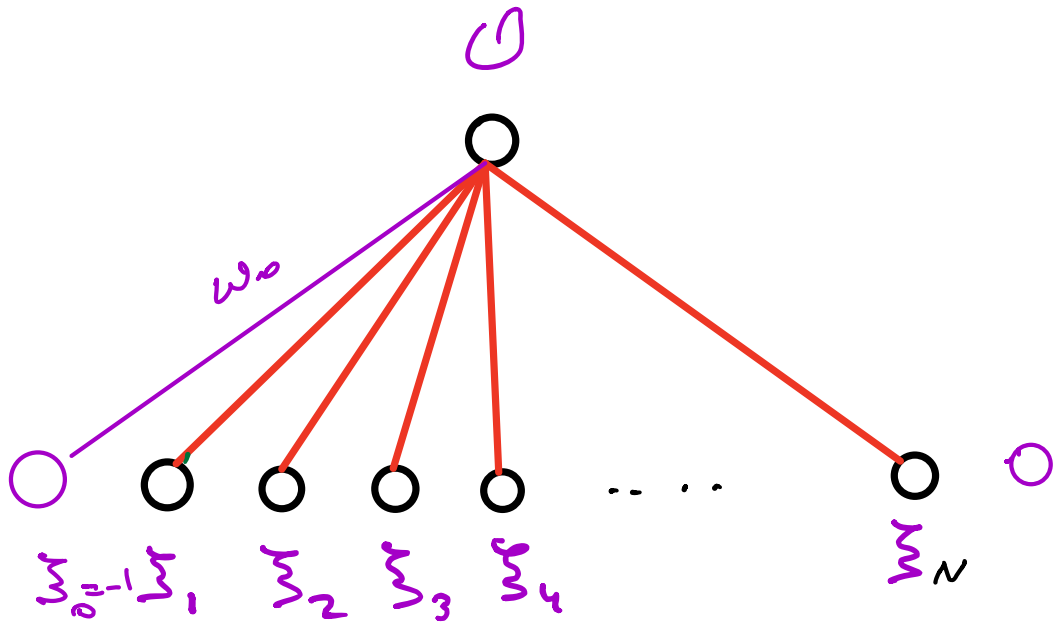
Noten que no se mezclan los M perceptrones simples.


v_1
w_1
w_2
w_3
w_4
w_5


v_2
w_1
w_2
w_3
w_4
w_5


v_m
w_1
w_2
w_3
w_4
w_5

Podemos entonces analizar un perceptron simple por neg



Suprimimos el indice del output.

$$O = g(h) = g\left(\sum_{k=1}^{N} w_k \xi_k\right)$$

donde g es la función activación y puede ser cualquiera de las que ya usamos, o alguna nueva.

Hemos omitido los umbrales. ¿Porque?

$$O = g\left(\sum_{k=1}^{N} w_k \xi_k - \theta\right)$$

$$= g\left(\sum_{k=0}^{N} w_k \xi_k\right)$$

si $w_0 = \theta$ y $\xi_0 = -1$ fijo.

O sea, el umbral se puede pensar como una variable de entrada fija en $\xi_0 = -1$ con $w_0 = \theta$. De esta forma todo a todos de igual manera.

Supongamos que tenemos un conjunto de entradas etiquetadas, tal que si entra $\vec{\xi}^{\mu}$ sabemos que debe

Dar $O = \xi^\mu$ $\quad \vec{\xi}^\mu \longrightarrow \xi^\mu$

$$O^\mu = g(h^\mu) = g\left(\sum_{k=0}^{N} w_k \xi_k^\mu\right)$$

Todo dependerá ahora del Tipo de funciones de salida, o sea, de cual es la función de activación $g(h)$

hetero-asociación: { la salida es de naturaleza diferente al input

EL CASO BINARIO

$$g(h) = \text{signo}(h) \quad \begin{cases} -1 \\ +1 \end{cases}$$

Suponemos que ξ_i^μ toman valores binarios

$$\xi_i^\mu = \pm 1$$

Suponemos que de los 2^N posibles inputs conocemos p resultados

$$\vec{\xi}^\mu \longrightarrow \zeta^\mu = \pm 1$$

$$(w_1, w_2, \ldots, w_N) = \vec{w}$$

$$\sum_k w_k \xi_k^\mu = \vec{w} \cdot \vec{\xi}^\mu$$

$$O^\mu = \text{signo}(h^\mu) = \zeta^\mu$$

$$\text{signo}\left(h^\mu\right) = \text{signo}\left(\sum_k w_k \xi_k^\mu\right) = \zeta^\mu$$

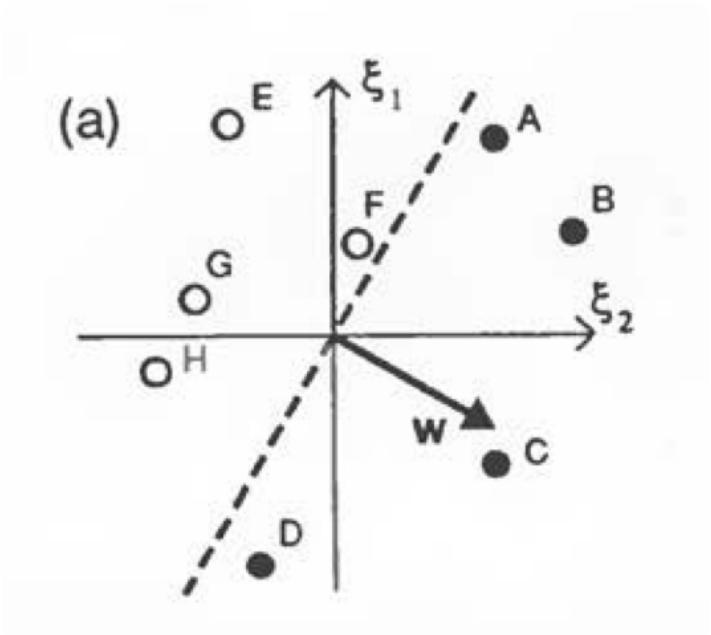
$$\underbrace{\text{signo}\left(\bar{w}\cdot\bar{\xi}^\mu\right) = \zeta^\mu}_{\text{lo deseado}}$$

Esto debe valer para los p valores de μ.

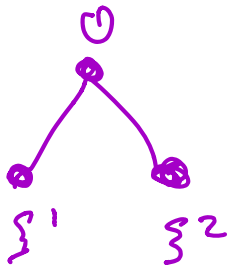
$\bar{w}\cdot\bar{\xi}^\mu$ es la proyección de $\bar{\xi}^\mu$ sobre $\bar{w}$

$$\bar{w}\cdot\bar{\xi}^\mu = |\bar{w}|\,|\bar{\xi}^\mu|\cdot\cos(\alpha)$$

donde α es el ángulo entre $\bar{w}$ y $\bar{\xi}^\mu$.



Debe existir un plano (en este caso una recta) que separe los ● de los ○ .



- ● significa output deseado $\zeta^\mu = +1$
- ○ significa output deseado $\zeta^\mu = -1$

Otra forma:

$$\text{signo}(\bar{w} \cdot \bar{\xi}^\mu) = \zeta^\mu$$

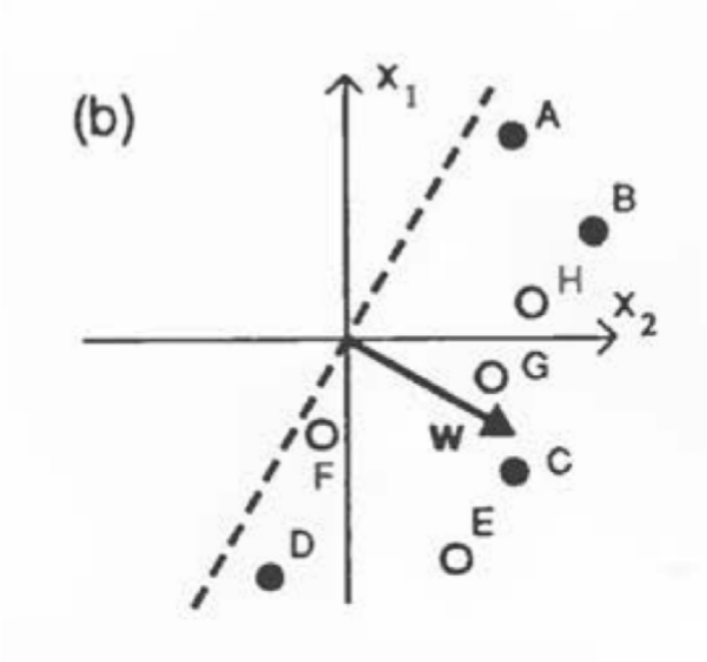
$$\zeta^\mu \cdot \text{signo}(\bar{w} \cdot \bar{\xi}^\mu) = \zeta^\mu \zeta^\mu = +1$$

$$\text{signo}\left(\zeta^\mu (\bar{w} \cdot \bar{\xi}^\mu)\right) = +1$$

$$\text{signo}\left(\bar{w} \cdot (\zeta^\mu \bar{\xi}^\mu)\right) = +1$$

$$\text{signo}\left(\bar{w} \cdot \bar{x}^\mu\right) = +1$$

$$\bar{w} \cdot \bar{x}^\mu > 0 \quad \text{(deseado)}$$



¿Qué pasa si no se pueden separar?
En este caso no se puede resolver.

Un problema de clasificación binaria es linealmente separable si existe un plano en el espacio Ξ que separe los casos $J^\mu = +1$ de los casos $J^\mu = -1$.

SEPARABILIDAD LINEAL

Hemos que para si incluimos explícitamente el umbral

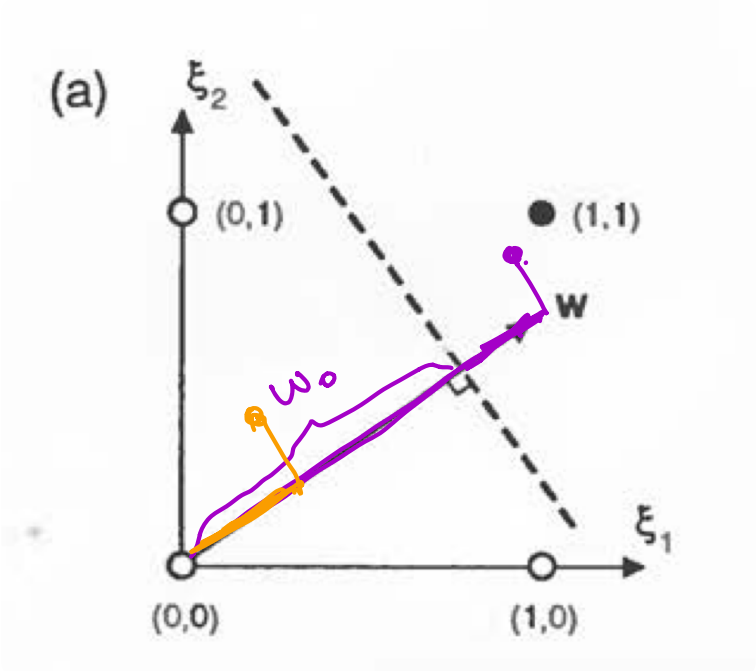
$$O = \text{sign}\left(\bar{w}.\bar{\xi} - w_0\right)$$

Las regiones con $\xi^\mu = +1$ y $\xi^\mu = -1$ deben separarse por un plano de dimensión N-1.

El plano se separa por

$$\bar{w}.\bar{\xi} = w_0$$

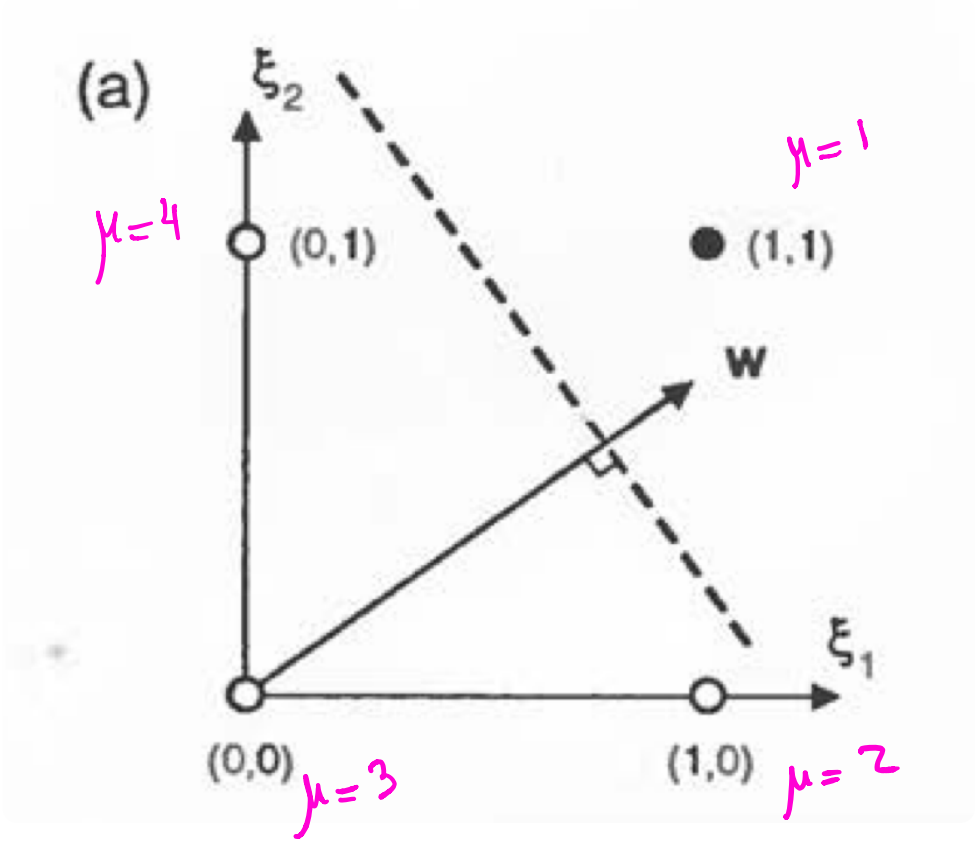
Signo $(\bar{w}\,\bar{\xi} - w_0)$

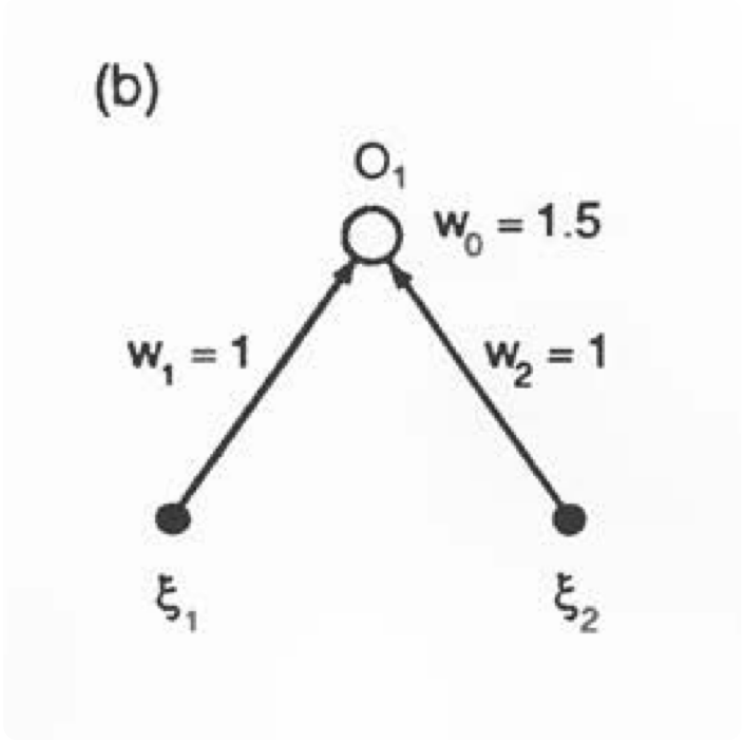


EJEMPLOS

LA FUNCIÓN "Y" O "AND".

μ	ξ_1^μ	ξ_2^μ	ζ^μ
1	1	1	+1
2	1	0	-1
3	0	1	-1
4	0	0	-1





$\mu = 1 \quad \xi_1 = 1 \quad \xi_2 = 1 \quad \zeta^1 = +1$

$$O = \text{sign}\left(w_1 \xi_1 + w_2 \xi_2 - 1.5\right)$$
$$= \text{sign}\left(1.1 + 1.1 - 1.5\right) = \text{sign}\left(2 - 1.5\right)$$
$$= \text{sign}(0.5) = +1 = \zeta^1$$

$\mu = 2 \quad \xi_1^2 = 1 \quad \xi_2^2 = 0 \quad \zeta^2 = -1$

$$O = \text{sign}\left(1.1 + 1.0 - 1.5\right) = \text{sign}\left(1 - 1.5\right)$$
$$= \text{sign}(-0.5) = -1 = \zeta^2$$

$\mu = 3 \quad \xi_1^3 = 0 \quad \xi_2^3 = -1 \quad \zeta^3 = -1$

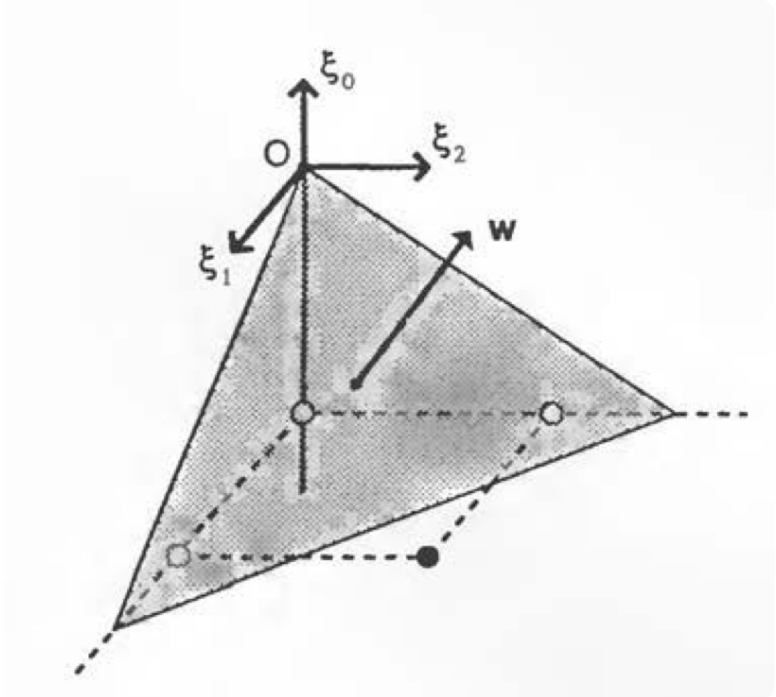
$0 = \text{sign}(0 \times 1 + 1.1 - 1.5) = \text{sign}(-0.5)$

$= -1 = \zeta^3$

$\mu = 4 \quad \xi_1^4 = 0 \quad \xi_2^4 = 0 \quad \zeta^4 = -1$

$0 = \text{sign}(0 \times 1 + 0 \times 1 - 1.5) = \text{sign}(-1.5)$

$= -1 = \zeta^4$



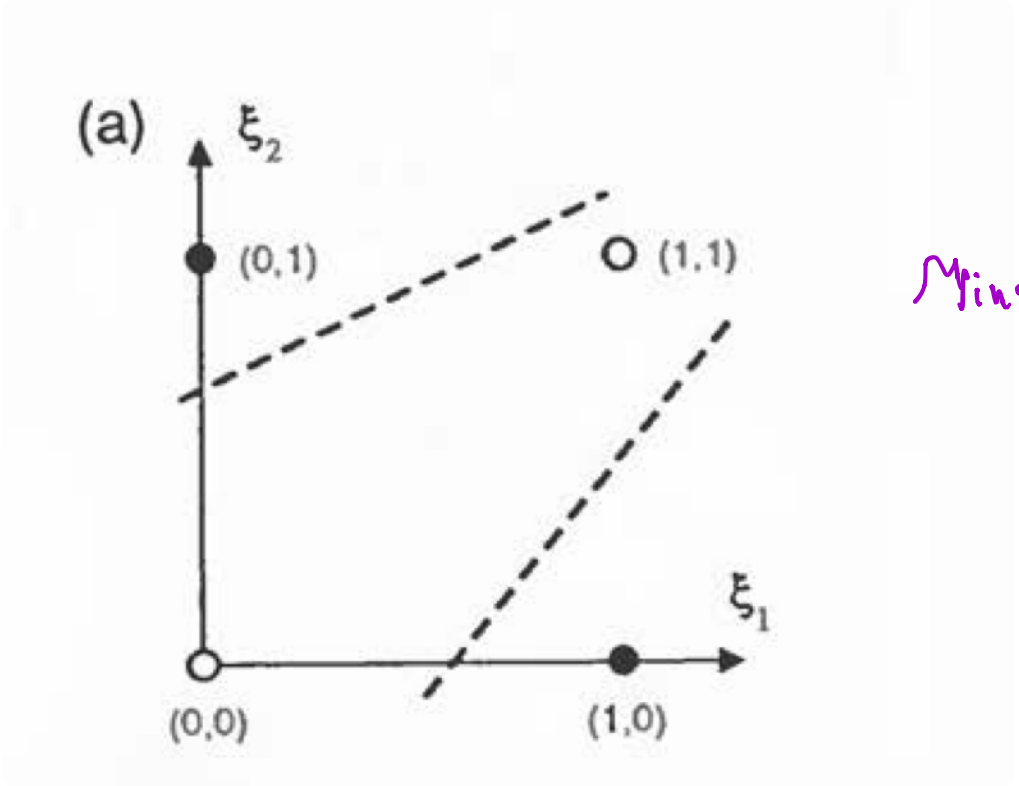
$\bar{w} = (w_0, w_1, w_2)$

La función XOR (OR exclusivo)

μ	ξ_1^μ	ξ_2^μ	ζ^μ
1	1	1	−1
2	1	0	+1
3	0	1	+1
4	0	0	−1

$(\xi^1\ \xi^2)$

$(\omega^1\ \omega^2)$



Minsky y Papert (1969)

¡No es linealmente separable!

Esto frenó el estudio de las redes neuronales.

El sistema a resolver es:

$$w_1 + w_2 - w_0 < 0 \Rightarrow w_1 + w_2 < w_0 \quad (i)$$

$$-w_1 - w_2 - w_0 < 0 \Rightarrow -w_1 - w_2 < w_0 \quad (ii)$$

$$w_1 - w_2 - w_0 > 0 \Rightarrow w_1 - w_2 > w_0 \quad (iii)$$

$$-w_1 + w_2 - w_0 > 0 \Rightarrow -w_1 + w_2 > w_0 \quad (iv)$$

De (i) y (iv) $w_1 < 0$

De (ii) y (iii) $w_1 > 0$

Esto muestra que no es posible encontrar w_1, w_2 y w_0 que resuelvan el problema.

UN ALGORITMO DE APRENDIZAJE

Comenzamos con valores aleatorios

w_k con $k = 1, 2, \ldots, N$.

$$w_{ik}^{\text{nuevo}} = w_{ik}^{\text{viejo}} + \Delta w_k$$

$$\Delta w_{ik} = \begin{cases} 2\eta \zeta_i^\mu \xi_k^\mu & \text{si } \zeta_i^\mu \neq O_i^\mu \\ 0 & \text{si } \zeta_i^\mu = O_i^\mu \end{cases}$$

ó

$$\Delta w_{ik} = \eta \left(1 - \zeta_i^\mu O_i^\mu\right) \zeta_i^\mu \xi_i^\mu$$

ó

$$\Delta w_{ik} = \eta \left(\zeta_i^\mu - O_i^\mu\right) \xi_i^\mu$$

η : learning rate.

Un proceso más "robusto" es:

$$\xi_i^\mu h_i^\mu \equiv \xi_i^\mu \sum_k w_{ik} \xi_k^\mu > N\kappa$$

$$\Delta w_{ik} = \eta\, \Theta\left(N\kappa - \xi_i^\mu h_i^\mu\right) \xi_i^\mu \xi_k^\mu$$

Θ : función escalón.

Este es la regla de aprendizaje del perceptron simple (Rosenblatt, 1962) para una única salida ($M=1$)

$$\vec{w} \cdot \vec{x}^\mu > N\kappa$$

Veamos el problema geométricamente

$$\Delta \bar{w} = \eta \ominus (\nu_k - \bar{w} \cdot \bar{x}^\mu)\bar{x}^\mu$$

O sea $\Delta \bar{w}$ apunta hacia $\bar{x}^\mu$ si

$$\nu_k < \bar{w}\bar{x}^\mu = |\bar{w}|\,|\bar{x}^\mu| \cos(\alpha)$$

$$\frac{\nu_k}{|\bar{w}|} < |\bar{x}^\mu| \cos(\alpha)$$: proyección de $\bar{x}^\mu$ sobre $\bar{w}$.

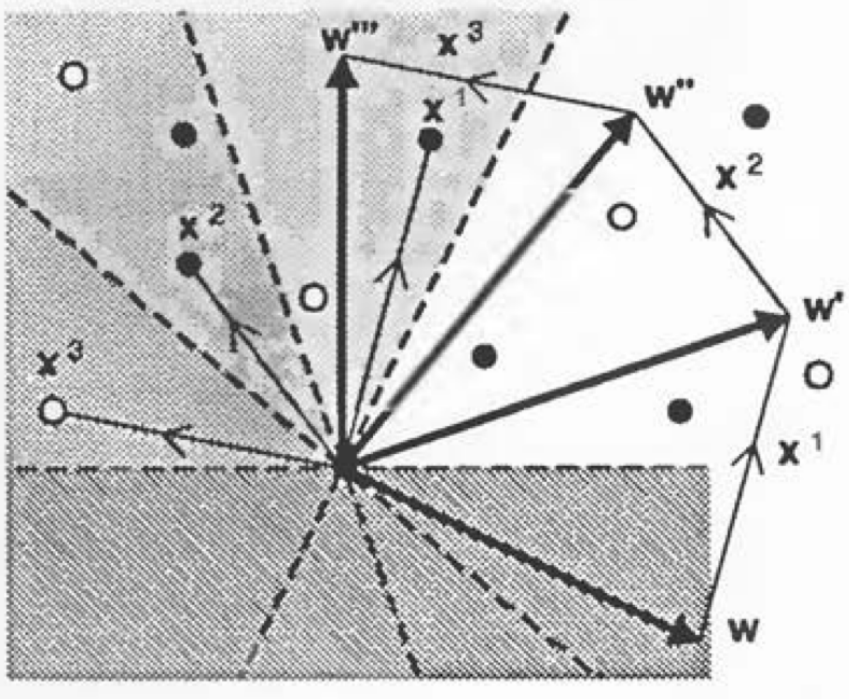
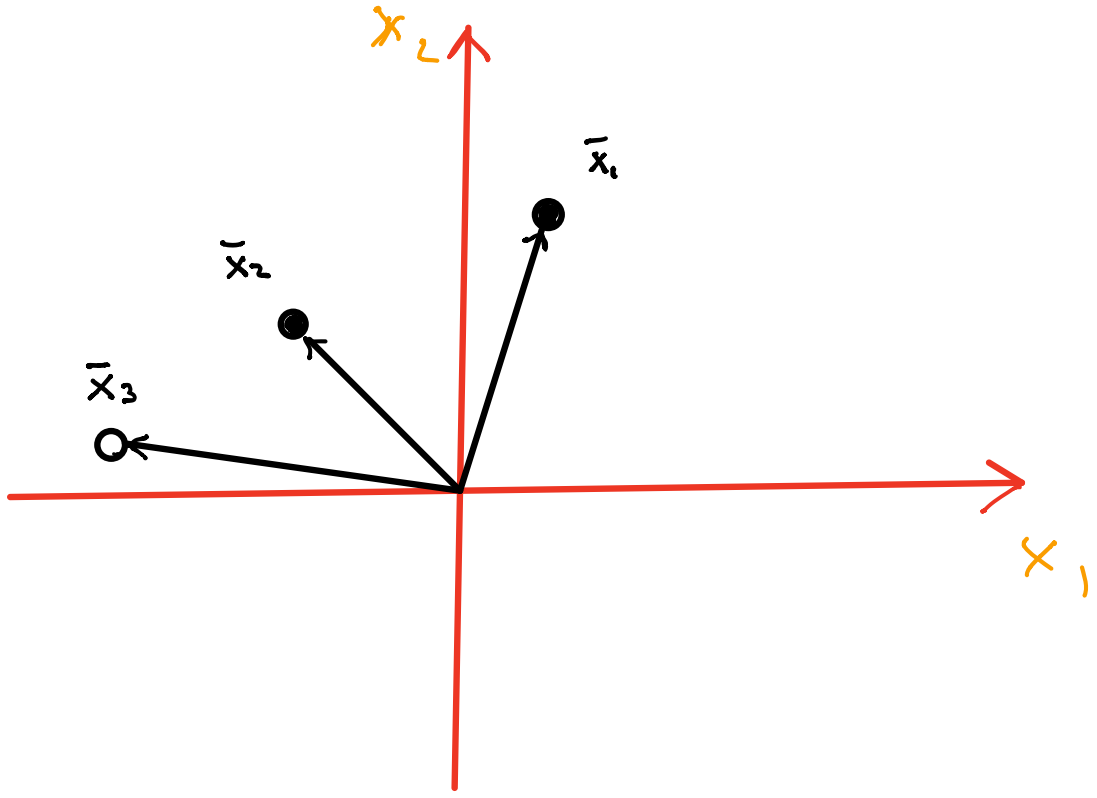


FIGURE 5.6 How the weight vector evolves during training for $\kappa = 0$, $\eta = 1$. Successive values of the weight vector are shown by $\mathbf{w}$, $\mathbf{w}'$, $\mathbf{w}''$, and $\mathbf{w}'''$. The darker and darker shading shows the "bad" region where $\mathbf{w} \cdot \mathbf{x} < 0$ for the successive $\mathbf{w}$ vectors. Each $\mathbf{w}$ is found from the previous one (e.g., $\mathbf{w}'$ from $\mathbf{w}$) by adding an $\mathbf{x}^\mu$ from the current bad region.

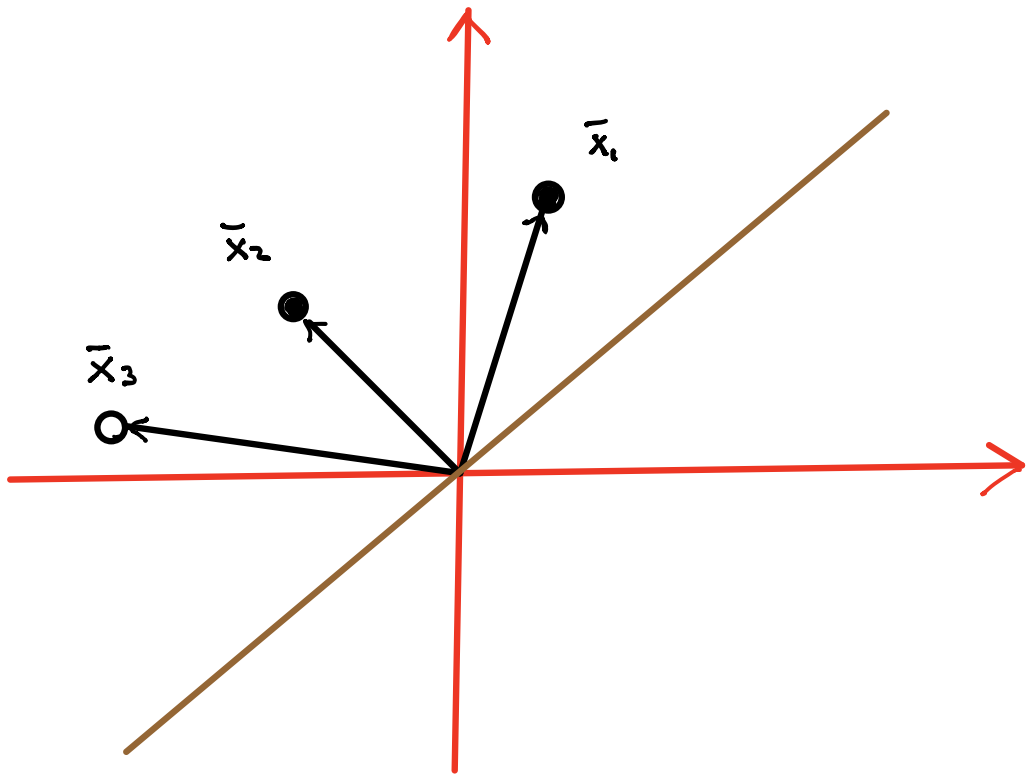
Ejemplo con $\kappa = 0$

$$D(\bar{w}) = \frac{1}{|\bar{w}|} \min_{\mu} (\bar{w} \cdot \bar{x}^{\mu})$$

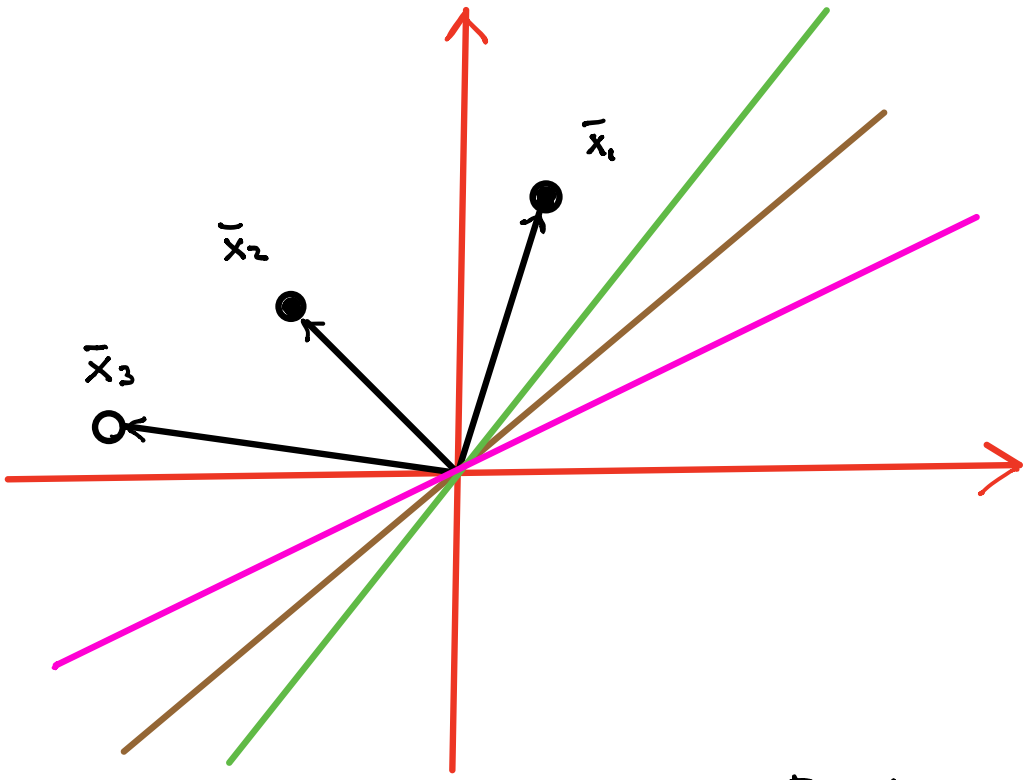
$$D_{max} \equiv \max_{\bar{w}} D(\bar{w})$$



Vemos que el problema admite solución ya que todos los puntos de entrenamiento quedan en un semi plano. Como $x=0$, la superficie que separa en 2 semi-planos debe pasar por el origen.

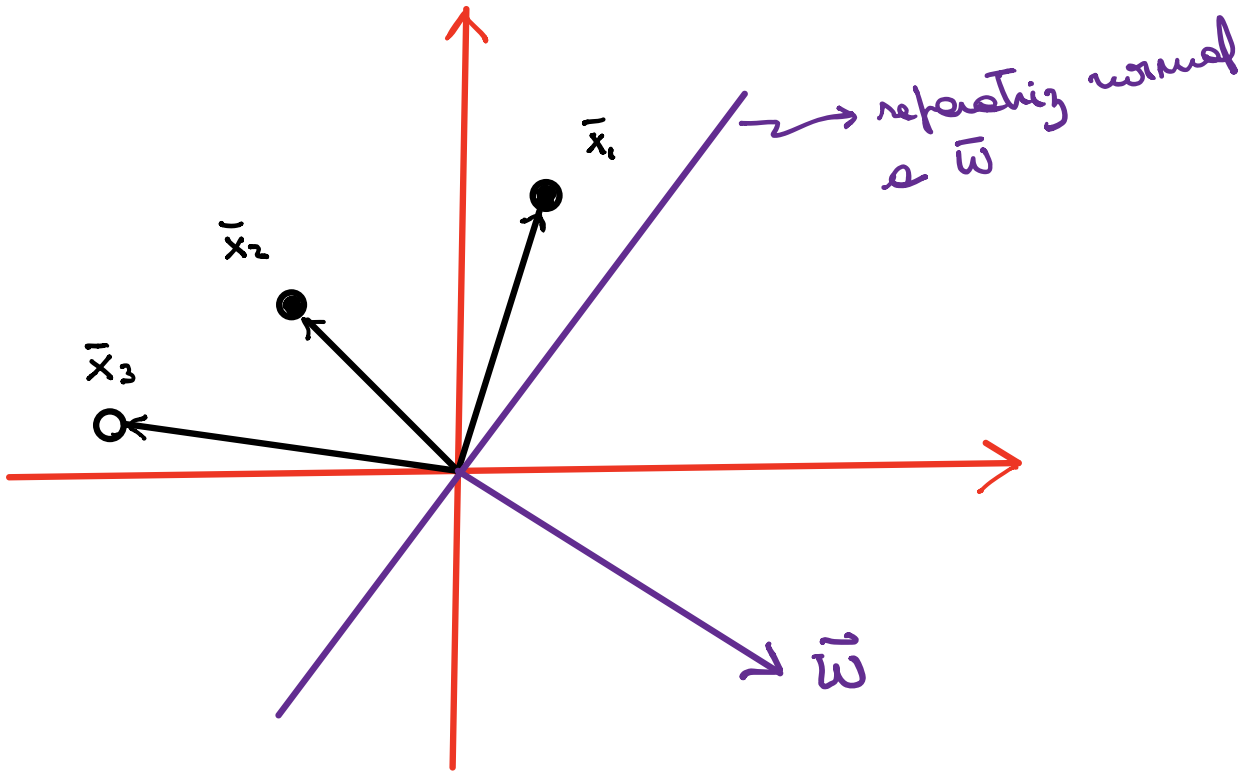


La línea marrón es una posible separatrix, pero hay muchas otras posibles. Este es un problema fácil.



Vemos que las líneas verde y rosa también son solución. Recuerden que $\bar{w}$ debe ser normal a la separatriz

Ahora elegimos un vector $\bar{w}$ inicial aleatorio y lo graficamos en el plano $\bar{x} = (x_1, x_2)$.



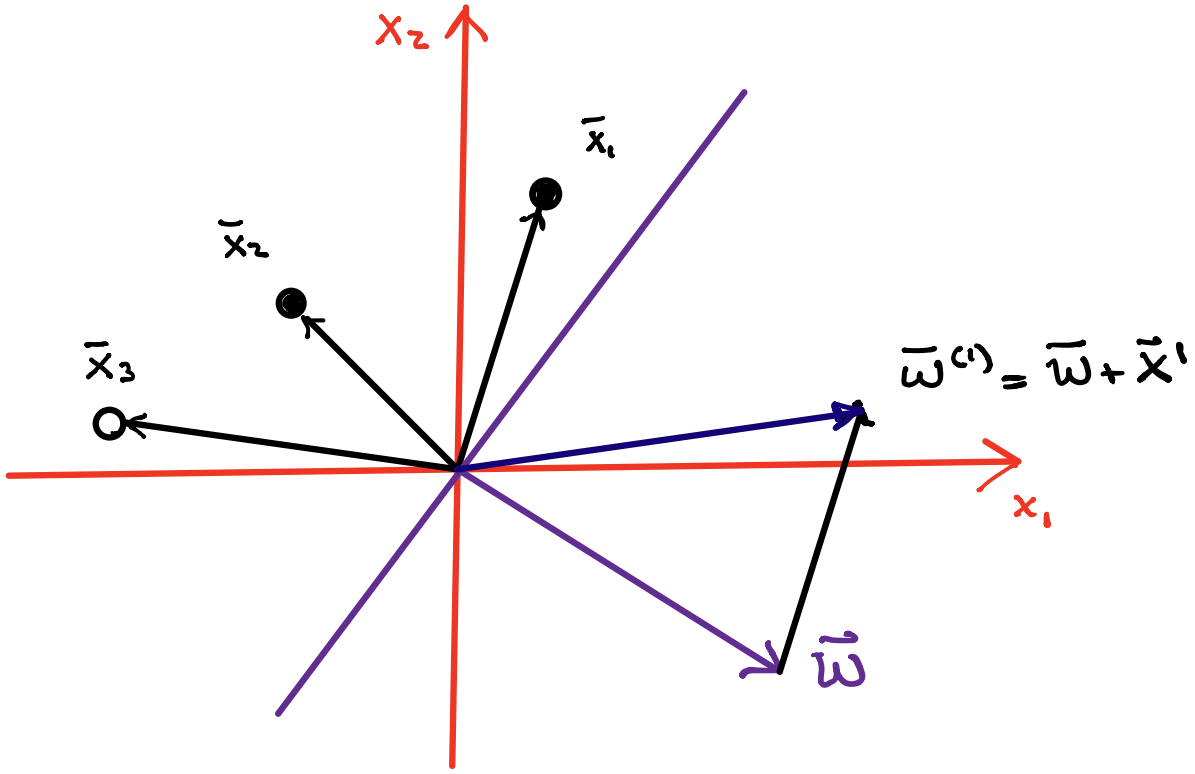
Vemos que este $\bar{w}$ no cumple la condición de tener proyección positiva con todos los $\bar{x}^{\mu}$ (en este caso $\mu = 1, 2$ y 3). Este $\bar{w}$ en particular es HORRIBLE, pues no tiene proyección positiva con ningún ejemplo

Ahora aplico la regla de aprendizaje con el ejemplo 1. Como la proyección con $\bar{x}^1$ es negativa

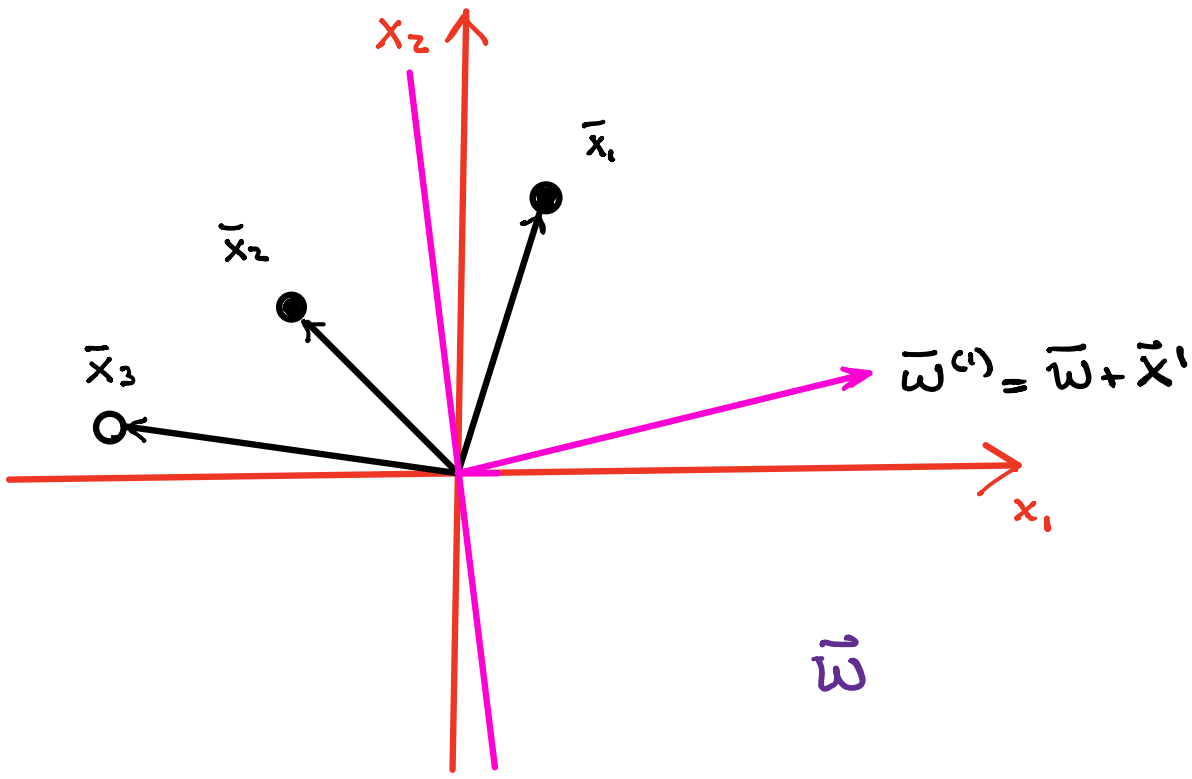
$$\bar{w} \longmapsto \bar{w} + \eta \bar{x}^1$$

Si hago $\eta = 1$

$$\bar{w} \longrightarrow \bar{w} + \bar{x}^1 = \bar{w}^{(1)}$$



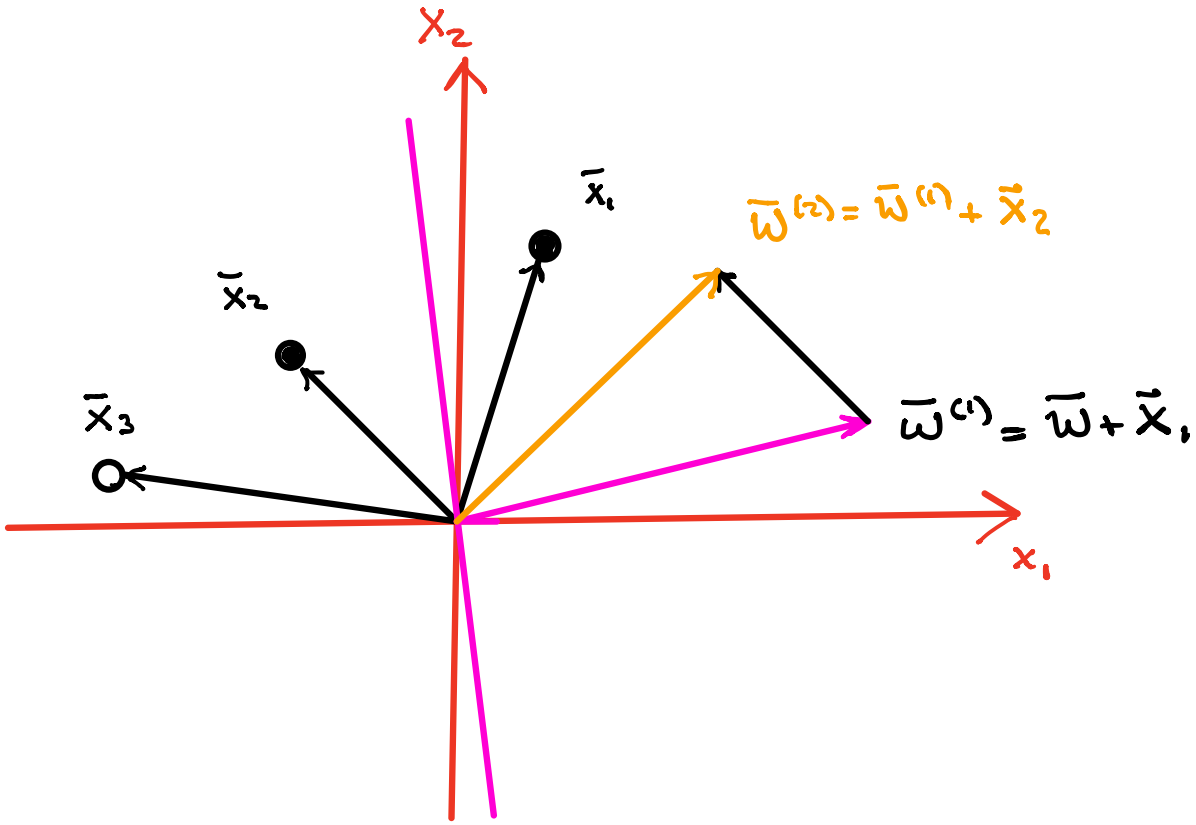
Bosquejo el $\bar{w}$ inicial y su recta normal para ejemplo



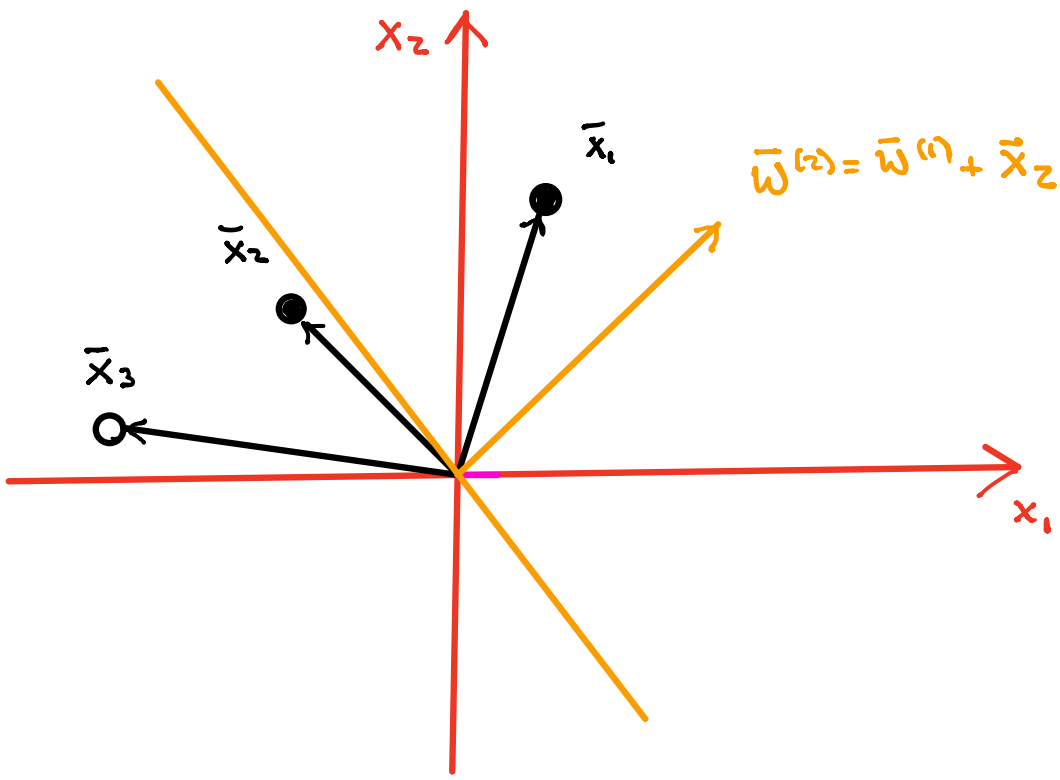
Noten que ahora el nuevo $\bar{w}^{(1)}$ tiene el menor proyección positiva con el ejemplo $x=1$

Ahora repitamos lo mismo para $x=2$.

$$\vec{w}^{(2)} = \vec{w}^{(1)} + \vec{x}_1$$



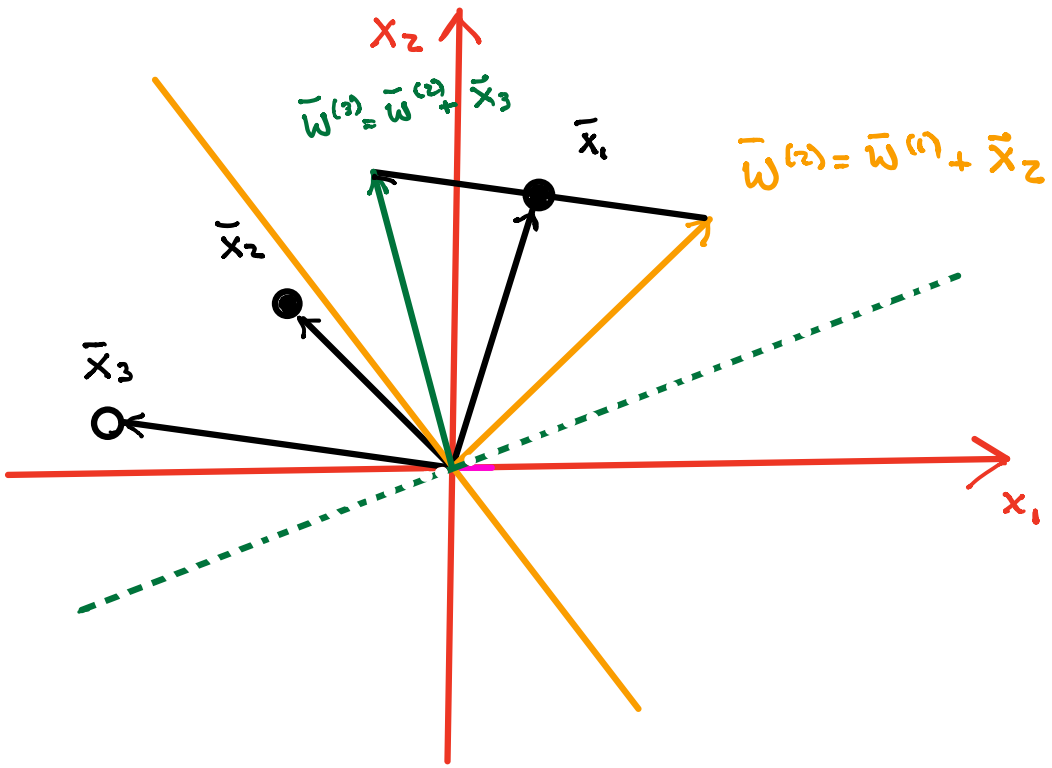
Bonemos lo superfluo:



Seguimos teniendo proyección positiva solo con $\mu=1$ pero estamos más cerca de resolver $\mu=2$ 

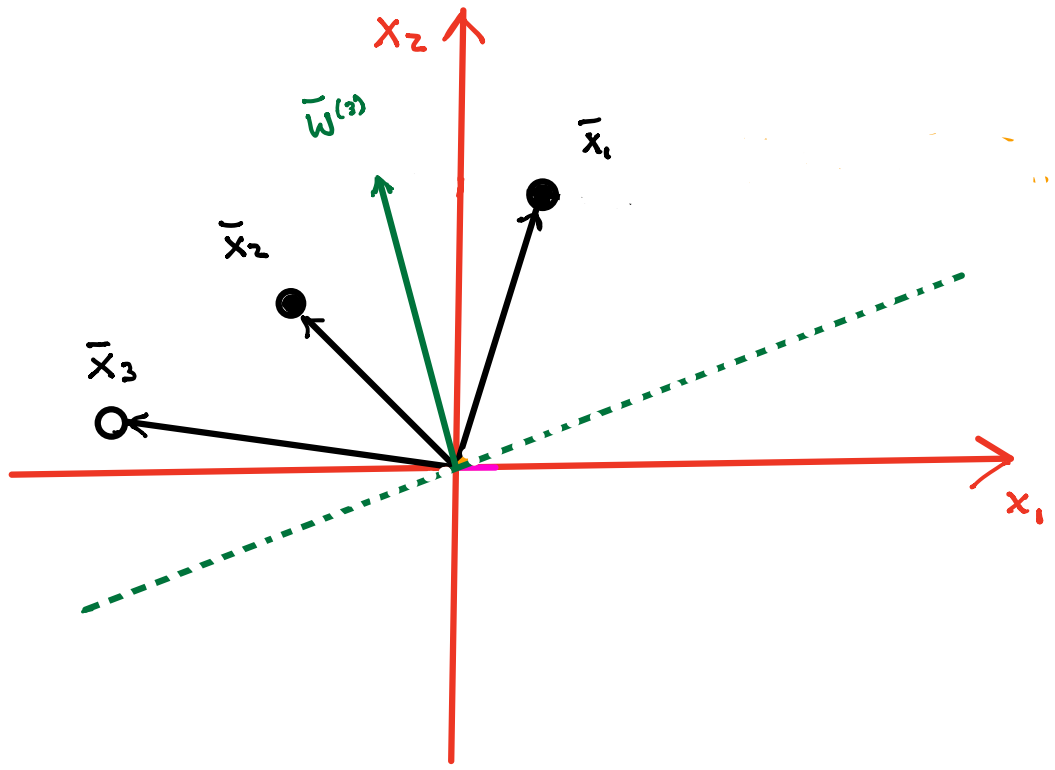
Repitamos para $\mu=3$.

$$\vec{w}^{(3)} = \vec{w}^{(2)} + \vec{x}^{3}$$



Ya resolvimos el problema en tres pasos, visitando una vez cada muestra. Noten que $\vec{w}^{(3)}$ define una recta perpendicular a él, que pasa por el origen y que todos los $\vec{x}_\mu$ están de un lado del plano (en un semiplano), y es el lado en el que vive $\vec{w}^{(3)}$.

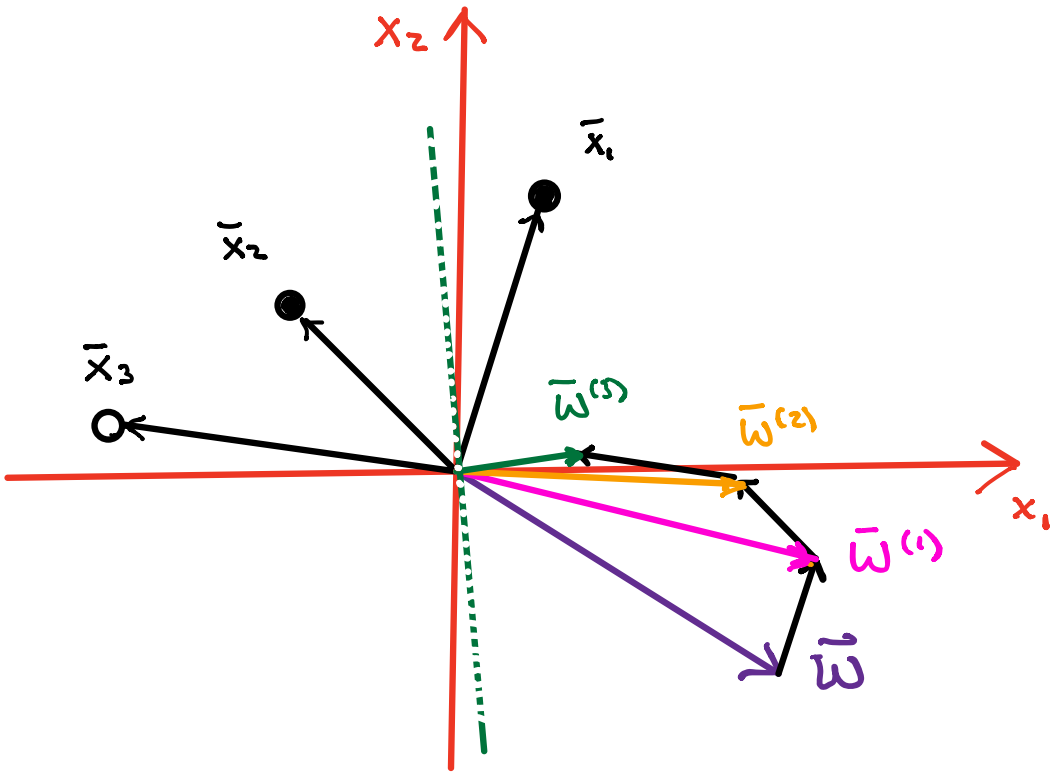
Esto quiere decir que $\bar{w}^{(3)}$ tiene proyección positiva con los tres vectores $\bar{x}_1$, $\bar{x}_2$ y $\bar{x}_3$



Este ejemplo es muy muy simple, pero los casos reales son mucho más complicados.

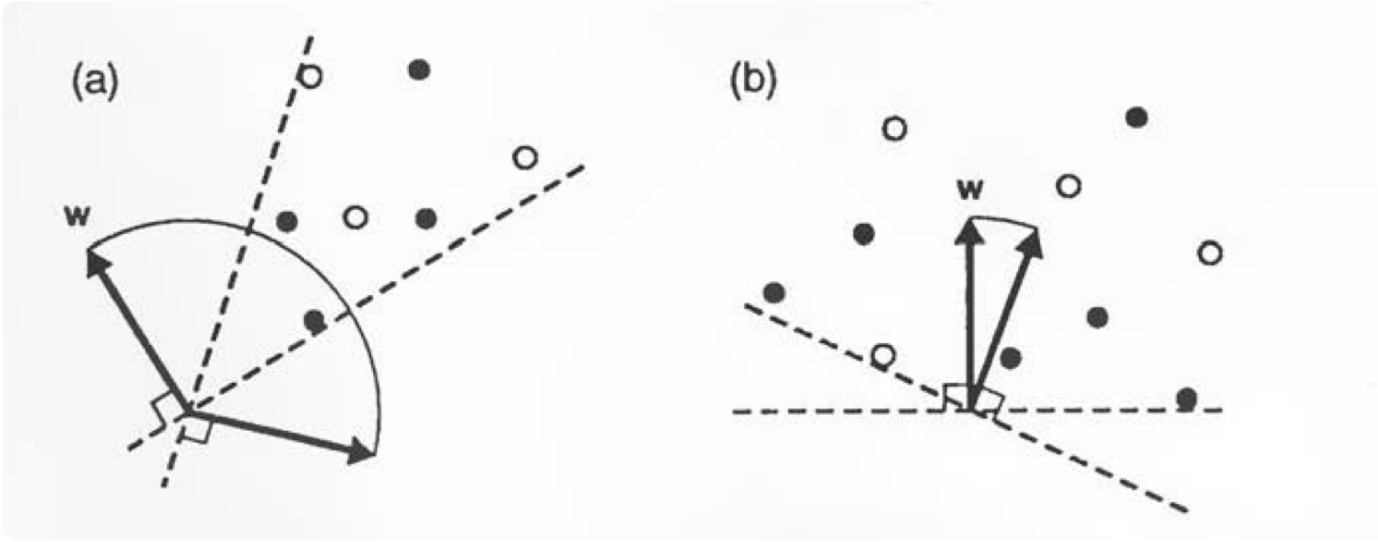
- casi nunca tenemos 2 variables de entrada sino muchas más.
- casi nunca tenemos 3 ejemplos etiquetados en el conjunto de entrenamiento, sino muchos más.
- casi nunca funciona usar $\eta = 1$, con lo cual todo se hace mucho mas lento

  Abajo muestro como hubiera sido el proceso si hubiésemos tomado $\eta = \frac{1}{2}$



Vemos que $\bar{w}^{(3)}$ obtenido con $\eta = \frac{1}{2}$ no resuelve pero vemos que si tenemos muchos ejemplos en η pequeño (¿cuán pequeño) es mucho mejor. Nuestro algoritmo dice que ahora debemos volver a repetir el mismo procedimiento pero solo para los ejemplos que aún no fueron resueltos (en este caso $\mu = 2$ y $\mu = 3$ pues $\mu = 1$ está resuelto con $\bar{w}^{(3)}$).

Vemos que en general hay problemas más fáciles que otros

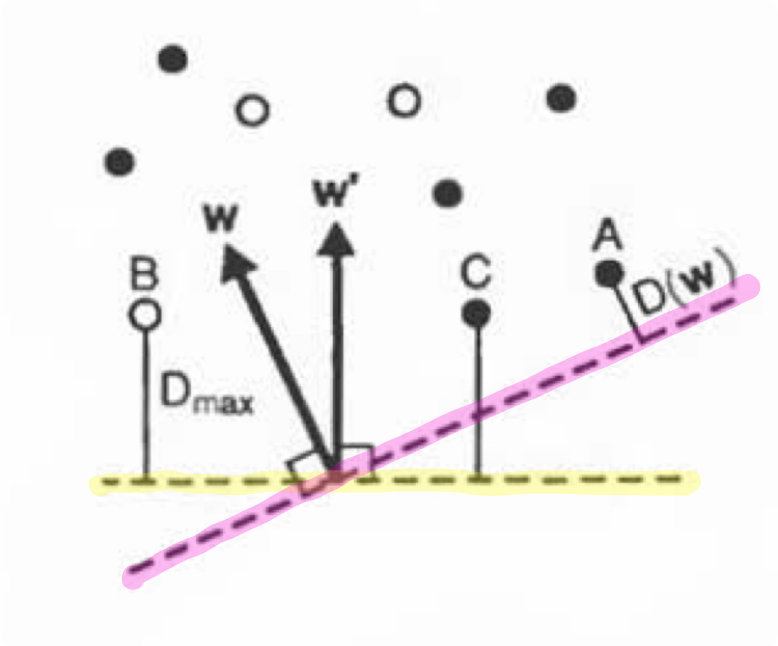


Discutamos porqué el ejemplo (a) es fácil y el (b) es difícil.

Notemos que no existe una única solución.

- si cambiamos el modulo del vector $\bar{w}$ que resuelve el problema, el mismo también lo resuelve. (no podemos cambiar la dirección).

- además de cambiar el módulo tenemos cierto margen de variación para elegir $\bar{w}$.



Vemos que la separating ensemble es más robusta que la rosa, y por ende preferimos $\bar{w}'$ a $\bar{w}$. Tomaremos como "OPTIMA" aquella que nos de el mayor $D(\bar{w})$. Recuerden que

$$D(\bar{w}) = \frac{1}{|\bar{w}|} \min_{\mu} (\bar{w} \cdot \bar{x}^{\mu})$$

$$D_{max} \equiv \max_{\bar{w}} D(\bar{w})$$

Pero, ¿cómo sabemos que encontraremos una solución? Gracias a la matemática

Teorema de convergencia de la regla del perceptron

El teorema tiene una formalización compleja para este curso. Básicamente establece que el algoritmo de la regla de aprendizaje converge a una solución en un número finito de pasos, suponiendo que tenemos un conjunto de datos de entrenamiento linealmente separables.

No lo demostraremos aquí.

Si tenemos suerte, este ω resolverá también casos que no estaban en el conjunto de entrenamiento. Esto significa que puede generalizar a partir de los ejemplos aprendidos.

Resumen hasta ahí.

- Consideramos un perceptron de una capa de M neuronas y lo reducimos en M perceptrons simples (con una única neurona de salida, o sea, $M=1$).
- Suponemos que la función de activación es la función signo (o sea $\theta = \pm 1$) y no nos preocupamos por las entradas.
- Vimos que el conjunto de datos de entrenamiento deben ser LINEALMENTE SEPARABLE.
- Si los datos son linealmente separables existe un algoritmo de aprendizaje que nos da una solución posible en un número finito de interacciones.

Nosotros podemos fijar el valor de M y el orden en que visitamos los ejemplos.

Combinando M perceptrons simples podemos hacer un perceptron de N entradas y M salidas.