REDES NEURONALES

2021

Clase 17 Parte 1

Facultad de Matemática, Astronomía, Física y Computación
Universidad Nacional de Córdoba

Martes 12 de octubre 2021

http://www.famaf.unc.edu.ar/~ftamarit/redes2021

https://www.famaf.unc.edu.ar/course/view.php?id=798

Es muy importante dividir a todos lo ejemplos etiquetados en dos grupos:

$\{\bar{\xi}^{\mu}\}$ con $\mu = 1, 2, \dots, P$ Conjunto de entrenamiento

$\{\bar{\xi}^{\nu}\}$ con $\nu = 1, 2, \dots, T$ Conjunto de testeo

En general tenemos que $P > T$

El conjunto de entrenamiento es una muestra del universo de datos que debemos clasificar y la solución que ajusta bien al conjunto de entrenamiento seguramente ajustará peor a los casos que están fuera del consjunto de enternamiento.

EL PERCEPTRON SIMPLE COMO CLASIFICADOR II

Hemos visto en la última clase que podemos usar un perceptrón simple como clasificador binario siempre y cuando el problema sea **linealmente separable**.

No olvidemos que ésta es una condición que debe satisfacer el problema (el conjunto de entrenamineto) y no el algoritmo.

Recordemos que el objetivo es, para problemas *linealmente separables*, encontrar los N valores de los parámetros $\vec{W}$ y el valor del umbral.

$$\vec{w}^{\,nuevo} = \vec{w}^{\,viejo} + \Delta\vec{w}$$

$$\boxed{\Delta w_k = \eta\, \Theta\left(N\kappa - \zeta^{\mu} h^{\mu}\right) \zeta^{\mu} \xi_k^{\mu}}$$

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

El algoritmo en pseudo código

Elegimos los parámetros ***W***

Sea *época=1*
Repetimos hasta que *época=época_maxima*

Sea mu =1

Repetimos, hasta que *mu=p*

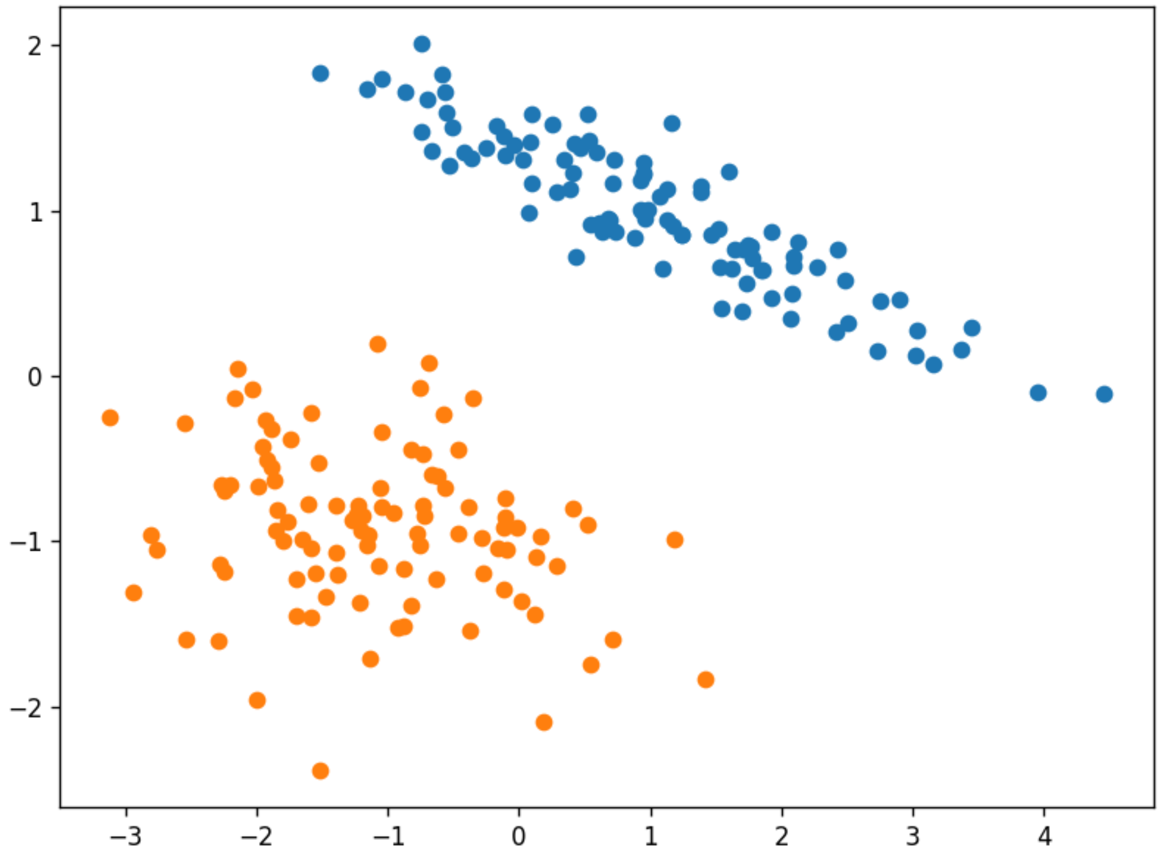
Calculamos $O_i^\mu = \text{signo}(h_i^\mu) = \text{signo}(\bar{W} \cdot \bar{\xi}^\mu)$

Si $O_i^\mu = \xi_i^\mu$ entonces no hacemos nada,
pero si $O_i^\mu \neq \xi_i^\mu$ entonces cambiamos W

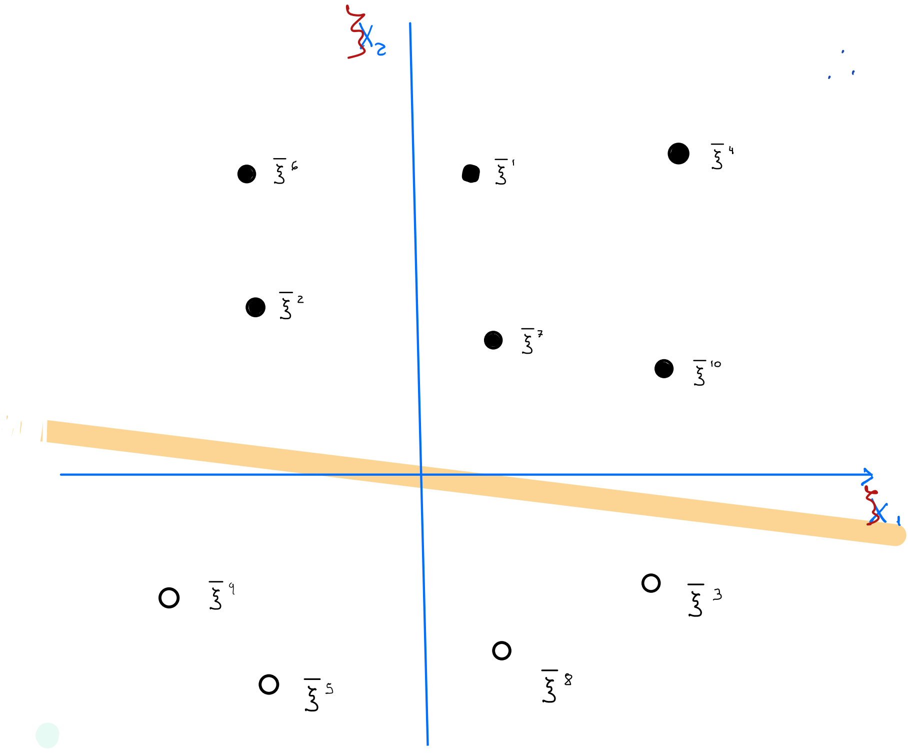
$$W_{ik} = W_{ik} + \eta(\xi^\mu - O^\mu)\xi_i^\mu$$

$$\left[\begin{array}{l} \text{si } \xi_i^\mu = 1 \text{ y } O^\mu = -1 \quad W_{ik} = W_{ik} + 2\eta\,\xi_i^\mu \\ \text{si } \xi_i^\mu = -1 \text{ y } O^\mu = 1 \quad W_{ik} = W_{ik} - 2\eta\,\xi_i^\mu \end{array}\right]$$

mu = mu+1
Volvemos a “Repetimos (mu)”
época = época + 1
Volvemos a “Repetimos (época)”

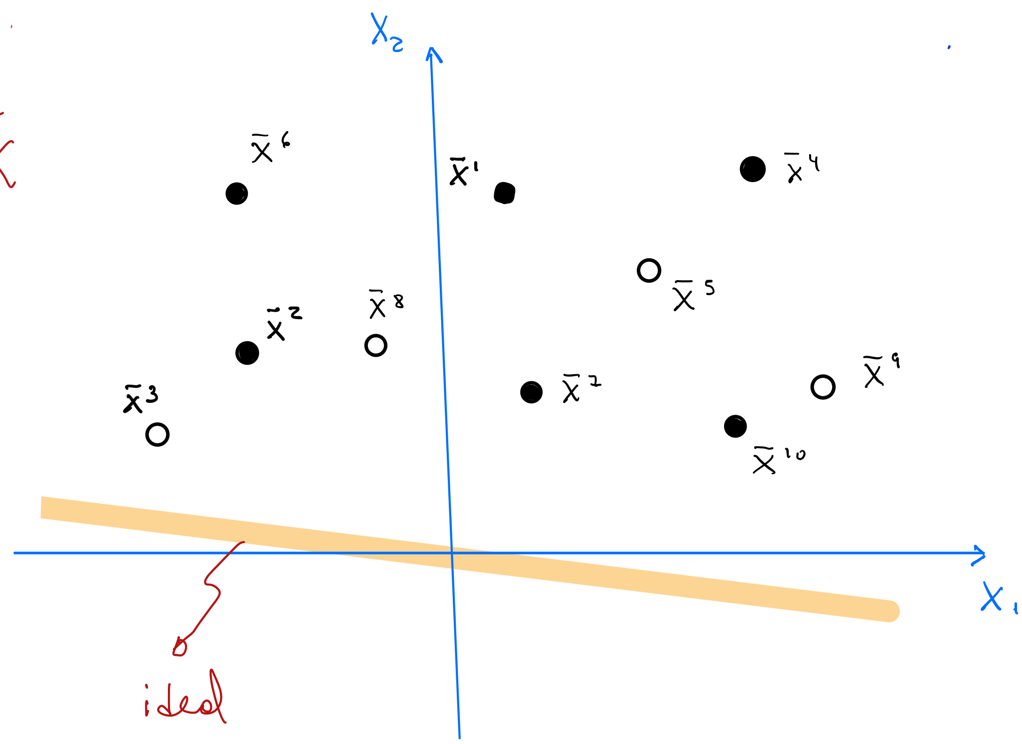

2
1
0
−1
−2
−3
−2
−1
0
1
2
3
4

Representación $\bar{\xi}$

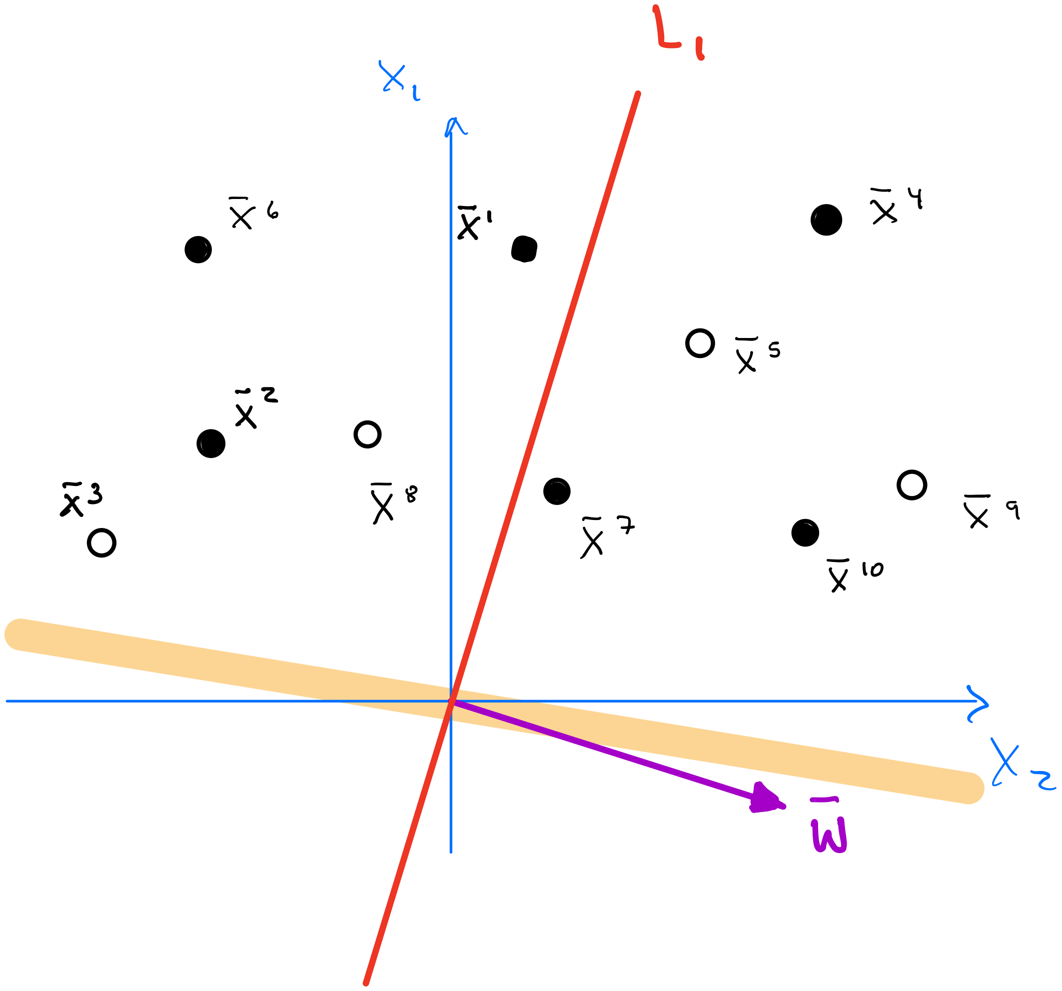


$$\bar{x}^\mu = \zeta^\mu \bar{\xi}$$

Representación $\bar{X}$



Ahora elegimos un valor inicial aleatorio de $\bar{W}$ y este vector define una recta L_1. Vemos que este $\bar{W}$ resuelve bien los ejemplos 4, 5, 7, 9 y 10 pero, resuelve mal los ejemplos 1, 2, 3, 6 y 8.



Le mostramos el ejemplo 1, que tiene que dar

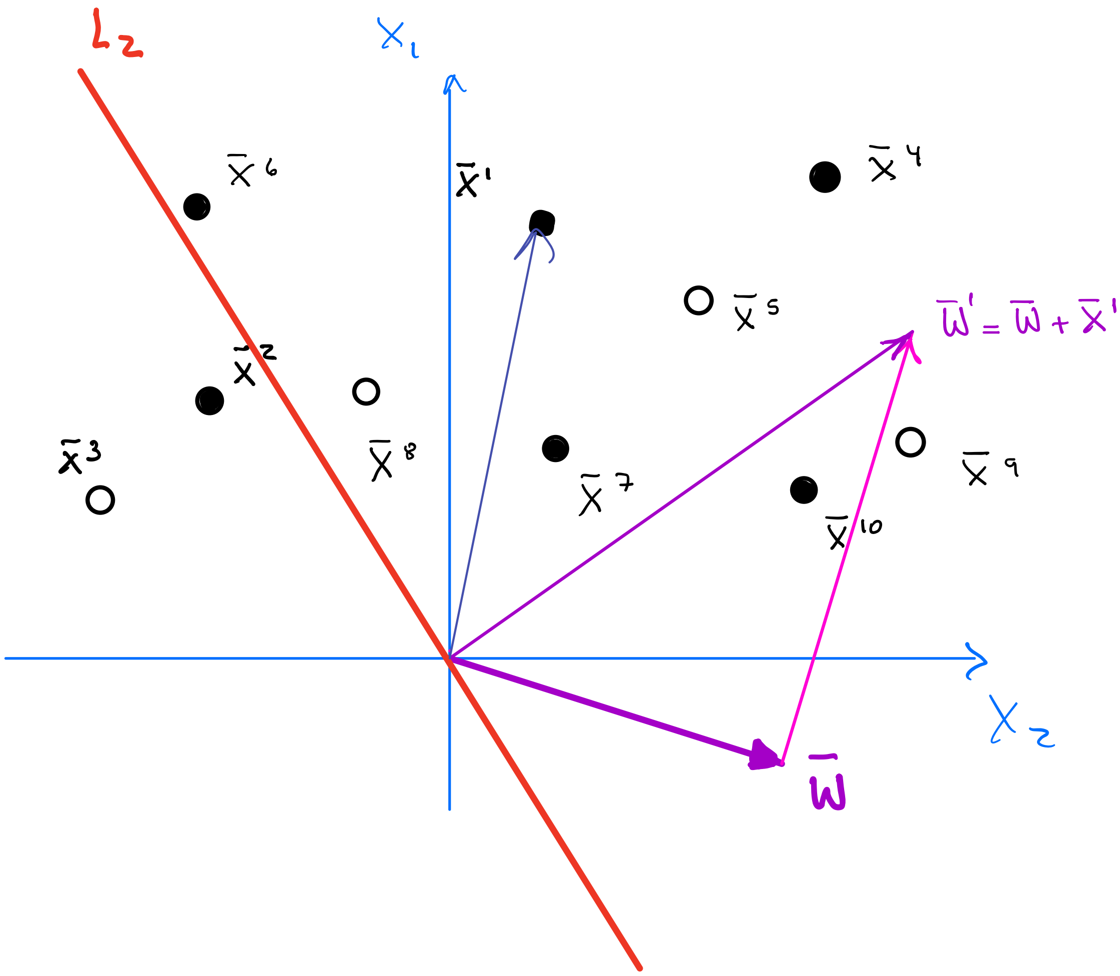
$$O^1 = \zeta^1 = +1$$

Pero como el ángulo entre $\bar{W}$ y $\bar{X}$ es obtuso, da mal:

$$\zeta^1 O^1 = O^1 = \text{signo}\,(\bar{w} \cdot \bar{x}) = -1$$

Tomamos $\eta = \frac{1}{2}$ (exageradamente alto)

$$\bar{W}' = \bar{W} + \bar{X}^1$$



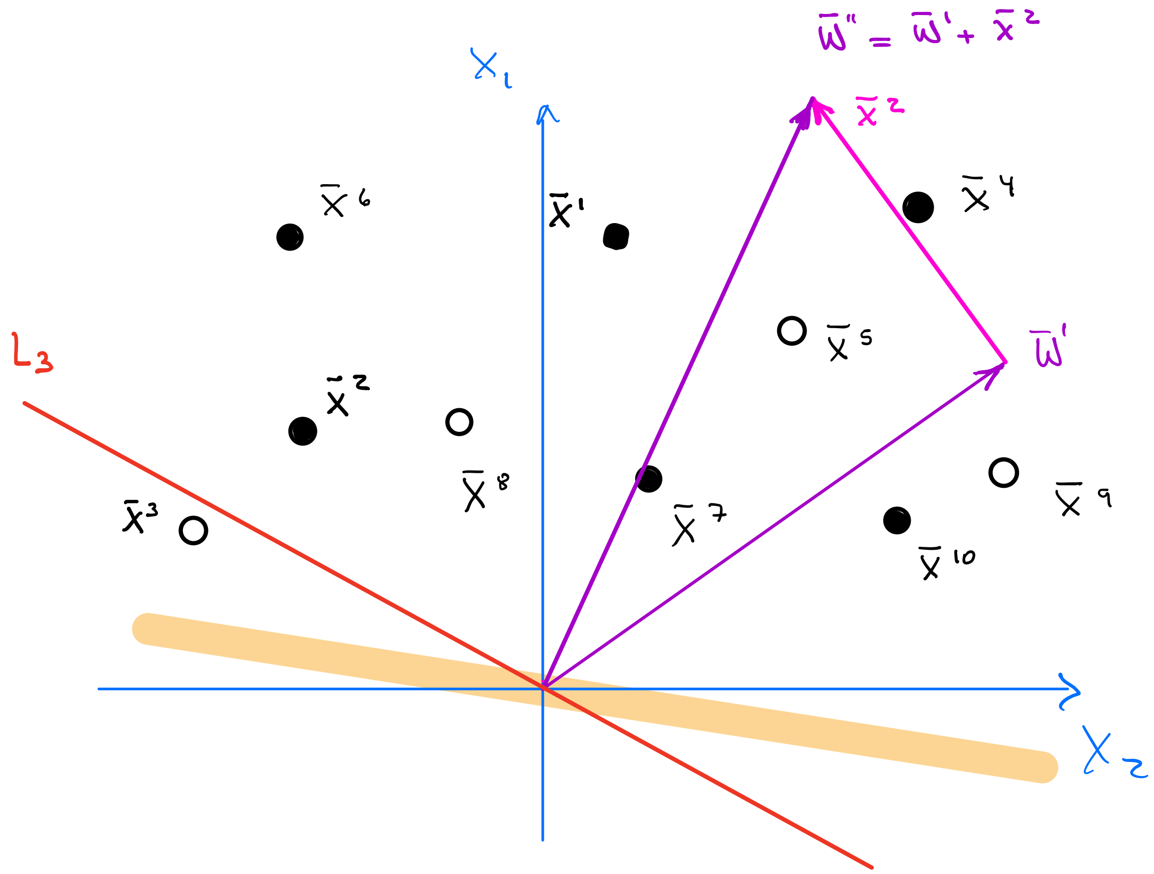
El nuevo vector $\bar{W}'$ define una nueva recta L_2 que ahora resuelve bien los ejemplos todos los ejemplos menos el 2 y el 3.

Ahora pasamos a tratar el segundo elemento del conjunto de entrenamiento

$$\mu \to \mu + 1 \Rightarrow \mu = 2$$

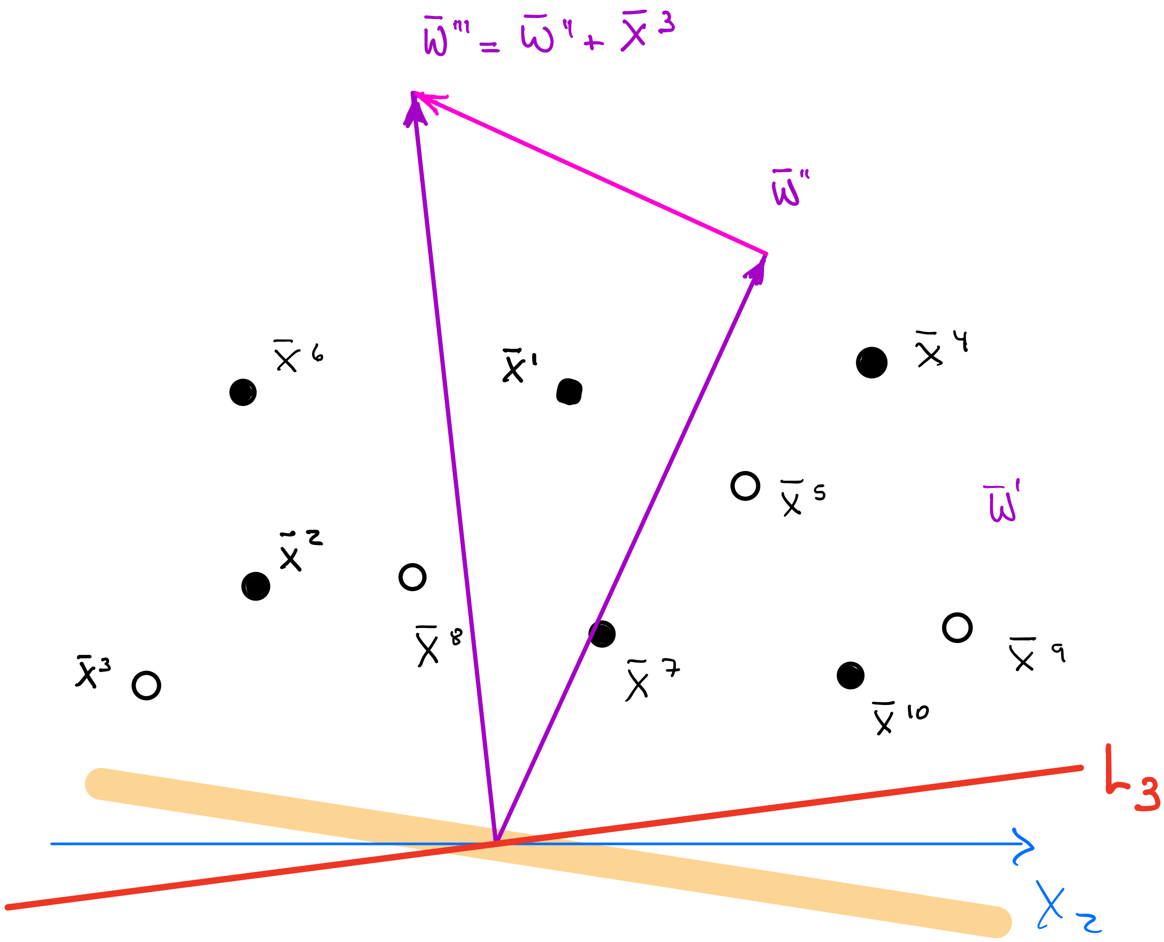
Vemos que el algoritmo resuelve mal este caso y hacemos

$$\bar{W}'' = \bar{W}' + \bar{X}^2$$



El nuevo vector $\bar{W}''$ define una nueva recta L_3 que resuelve mal solo el ejemplo 3. Pasamos ahora a mostrarle el ejemplo 3 y como lo hace mal aplicamos la regla del perceptrón:

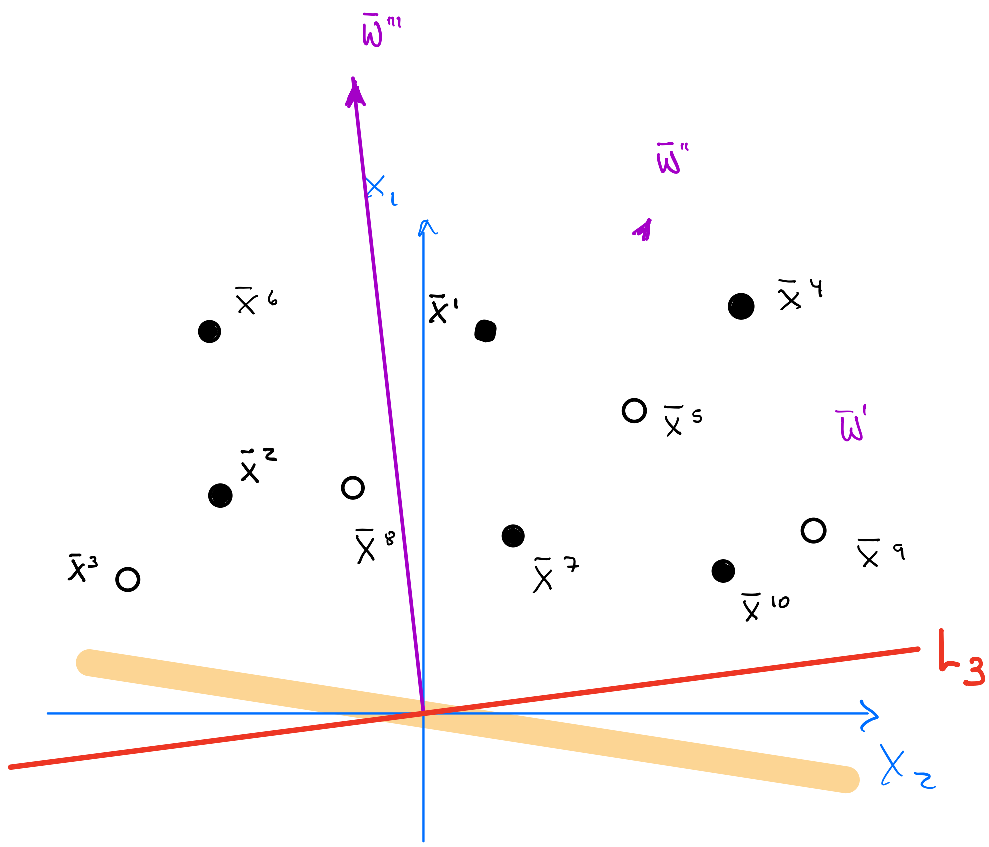
$$\bar{W}''' = \bar{W}'' + \bar{X}^3$$



Ahora le mostramos los restantes a partir del cuarto y hasta el décimo,y a todos los resuelve bien. En pocos pasos hemos encontrado una solución, un vector $\widetilde{W}'''$ que resuelve todos los casos.

Cada vez que le presentamos todos los ejemplos decimos qué pasó "*una época*".

Cuando pasamos una época sin errores el algoritmo se detiene.



Observemos que $\overline{W}'''$ resuelve bien los diez ejemplos del conjunto de entrenamiento, pero no nos da la solución exacta del problema real, que está dada por la recta naranja. Cuando le presentemos al algoritmo casos reales que están entre nuestra recta solución y la solución exacta, lo resolverán mal