REDES NEURONALES 2024

Clase 15 parte 3

Lunes 7 de octubre 2024

FAMAF, UNIVERSIDAD NACIONAL DE CÓRDOBA

INSTITUTO DE FÍSICA ENRIQUE GAVIOLA (UNC-CONICET)

Hasta aquí vimos bajo cuáles condiciones existe una solución al problema de clasificación, pero no tenemos ninguna regla para construir un vector sináptico $\overline{w}$ que resuelva el problema.

Y si esto lo debemos trabajar con un número grande de entradas, como sucede en la vida real y con un número muy grande de elementos en el conjunto de entrenemiento, el problema se torna inviable si no lo hacemos de forma algorítmica.

A continuación presentaremos el primer y más simple algoritmo de aprendizaje de este curso y de la historia de la Inteligencia Artificial.

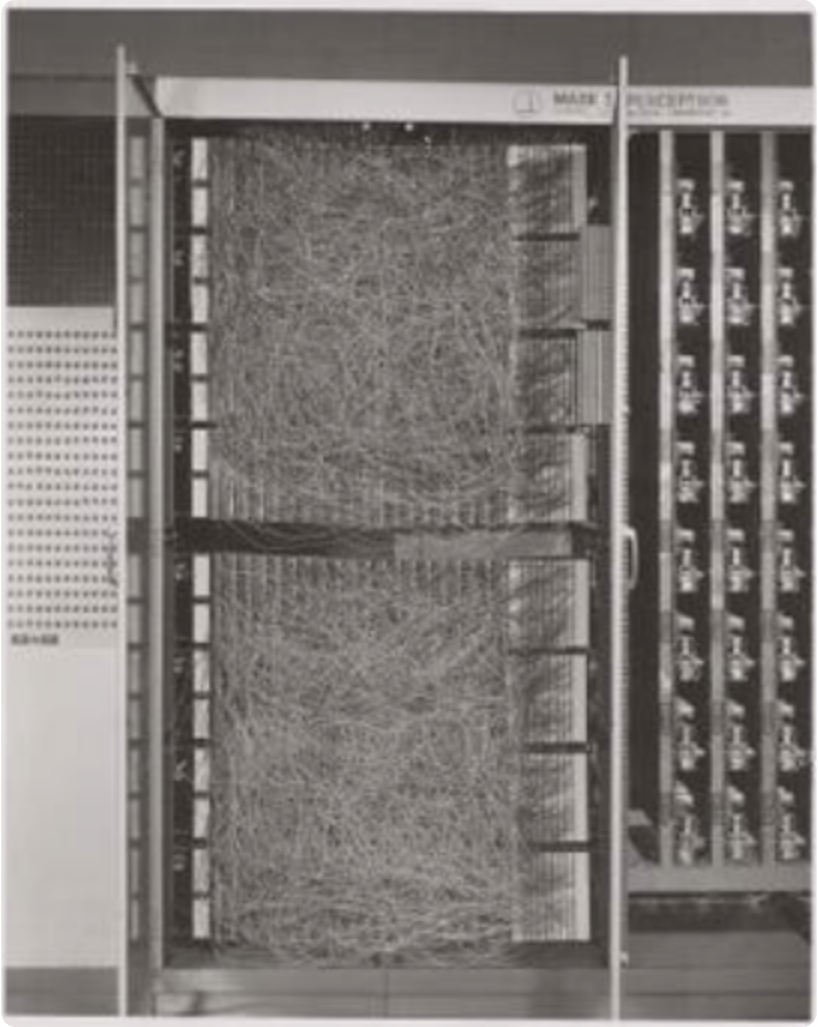
La idea de una neurona artificial matemática, recordemos, había sido introducida por *McCulloch and Pitts* en 1943, inspirados en los trabajos pioneros sobre el funcionamiento de las neuronas de Santiago Ramón y Cajal y Charles Scott Sherrington. Aquí aparece otra persona central de la historia de la inteligencia artificial llamada Frank Rosenblatt. Frank era un joven formado en ciencias que desde temprano se orientó al estudio de la psicología cognitiva. Fue contratado en el Cornell Aeronautical Laboratory en la ciudad de Buffalo, donde haría sus grandes contribuciones. Pero a nosotros nos interesa el desarrollo del primer algoritmo para encontrar automáticamente los pesos del modelo de McCulloch y Pitts. Este fue el punto inicial del Aprendizaje Automático.

Veremos que Rosenblatt generó el primer programa convencional que podía aprender iterativamente los pesos sinápticos de un perceptron simple por medio de la realización de pruebas y las evaluaciones de los correspondientes errores, como hacemos los humanos y como hacemos los científicos naturales.

Primero implementó el algoritmo por software en una IBM 704, la primera computadora comercial de la historia de la cual se vendieron apenas 120 ejemplares. Esta computadora hacia solo 40000 operaciones de punto flotante (de números reales) por segundo. Hoy un procesador I7 realiza 100.000,000.000 operaciones por segundo, o sea, poco más de dos millones de veces más rápido y mide unos pocos milímetros cuadrados.

Rosenblatt no se conformó con desarrollar el algoritmo que veremos hoy. Él implementó el algortimo físicamente en una máquina llamada **MARK I PERCEPTRON**, dedicada exclusivamente a clasificar imágenes de 400 píxeles (20 x 20). En particular, se eligió clasificar imágenes de hombres y mujeres.

En 1958 Rosenblatt escribió en un artículo de opinión en el New York Times en el cual afirmaba: *“Este es el embrión de una nueva computadora electrónica que espera poder caminar, hablar, ver, escribir, reproducirse y tener consciencia de su existencia”*. Como es de imaginar, le llovieron las burlas. Hoy sabemos que él estaba creando las bases de la algorítmica del actual Deep Learning.



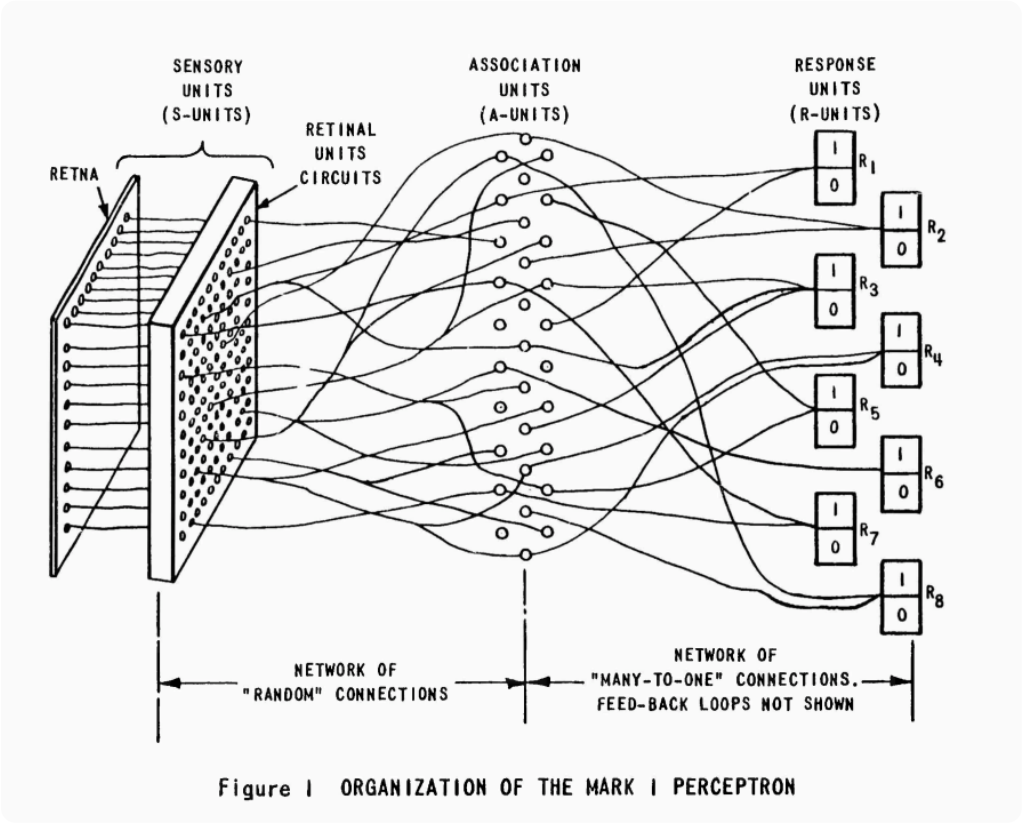


Figure 1 ORGANIZATION OF THE MARK 1 PERCEPTRON

LA REGLA DE APRENDIZAJE DEL PERCEPTRON

Veamos entonces el primer algoritmo de aprendizaje de máquina.

Supongamos que tenemos un **PERCEPTRÓN** simple con N entradas y una neurona de salida binaria.

Supongamos que tenemos p entradas etiquetadas correctamente, a las cuales le asignamos la salida (el valor de la neurona) correcta.

$$\vec{\xi}^{\,\alpha} \longrightarrow \zeta^{\alpha} = \pm 1 \quad (\alpha = 1, 2, \dots, p)$$

Finalmente, supongamos que el problema es linealmente separable. Nos preguntamos entonces

¿Cómo encontrar un $\vec{W}$ adecuado?

Elegimos, ante la falta de mejor criterio, los parámetros iniciales de nuestro perceptrón simple (las $N+1$ componentes de $\boldsymbol{W}$) de forma aleatoria. Por ejemplo, podemos elegir cada componente $\boldsymbol{W}$ en forma independiente entre sí y con distribución gaussiana centrada en cero y con desviación 1.

La idea central del algoritmo consiste en presentar sucesivamente a la red cada elemento del conjunto de entrenamiento, ya sea en un orden pre-establecido o aleatorio, y, dependiendo de que si el resultado es correcto o no, modificar levemente los parámetros.

$$O^{\alpha} = g(h^{\alpha})$$

Si $O^\alpha = \zeta^\alpha$ no cambiamos nada, pero

si $O^\alpha = -\zeta^\alpha$ castigamos al perceptrón cambiando levemente todos los acoplamientos.

$$W_k^{\text{nuevo}} = W_k^{\text{viejo}} + \Delta W_k^\alpha$$

$$\Delta W_k^\alpha = \begin{cases} 2 \cdot \zeta^\alpha \xi_k^\alpha \eta & O^\alpha \neq \zeta^\alpha \\ 0 & O^\alpha = \zeta^\alpha \end{cases}$$

$$\Delta W_k^\alpha = \eta (1 - \zeta^\alpha O^\alpha) \xi_k^\alpha \zeta^\alpha$$

$$= \eta (\zeta^\alpha - \zeta^{\alpha 2} O^\alpha) \xi^\alpha = \eta (\zeta^\alpha - O^\alpha) \xi^\alpha$$

$$\overline{W}^{\text{nuevo}} = \overline{W}^{\text{viejo}} + \Delta \overline{W}^\alpha$$

Y

En forma vectorial: $\Delta \overline{W}^\alpha = 2 \eta \zeta^\alpha \overline{\xi}^\alpha$ si $\zeta^\alpha \neq O^\alpha$

Si la respuesta es la equivocada, llevamos la componente *k* del vector **W** hacia el vector $\overline{\xi}^\alpha$ si $\zeta^\alpha = 1$, y hacia $-\overline{\xi}^\alpha$ si $\zeta^\alpha = -1$.

El parámetro η se llama *razón de aprendizaje* y hoy lo presentamos para que siempre nos acompañe. Él regulará, como veremos, la velocidad de aprendizaje y deberemos invertir mucho esfuerzo, en cada caso, para asignarle un valor adecuado.

Es un *hiperparámetro*, pues una vez determinado, en general (no siempre) estará fijo durante el aprendizaje. Para llegar al aprendizaje profundo actual, fue preciso invertir enormes esfuerzos en entender el sentido y función de este hiperparámetro. Ya lo veremos.

Recordemos la condición de estabilidad que debe cumplir el conjunto de pesos sinápticos:

$$O^\alpha = \text{signo}(h^\alpha) = \xi^\alpha$$

Lo que sale — Lo que quiero que salga

$$\xi^\alpha \,\text{signo}(h^\alpha) = \xi^\alpha \xi^\alpha = 1$$

$$\text{signo}(\xi^\alpha h^\alpha) = 1$$

$$\xi^\alpha h^\alpha > 0$$

{ con esto alcanzaría

Ya tenemos una regla de aprendizaje automático que encontrará una buena solución (no necesariamente la mejor) en un tiempo finito siempre y cuando el conjunto de entrenamiento sea linealmente separable.

Un pseudo código

Entrada: η y una secuencia de ejemplos de entrenamiento, o sea, de entradas $\bar{\xi}^\alpha$ y sus correspondientes etiquetas ζ^α correctas:

$$C\left(\{\bar{\xi}^\alpha, \zeta^\alpha\}\right) \quad \alpha = 1, 2, \ldots, p$$

Código: Inicializamos $\bar{W}_0$ aleatorio (en R^n)

Para cada ejemplo α $(\bar{\xi}^\alpha, \zeta^\alpha)$ en el conjunto de entrenamiento:

$$O^\alpha = \text{signo}(\bar{\xi} \cdot \bar{W})$$

Si $O^\alpha \neq \zeta^\alpha$

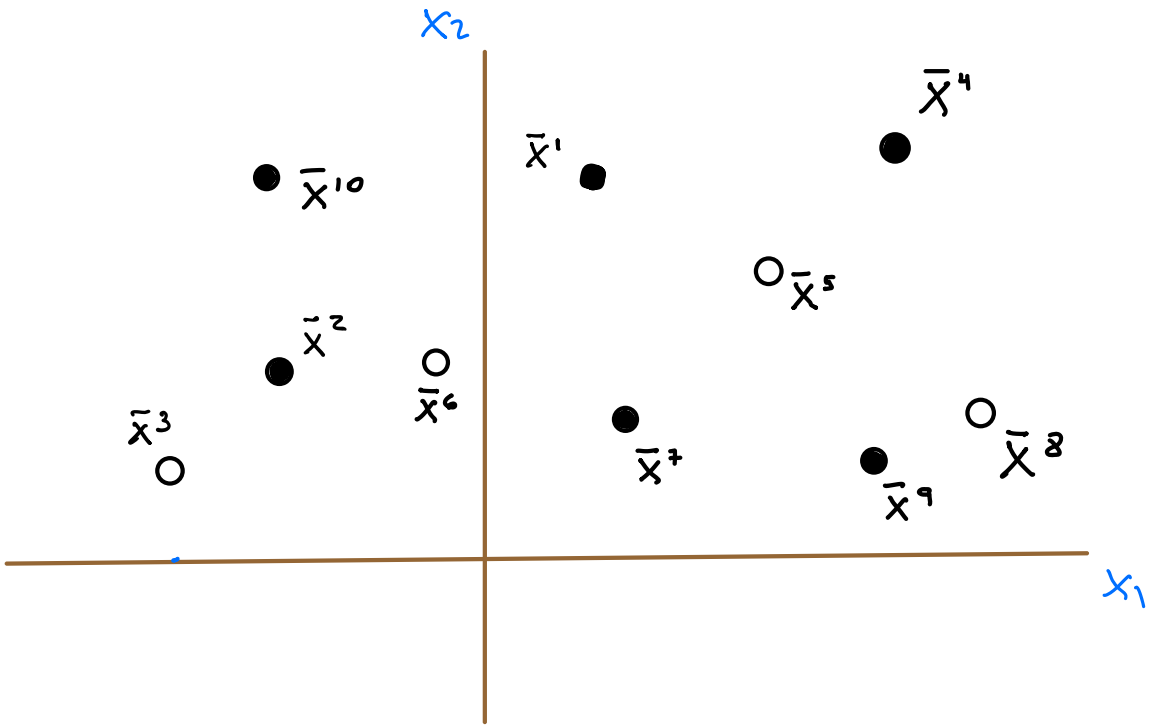
$$\Delta\bar{\omega} = 2\eta\,\zeta^\alpha \bar{\xi}^\alpha$$

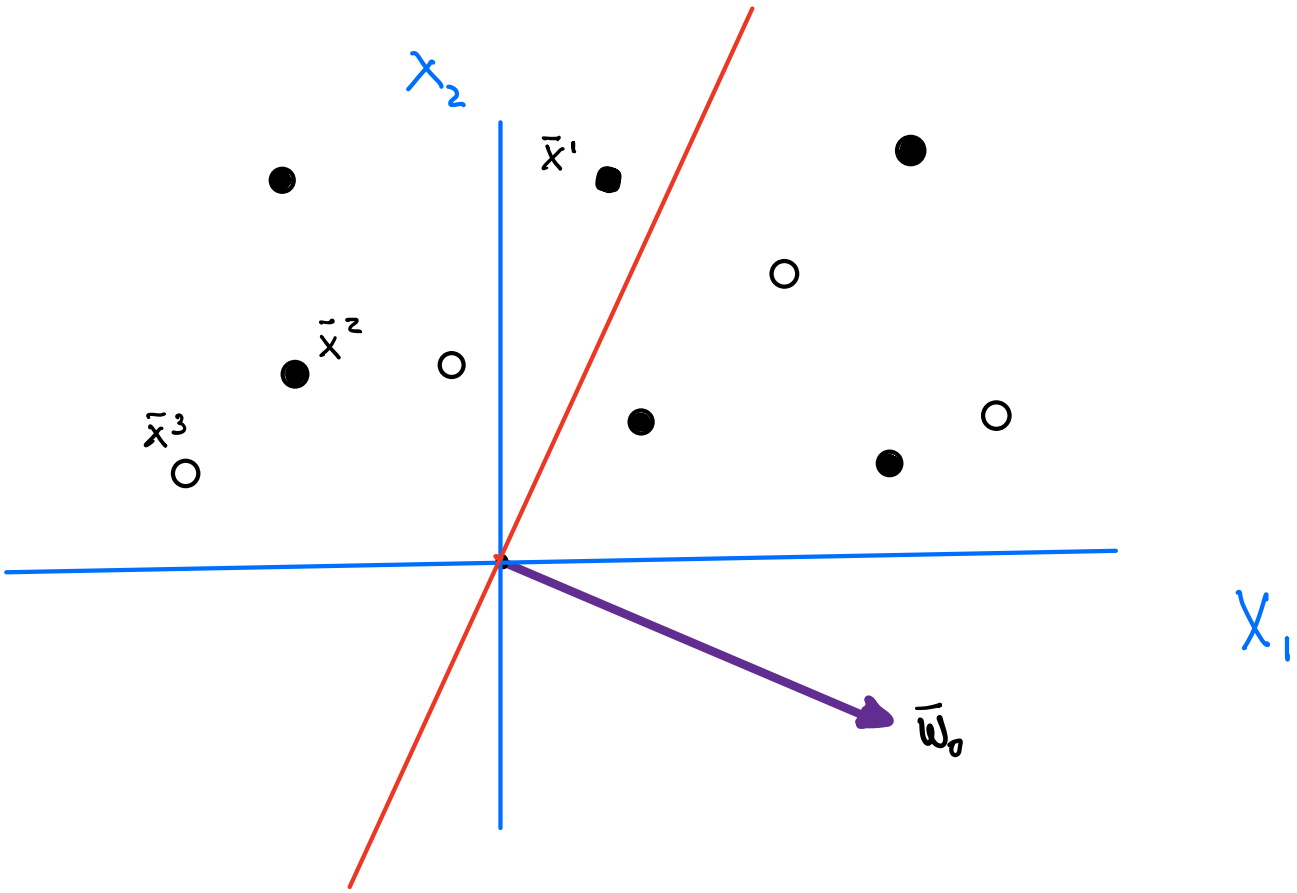
$$\bar{W}_{t+1} \longleftarrow \bar{W}_t + \Delta\bar{\omega}$$

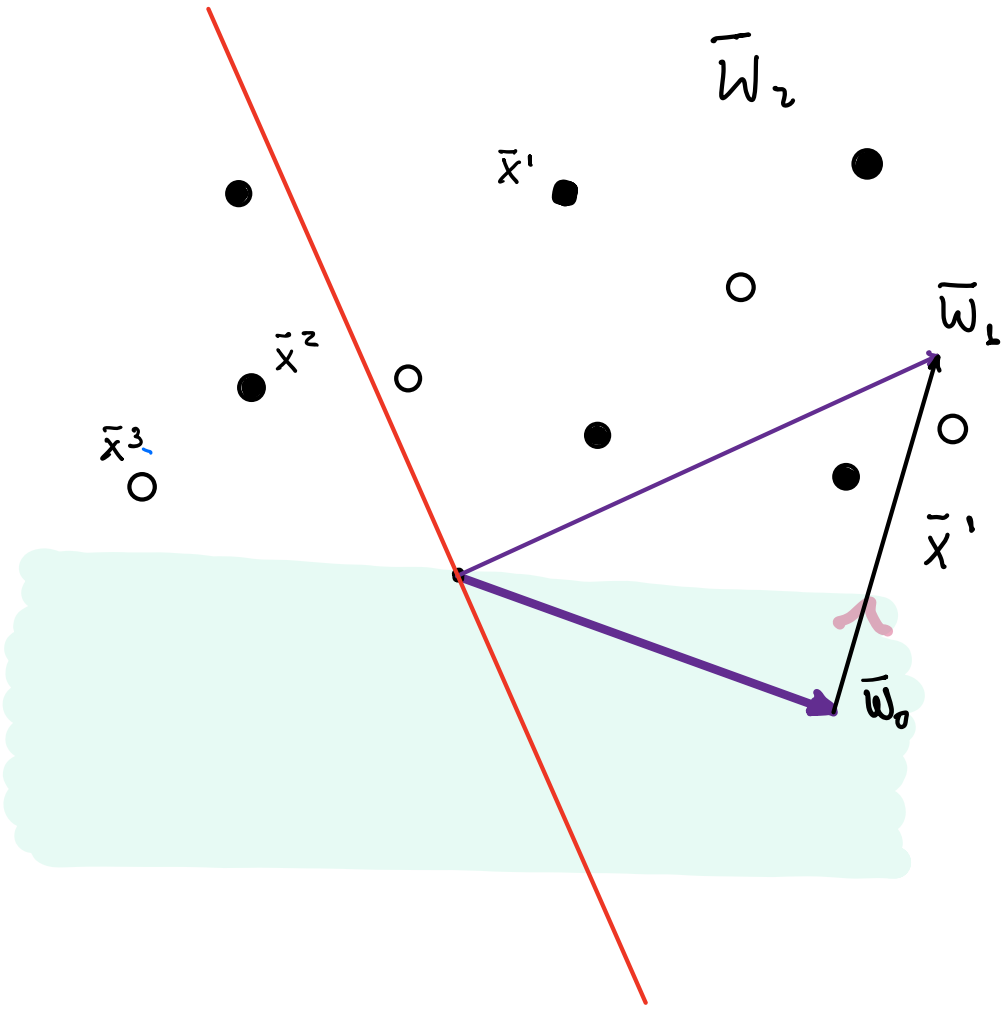
Salida: el vector final de acoplamiento $\bar{W}_f$.

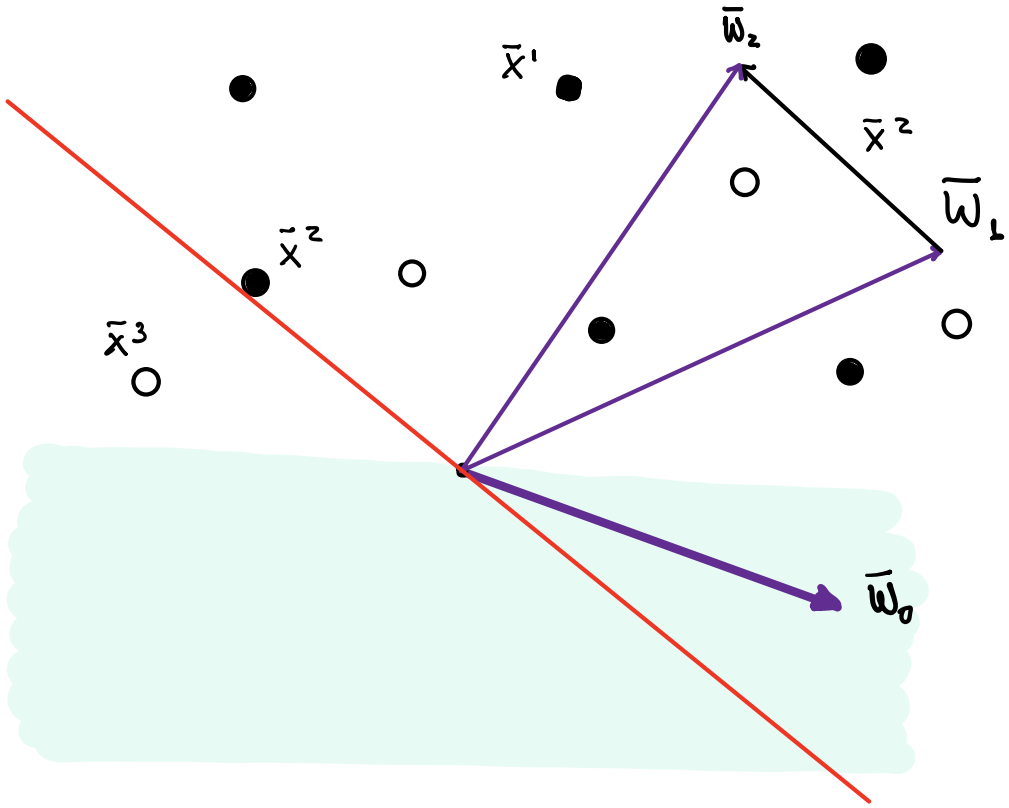
Miremos cómo funciona el algoritmo:

$$\eta = \frac{1}{2}$$

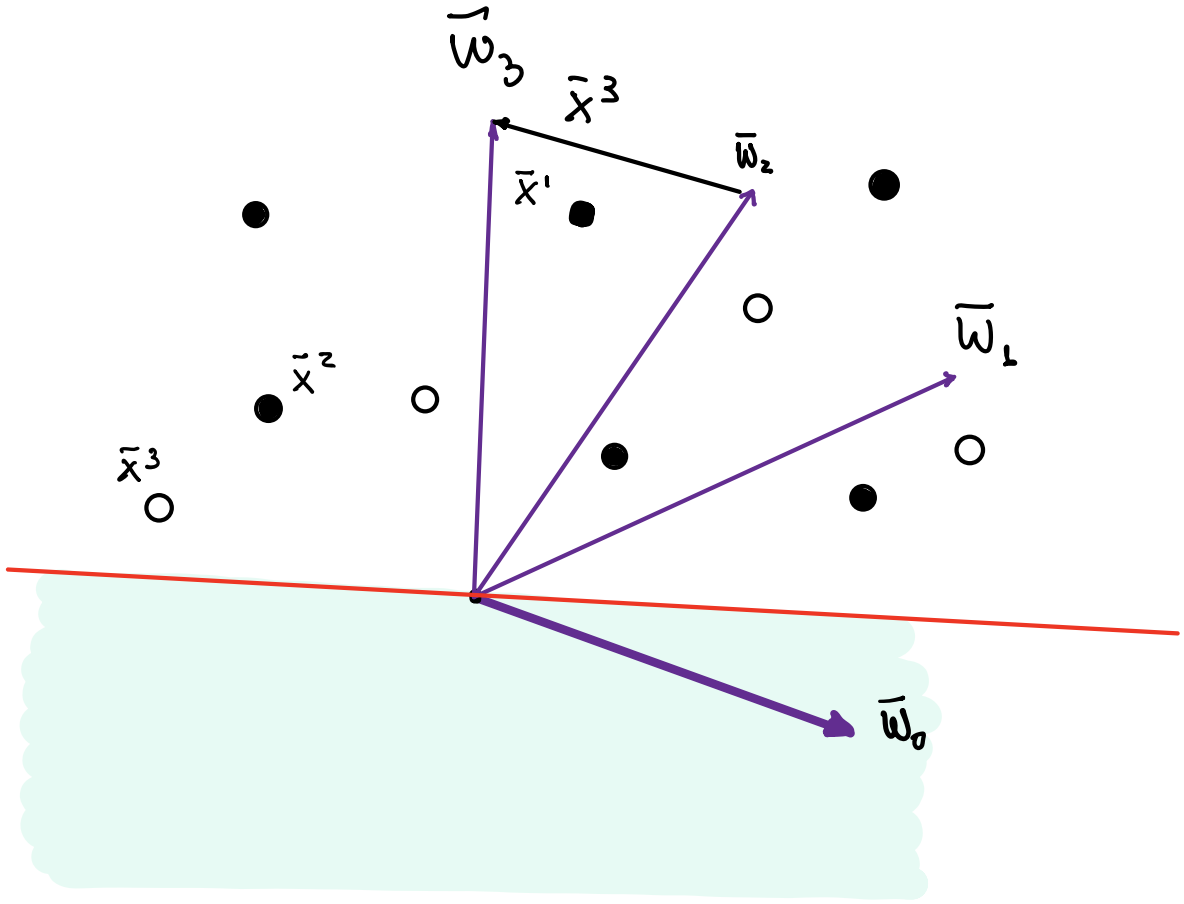


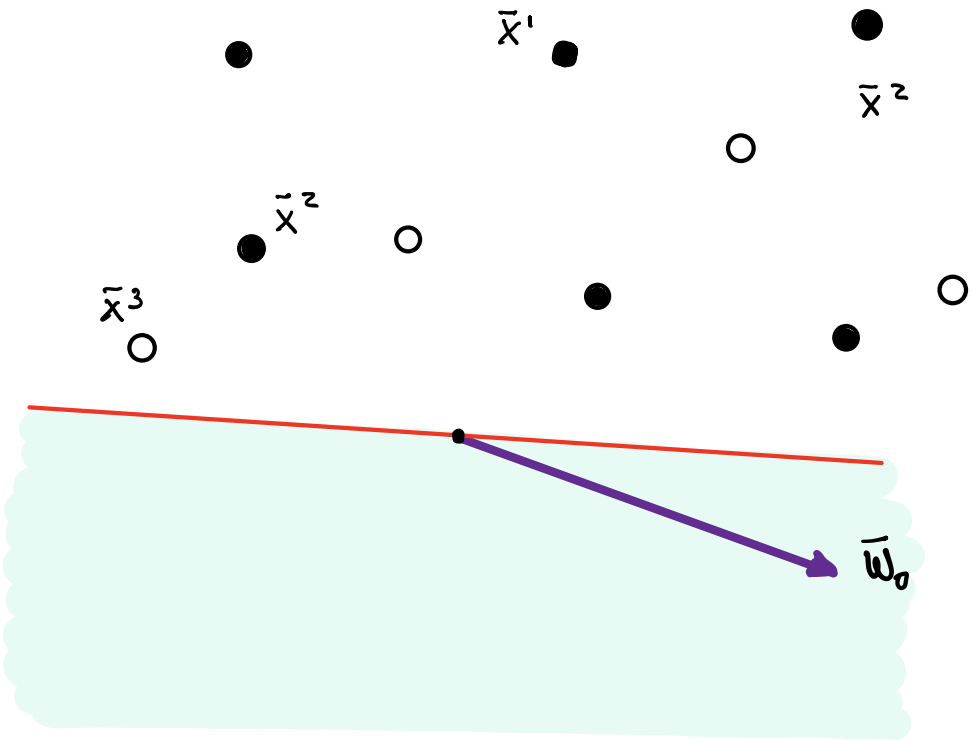

x_2
$\vec{x}^1$
$\vec{x}^2$
$\vec{x}^3$
$\vec{w}_0$
x_1


$\bar{w}_2$
$\bar{x}^1$
$\bar{w}_1$
$\bar{x}^2$
$\bar{x}^3$
$\bar{x}^1$
$\bar{w}_0$

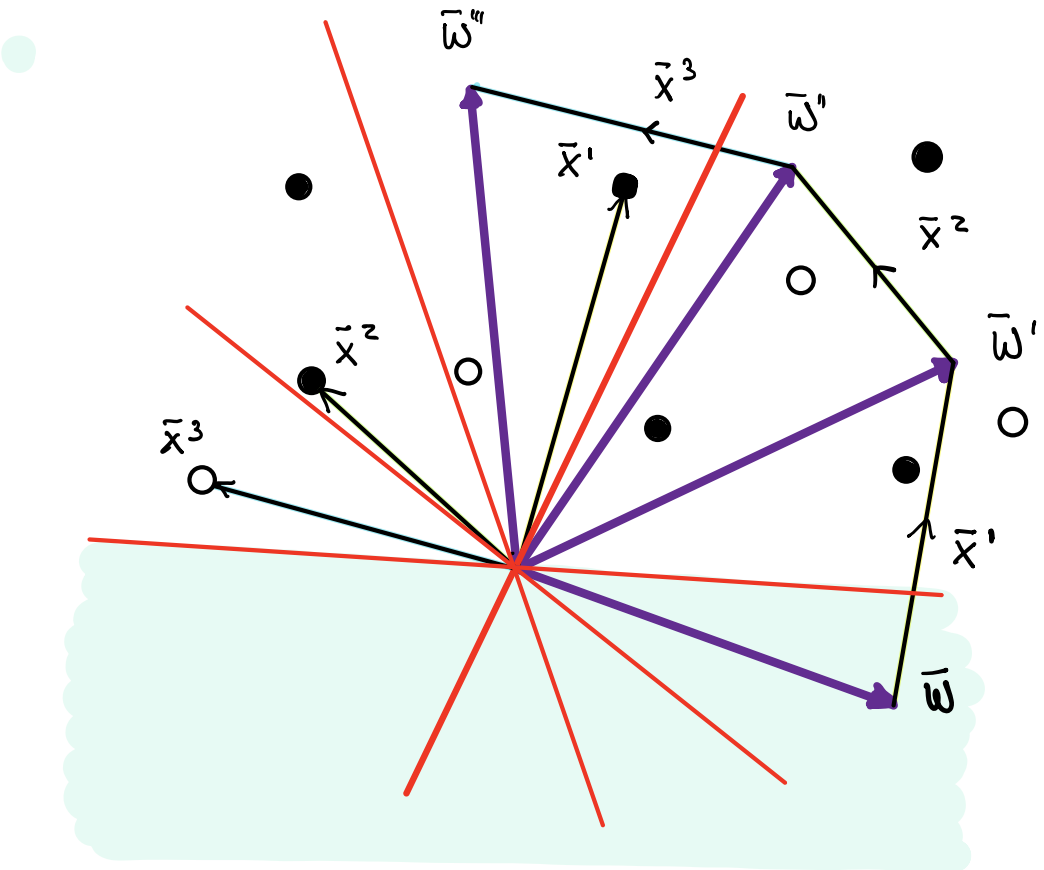


η


$\vec{w}_3$
$\vec{x}^3$
$\vec{w}_2$
$\vec{x}^1$
$\vec{w}_1$
$\vec{x}^2$
$\vec{x}^3$
$\vec{w}_0$


$\tilde{x}^1$
$\tilde{x}^2$
$\tilde{x}^2$
$\tilde{x}^3$
$\vec{w}_o$

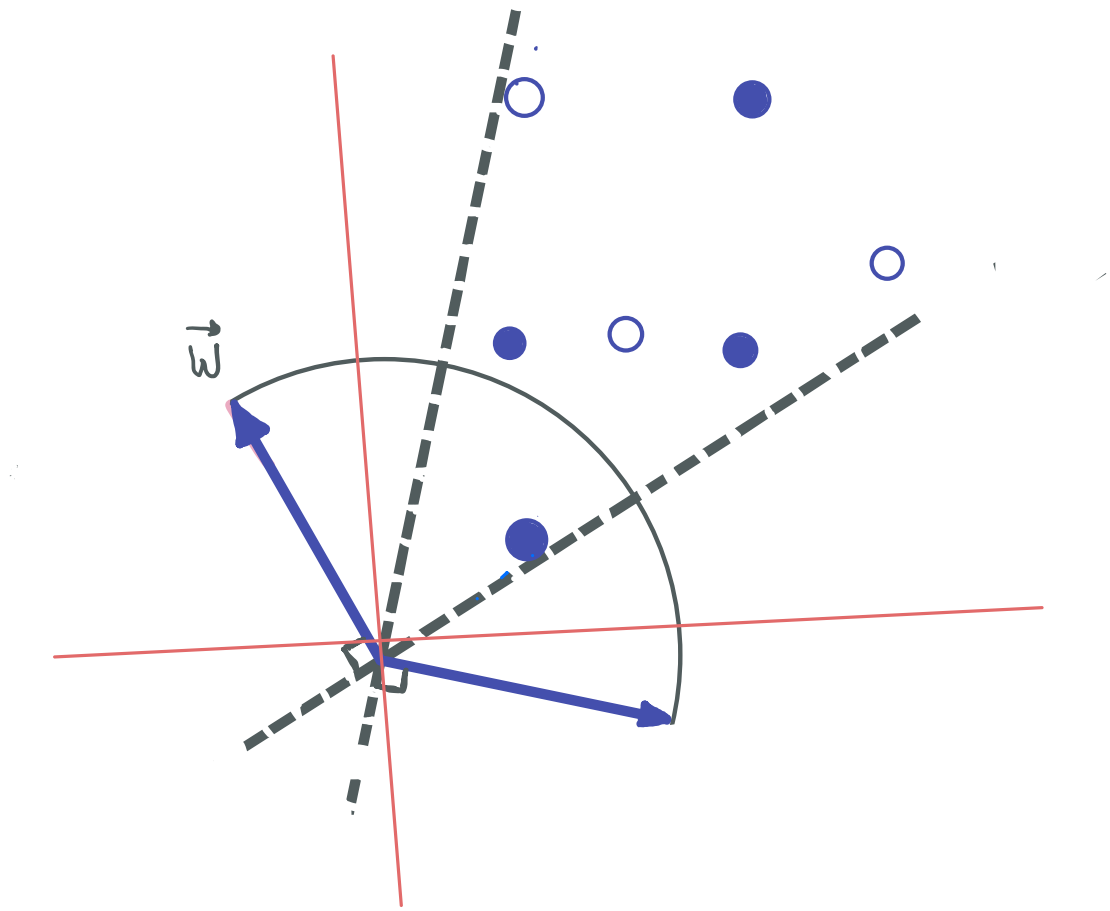
Miremos la evolución en el tiempo de la solución. Aquí llegamos en tres pasos pero para problemas reales necesitaremos millones de iteraciones.

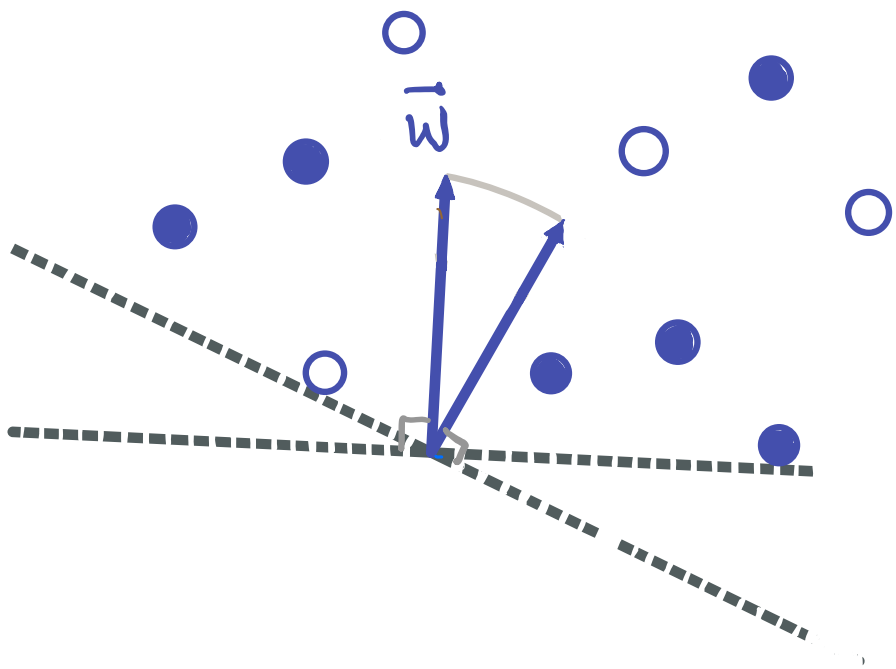


Notemos que una actualización del vector $\overline{w}$ implica mostrarle todos los elementos del conjunto de entrenamiento a la red. A este número de iteraciones lo llamamos una **ÉPOCA**.

Salvo casos muy patológicos, no existe en general un único vector $\vec{w}$ que resuelve el desafío, sino infinitos.

Miremos dos buenos ejemplos artificiales:





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

para cada posible $\overline{W}$. $D(\overline{W})$ es la mínima proyección de $\overline{W}$ sobre los $\bar{X}^{\alpha}$ del conjunto de entrenamiento. Cómo está dividido por $\overline{W}$ solo depende de la dirección de $\overline{W}$ pero no de su magnitud.

Para un dado valor de K , W puede variar en magnitud y en dirección. Podemos definir la cantidad:

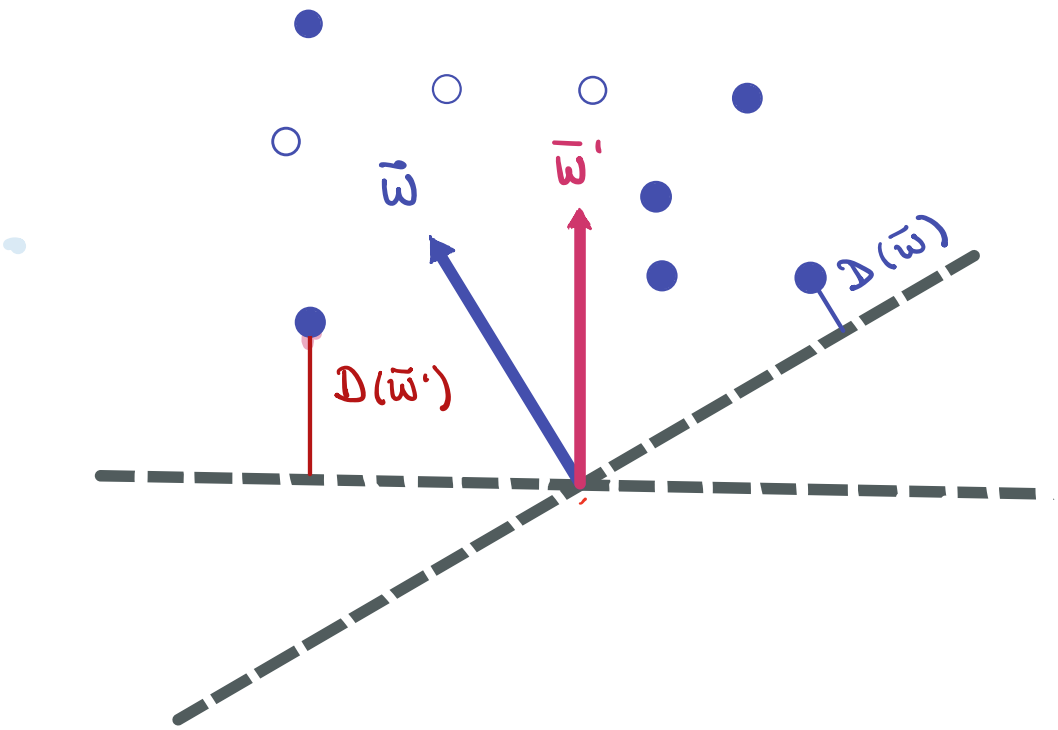
$$D(\vec{w}) = \min_{\mu} \frac{1}{|\vec{w}|} \cdot |\vec{w}| \cdot |\vec{x}^{\mu}| \cos(\theta_{\mu})$$

$$= \min_{\mu} |\vec{x}^{\mu}| \cos(\theta_{\mu})$$

Si $D(\overline{W})$, que es un número real (no un vector), es positivo, entonces todos los vectores $\bar{X}^{\alpha}$ están en el lado correcto. Si es negativo, hay al menos un $\bar{X}^{\beta}$ para el cual no funciona el método .

Supongamos que todos los elementos del conjunto de entrenamiento están del lado correcto. Podemos definir el mejor ***D***:

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



Pero no nos conformamos con esto. Para que la solución que buscamos sea más robusta (expliquemos esto) vamos a pedir que

$$\xi^\mu h^\mu > N\kappa$$

kappa

$$\Delta w_k = \eta\, \Theta(N\kappa - \xi^\mu h^\mu)\, \xi^\mu \xi_k^\mu$$

donde:

$$\Theta(z) = \begin{cases} 1 & z \geq 0 \\ 0 & z < 0 \end{cases}$$

Esta regla se llama *REGLA DEL APRENDIZAJE DEL PERCEPTRÓN* y fue publicada por Frank Rosenblatt en 1962.

Se puede demostrar rigurosamente que si nuestro problema por aprender es **LINEALMENTE SEPARABLE**, la regla converge a una solución $\overline{W}$ en un número finito de pasos.