

Inteligencia Artificial

Perceptrones

Adaptado de: **CS 188: Artificial Intelligence**

University of California, Berkeley

[transparencias adaptadas de Dan Klein, Pieter Abbeel con ayuda de Ekaitz Jauregi]



Universidad
del País Vasco

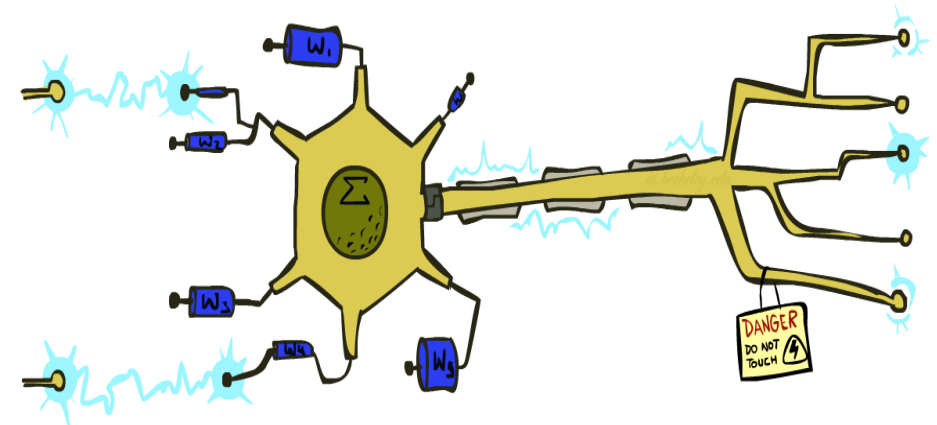
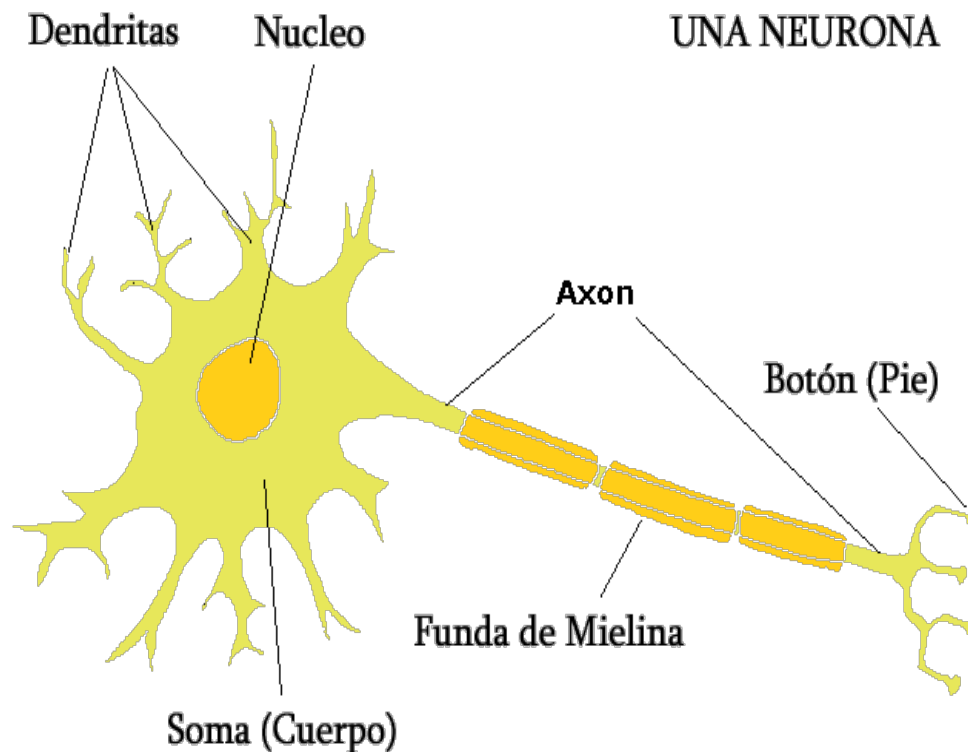
Euskal Herriko
Unibertsitatea

HiTZ

Hizkuntza Teknologiako Zentroa
Basque Center for Language Technology

Un poco de Biología (Simplificada)

- Una ligera inspiración: neuronas humanas



Cuando las dendritas reciben mensajes químicos, las neuronas cambian el balance de iones (átomos cargados electrónicamente) entre su interior y el exterior de la membrana celular. Cuando este cambio alcanza un nivel umbral, este efecto se expande a través de la membrana de la célula hasta el axón. Cuando alcanza al axón, se inicia un potencial de acción (Dr. C. George Boeree).

Clasificación

Ejemplos: $x_1, x_2, x_3, \dots, x_n$

Clasificarlos: $y_1 = -1, y_2 = 1, y_3 = -1, \dots, y_n = -1$

$$x_1 \rightarrow f(x_1) \rightarrow w * f(x_1) \rightarrow y'_1$$

Clasificación guiada por el error



Clasificación

➤ Ejemplos

```
hola,  
¿Quieres recibir gratis un Iphone? No tienes más  
que clickar en el siguiente link. . .
```

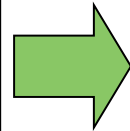
```
¡¡Hechamos la casa por la ventana!! . . .  
descuentos increíbles para los primeros 100  
usuarios que clicken en el siguiente link. ...
```

Vectores de rasgos

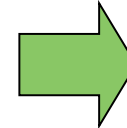
 x $f(x)$ y

Hola,

¿Quieres cratuchos de
impresora **gratuitos**?
¿Porque **pagar** mas
cuando los puedes
conseguir
¡COMPLETAMENTE
GRATIS! Solo

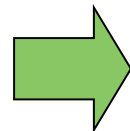


# gratis	: 2
hola	: 1
pagar	: 1
...	

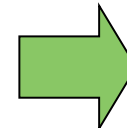


SPAM
0
NO SPAM

2



PIXEL 7,12	: 1
PIXEL 7,13	: 0
...	
NUM_LOOPS	: 1
...	



"2"

Errores, y ¿cómo actuar?

➤ Ejemplos de errores

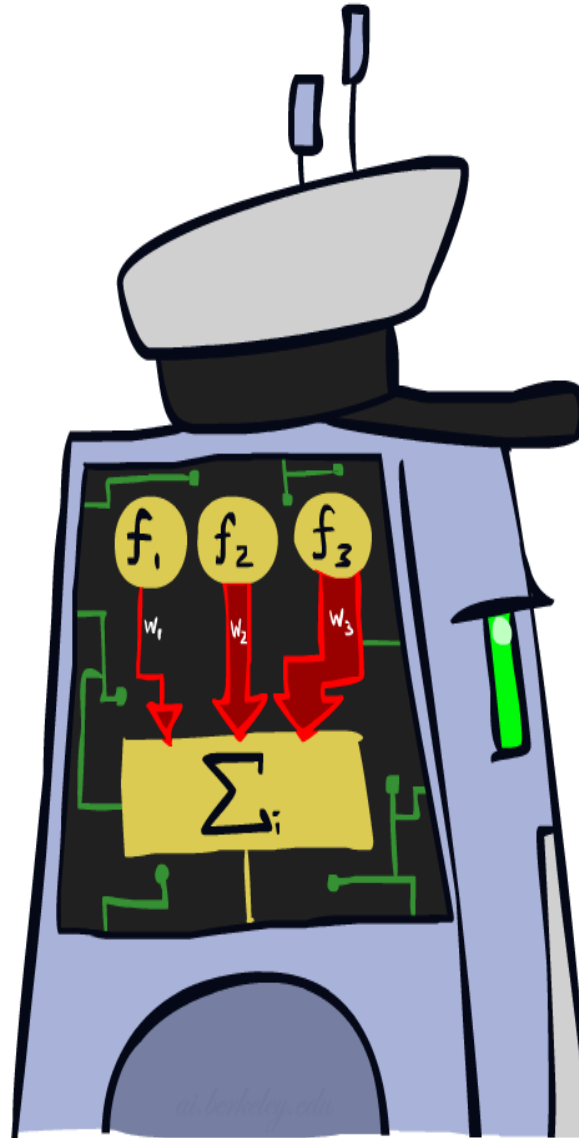
Querido cliente de GlobalSCAPE,
GlobalSCAPE se ha aliado a ScanSoft para ofecerle
la última versión de OmniPage Pro, por solo 99.99
euros- el precio habitual es ¡499!-. La más
frecuente de las preguntas sobre esta oferta es:
¿Es real? Queremos asegurarle que esta oferta está
autorizada por ScanSoft, y es genuina y veraz.
Puede recibir. . .

. . . Para recibir el certificado promcional de 30
euros, clickar sobre el siguiente link
<http://www.amazon.com/apparel>
Y verá el link que aparece resaltado de 30 euros
de oferta. Podrá encontrar todos los detalles
allí. Tenga en cuenta, que si no quiere recibir
futuros e-mails anunciando nuevos lanzamientos, por
favor haga click en ...

¿Qué hacer frente a los Errores?

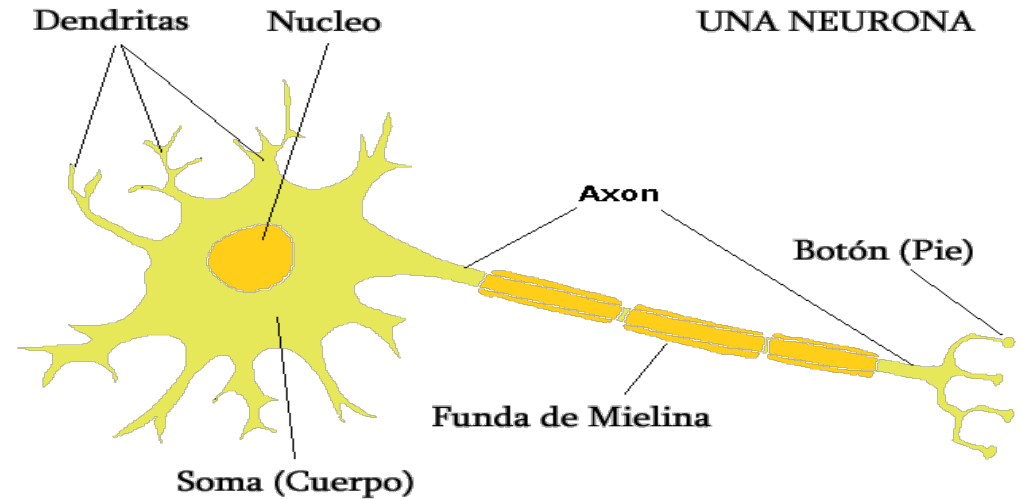
- Problema: todavía hay Spam en tu e-mail
- Necesitas más **rasgos** – ¡las palabras contenidas en el e-mail no son suficientes!
 - ¿El emisor de ese e-mail pertenece a tus contactos?
 - ¿Hay más usuarios que han recibido el mismo e-mail?
 - ¿Las URL que aparecen realmente te dirigen a los lugares que anuncian?
 - ¿Se dirigen a ti por tu nombre?

Clasificadores Lineales



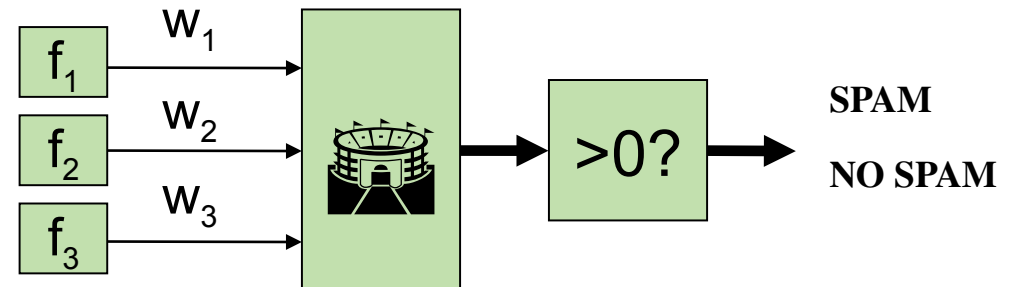
Clasificadores lineales

- Entradas son **valores para unos rasgos**
- A cada rasgo se le asigna un **peso**
- La suma **es la activación**



$$\text{activación}_w(x) = \sum_i w_i \cdot f_i(x) = w \cdot f(x)$$

- Si la activación es:
 - **Positiva, salida +1**
 - **Negativa, salida -1**



Pesos

- Clasificación binaria: comparar el vector de los rasgos con el vector de los pesos >0 o <0 (0 en este caso es el umbral)
- Aprendizaje: a partir de los ejemplos ajustar/adecuar los pesos

pesos

$$\begin{pmatrix} \# \text{ gratis} & : & 4 \\ \text{tu_nombre} & : & -1 \\ \text{ORTOGR_ERR} & : & 1 \\ \text{EMISOR_CONOCIDO} & : & -3 \\ \dots & & \end{pmatrix}$$

w

Rasgos EMAIL1

$$f(x_1) \begin{pmatrix} \# \text{ gratis} & : & 2 \\ \text{tu_nombre} & : & 0 \\ \text{ORTOGR_ERR} & : & 2 \\ \text{EMISOR_CONOCIDO} & : & 0 \\ \dots & & \end{pmatrix}$$

$$4*2 + -1*0 + 1*2 + -3*0 = 10 \rightarrow \text{SPAM}$$

Producto escalar $w \cdot f$ positivo supone clase positiva.

Por ejemplo:

Positivo \rightarrow SPAM

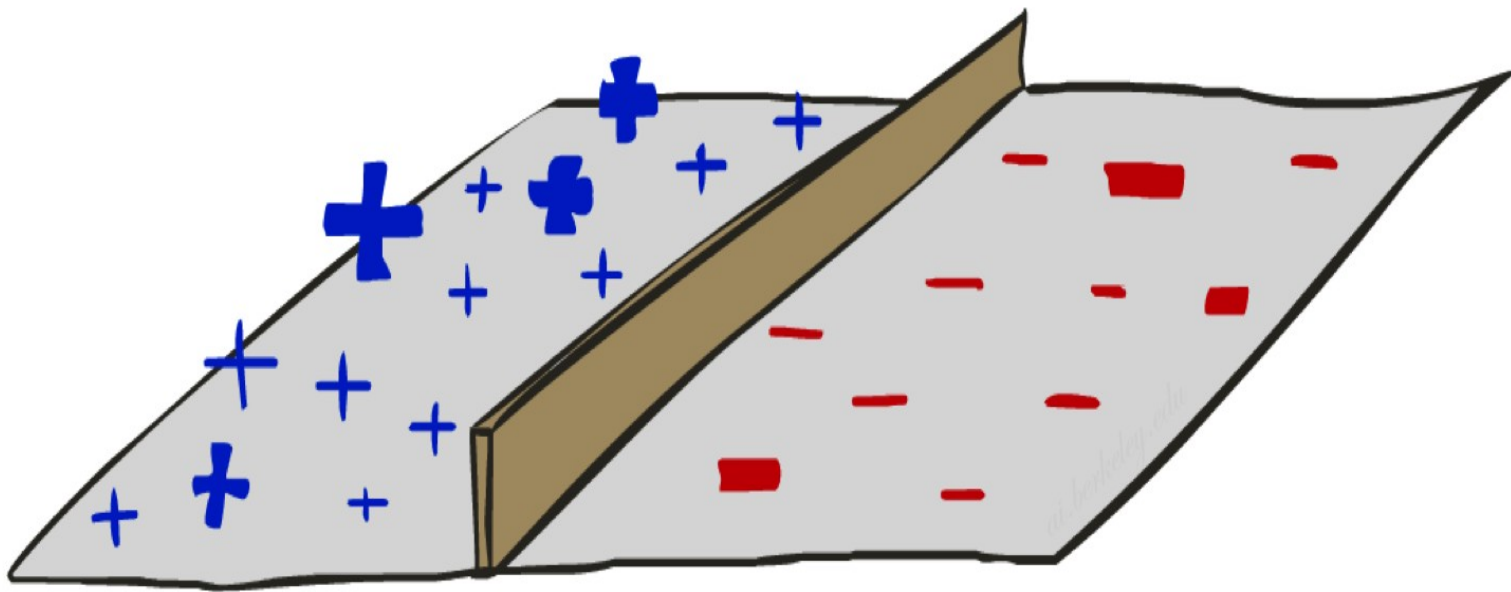
Negativo \rightarrow NO SPAM

$$4*0 + -1*1 + 1*1 + -3*1 = -3 \rightarrow \text{NO SPAM}$$

$$f(x_2) \begin{pmatrix} \# \text{ gratis} & : & 0 \\ \text{tu_nombre} & : & 1 \\ \text{ORTOGR_ERR} & : & 1 \\ \text{EMISOR_CONOCIDO} & : & 1 \\ \dots & & \end{pmatrix}$$

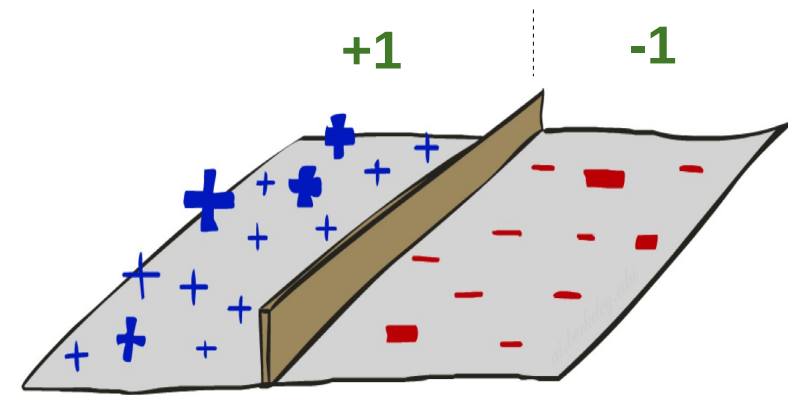
Rasgos EMAIL2

Fronteras de decisión



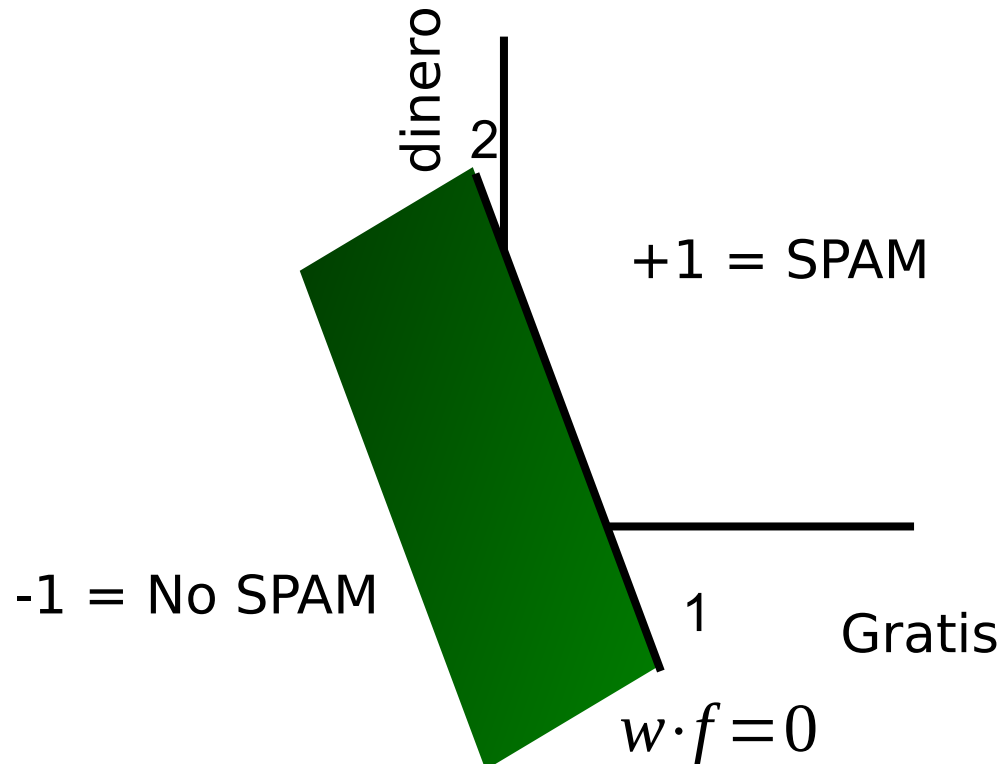
Decisión binaria

- En el espacio generado por los vectores de rasgos
 - Los ejemplos son puntos
 - Un vector de pesos será un hiperplano
 - Un lado se corresponde con $Y=+1$
 - El otro se corresponde con $Y=-1$

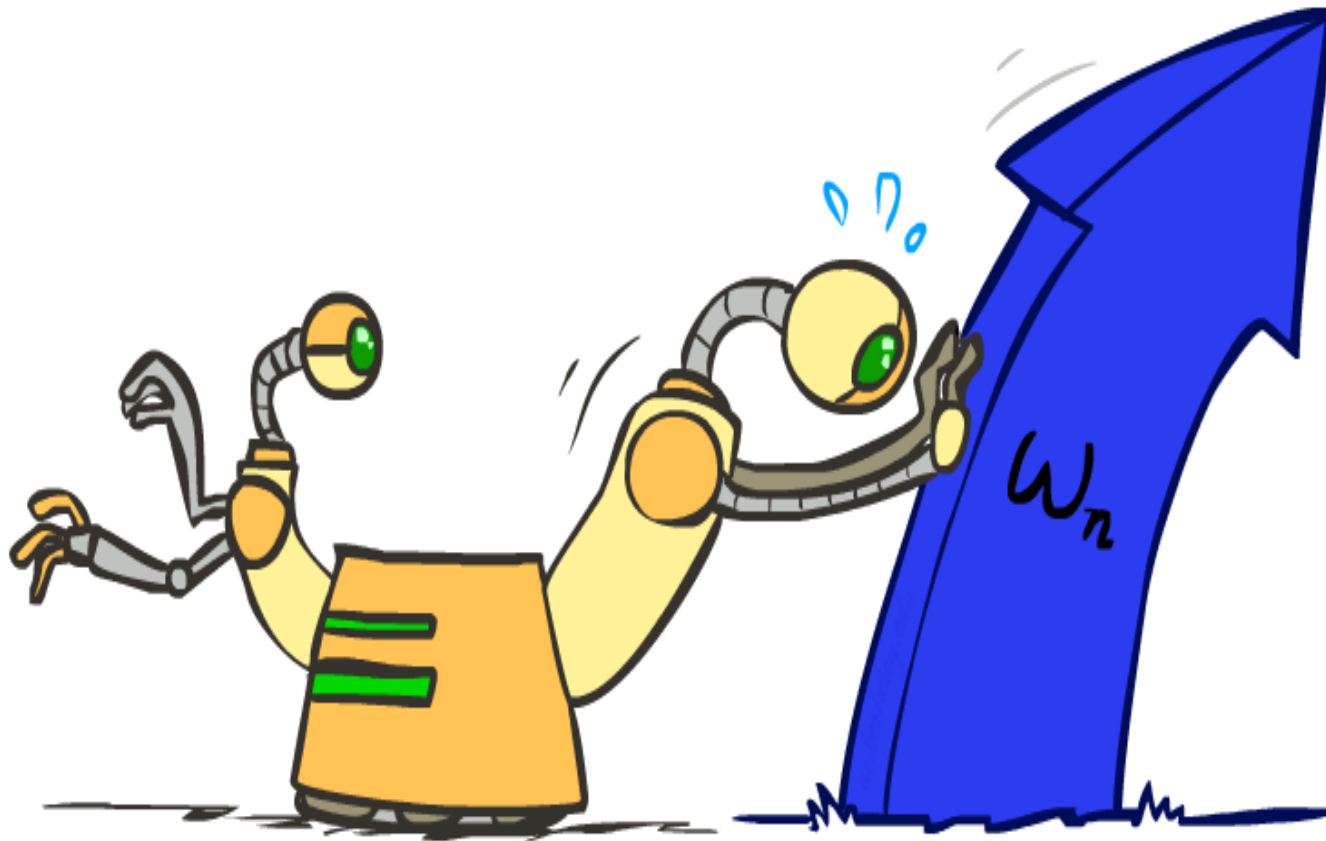


w

BIAS	:	-3
Gratis	:	4
dinero	:	2
...		



Actualización de Pesos



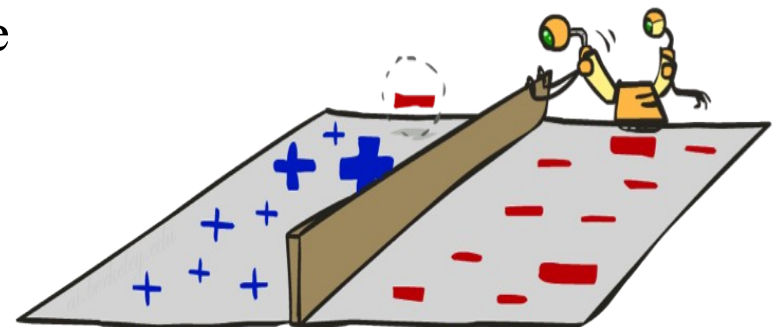
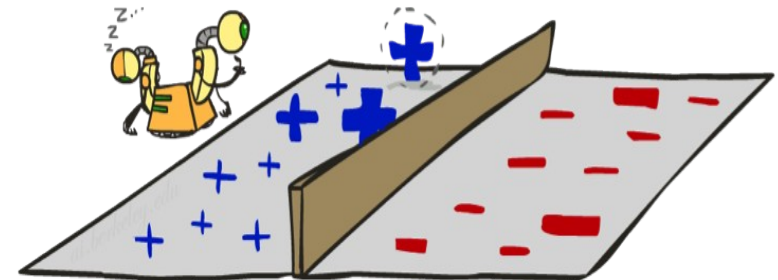
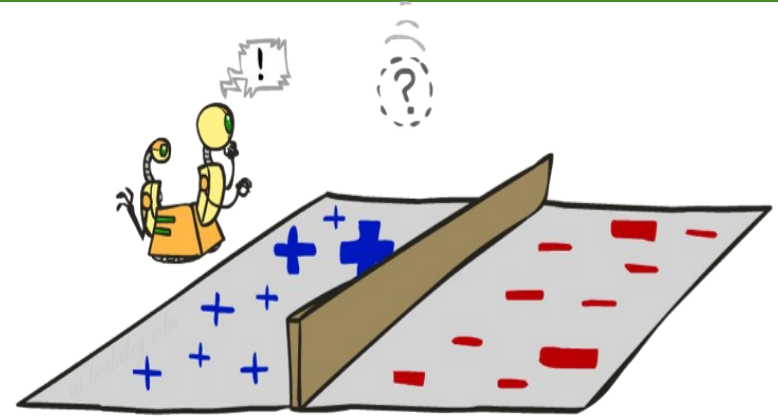
Aprendizaje: Perceptron Binario

y la clase correcta y' la predicha

- Inicializar pesos = valor random
- Por cada instancia de entrenamiento:
 - Clasificar empleando los pesos actuales

$$y' = \begin{cases} +1 & \text{if } w \cdot f(x) \geq 0 \\ -1 & \text{if } w \cdot f(x) < 0 \end{cases}$$

- Si correcto ($y = y'$), ¡no cambiar!
- Si incorrecto: ajustar (**actualizar**) el vector de pesos



Aprendizaje: Perceptron Binario

- Inicializar pesos = valor random
- Por cada instancia de entrenamiento:
 - Clasificar empleando los pesos actuales

$$y' = \begin{cases} +1 & \text{if } w \cdot f(x) \geq 0 \\ -1 & \text{if } w \cdot f(x) < 0 \end{cases}$$

- Si correcto ($y = y'$), ¡no cambiar los pesos!
- Si incorrecto: ajustar (actualizar) el vector de pesos sumando o restando el vector de los rasgos.
 - Restar si la clase y (la real) es -1 y Sumar si la clase y (la real) es +1
- Entonces si y' ha sido 1 & y ha sido -1 $\rightarrow w = w - f(x)$
- Entonces si y' ha sido -1 & y ha sido 1 $\rightarrow w = w + f(x)$

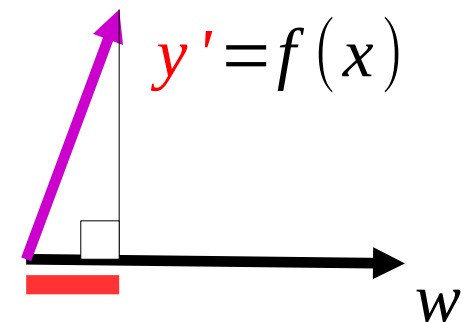
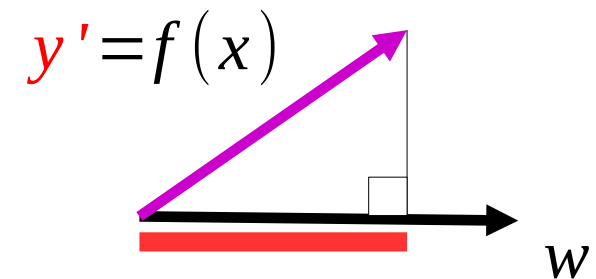
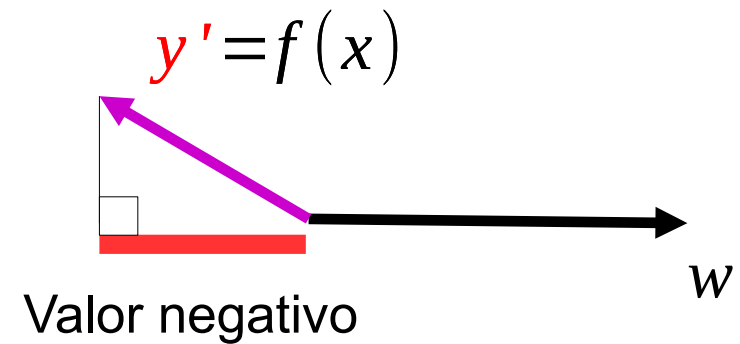
Aprendizaje: Producto punto de vectores y su significado

- Por cada instancia de entrenamiento:
 - Clasificar empleando los pesos actuales

$$y' = \begin{cases} +1 & \text{if } w \cdot f(x) \geq 0 \\ -1 & \text{if } w \cdot f(x) < 0 \end{cases}$$

Intuición tras el producto punto:

Cuanto de $f(x)$ va en la misma dirección que w o cuanto se asemejan



¿Qué sucede cuándo son ortogonales? ¿Y qué sucede cuándo son iguales?

Aprendizaje: Perceptron Binario

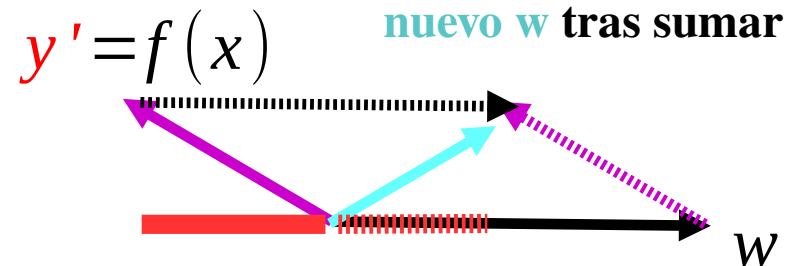
- Inicializar pesos = valor random
- Por cada instancia de entrenamiento:
 - Clasificar empleando los pesos actuales

$$y' = \begin{cases} +1 & \text{if } w \cdot f(x) \geq 0 \\ -1 & \text{if } w \cdot f(x) < 0 \end{cases}$$

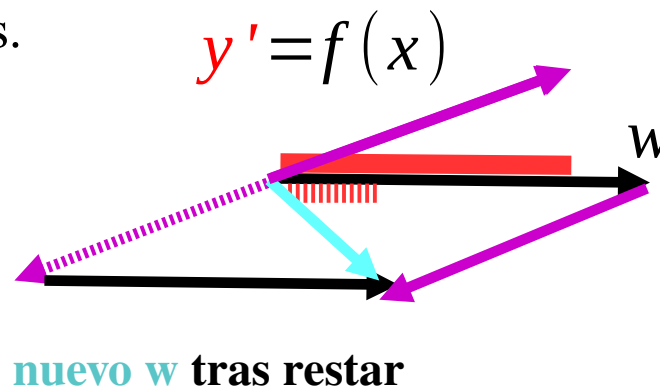
- Si correcto ($y = y'$), ¡no cambiar!
- Si incorrecto: ajustar (actualizar) el vector de pesos sumando o restando el vector de los rasgos.

- Sumar si la clase y (real) es +1
- Restar si la clase y (real) es -1

$$-y' = -f(x)$$



El resultado ha sido $y' = -1$
pero debería de haber sido 1 luego tienes
que intentar **acercar** los vectores para que el
coseno si se repitiese f (o similar) te de
positivo
Por eso **sumamos**



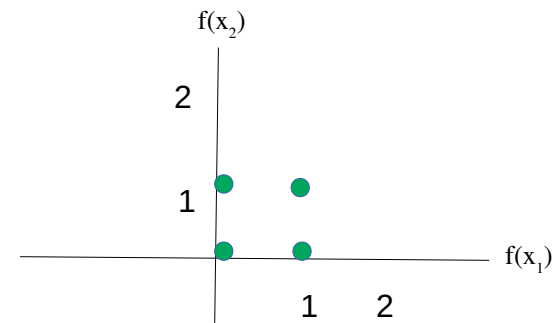
El resultado ha sido $y' = 1$
pero debería de haber sido
-1 luego tienes que intentar
alejar los vectores para que
el coseno si se repitiese f (o
similar) te de positivo
Por eso **restamos**

Ejemplo: AND

- Se quiere hacer un perceptron que entienda el operador AND

Inputs		Output
$f(x_1)$	$f(x_2)$	Z
0	0	0
0	1	0
1	0	0
1	1	1

Graficando los puntos:



- ¿Existe recta para dividir los puntos? Inicialización de pesos por RANDOM y valor BIAS = 1

$f(x_1)$	$f(x_2)$	Z	Bias	Clase Correcta
0	0	0	1	-1
0	1	0	1	-1
1	0	0	1	-1
1	1	1	1	1

$$w_0 = 0.03 \quad w_1 = 0.66 \quad w_2 = 0.80$$

$$y = w_0 * \text{BIAS} + w_1 * f(x_1) + w_2 * f(x_2) \dots$$

< 0 clase -1
 ≥ 0 clase 1

- Primer conjunto: $x_1=0, x_2=0, y=0$, **clase correcta = -1**

$(0.03 * 1) + (0.66 * 0) + (0.8 * 0) = 0.03 \Rightarrow$ al ser $\geq 0 \Rightarrow$ **la clase predicha: 1 \Rightarrow INCORRECTA**

- Ajuste de los pesos:

$$w_0 = w_0 + \text{ClaseCorrecta} * \text{bias}$$

$$w_n = w_n + \text{ClaseCorrecta} * f(x_n)$$

$$w_0 = 0.03 + (-1) * 1 = -0.97$$

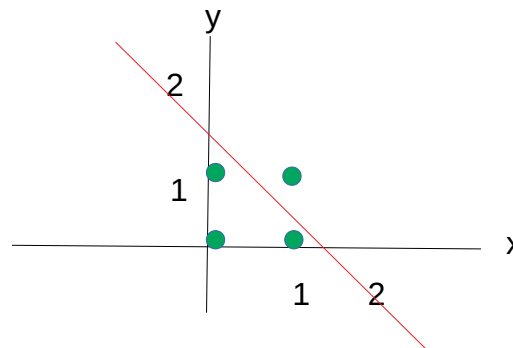
$$w_1 = 0.66 + (-1) * 0 = 0.66$$

$$w_2 = 0.80 + (-1) * 0 = 0.8$$

Ejemplo: AND

- Actualización de los pesos: $w_0 = -0.97$ $w_1 = 0.66$ $w_2 = 0.80$
- Segundo conjunto: $f(x_1)=0$, $f(x_2)=1$, $y=0$, clase = -1
 $(-0.97 * 1) + (0.66 * 0) + (0.8 * 1) = -0.17 \Rightarrow$ al ser $< 0 \Rightarrow$ Predicción de clase: -1 \Rightarrow **CORRECTO**
 - Tercer conjunto: $x_1=1$, $x_2=0$, $y=0$, clase = -1
 $(-0.97 * 1) + (0.66 * 1) + (0.8 * 0) = -0.31 \Rightarrow$ al ser $< 0 \Rightarrow$ Predicción de clase: -1 \Rightarrow **CORRECTO**
 - Cuarto conjunto: $x_1=1$, $x_2=1$, $y=1$, clase = 1
 $(-0.97 * 1) + (0.66 * 1) + (0.8 * 1) = 0.49 \Rightarrow$ al ser $\geq 0 \Rightarrow$ Predicción de clase: 1 \Rightarrow **CORRECTO**
 - Y que sucede con el primer conjunto?
 $(-0.97 * 1) + (0.66 * 0) + (0.8 * 0) = -0.97 \Rightarrow$ al ser $< 0 \Rightarrow$ Predicción de clase: -1 \Rightarrow **CORRECTO**

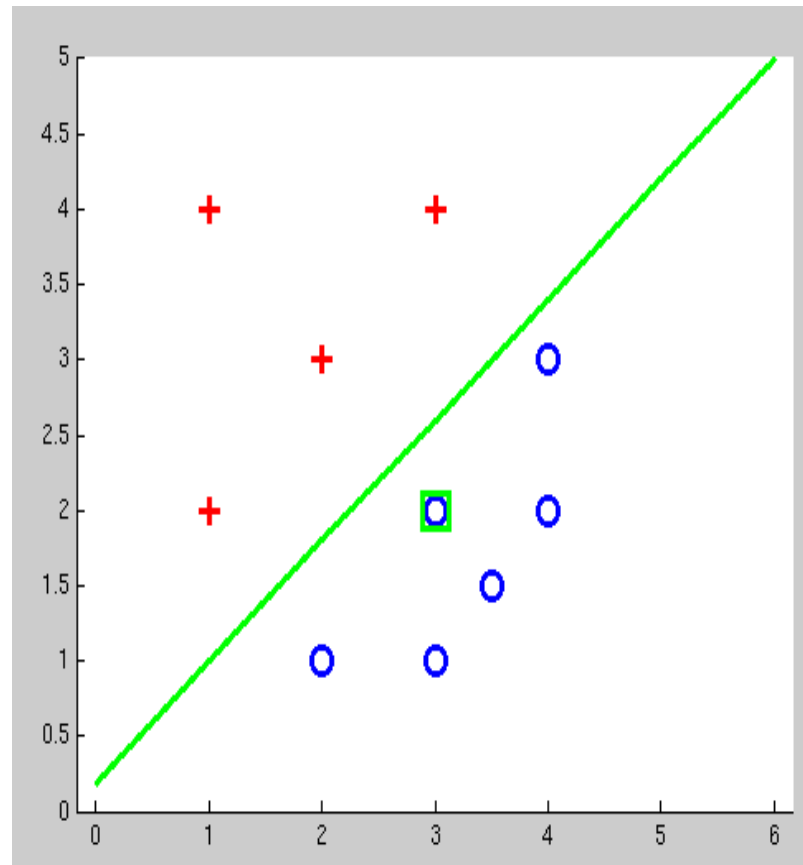
Si hacemos la ecuación de una recta con los pesos: $-0.97 + 0.66x + 0.8y = 0 \Rightarrow y = 1.2125 - 0.825x$



OJO! Se han consumido todos los items.
(Normalmente: 80% : training, 10% : dev
y 10% : test)

Ejemplos: Perceptron

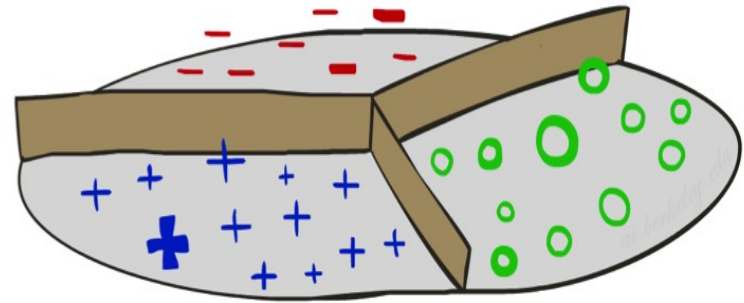
➤ Casos separables



Fronteras en decisiones Multiclase

- En clasificación multiclase:
- Un vector de pesos por cada clase:

$$w_y$$

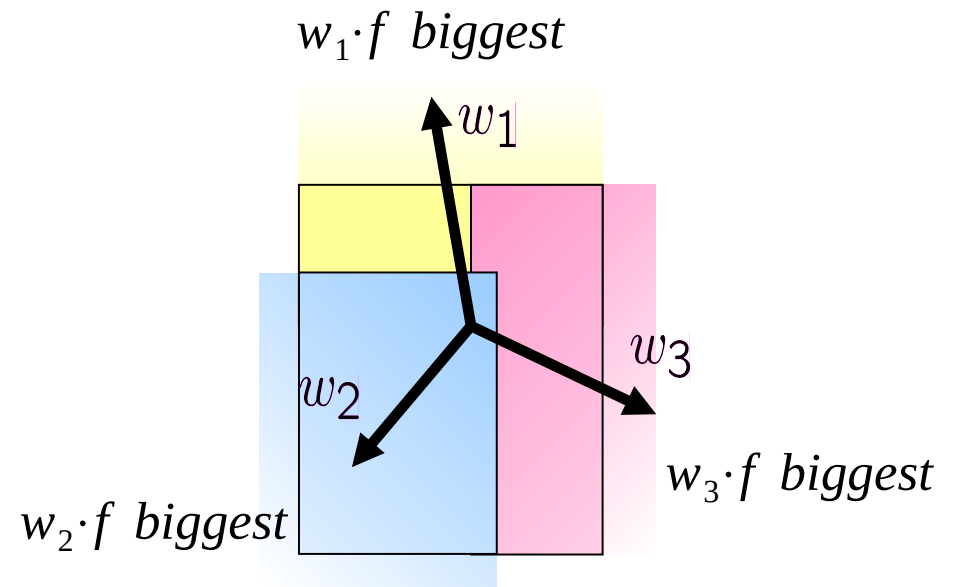


- Score (activación) de la clase y :

$$w_y \cdot f(x)$$

- Predicción con mayor score gana

$$y = \arg \max_y w_y \cdot f(x)$$



Los módulos de los vectores de pesos son distintos

Aprendizaje: Perceptron Multiclase

- Inicializar los pesos = valor random
- Por cada ejemplo
 - Clasificar empleando los pesos actuales

$$y = \arg \max_y w_y \cdot f(x)$$

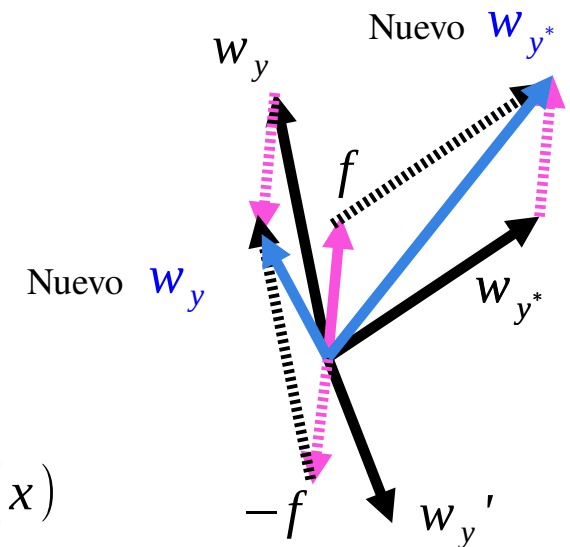
– Si **correcto**, ¡no cambiar!

– Si **erroneo**:

- decrementa el score de la incorrecta $\rightarrow w_y = w_y - f(x)$
- aumenta el de la correcta $\rightarrow w_{y^*} = w_{y^*} + f(x)$

w_{y^*} ➡ coche
 w_y ➡ moto
 $w_{y'}$ ➡ autobús

Me dice que es moto, pero sé que es coche!
Sumo a coche y resto a moto, y los de autobús no los toco



w_{y^*} Vector de pesos asociado a la clase correcta
 w_y Vector de pesos asociado a la clase predecida

Ejemplo: Perceptron Multiclase

X_1 : “ganar el voto”

$f(x_1) =$

BIAS	:	1
ganar	:	1
juego	:	0
voto	:	1
el	:	1
la	:	0
elección:		0
...		

X_2 : “ganar la elección”

$f(x_2) =$

BIAS	:	1
ganar	:	1
juego	:	0
voto	:	0
el	:	0
la	:	1
elección:		1
...		

X_3 : “ganar el juego”

$f(x_3) =$

BIAS	:	1
ganar	:	1
juego	:	1
voto	:	0
el	:	1
la	:	0
elección:		0
...		

Deporte

BIAS	:	1
ganar	:	0
juego	:	0
voto	:	0
el	:	0
la	:	0
elección:		0
...		

Política

BIAS	:	0
ganar	:	0
juego	:	0
voto	:	0
el	:	0
la	:	0
elección:		0
...		

Tecnología

BIAS	:	0
ganar	:	0
juego	:	0
voto	:	0
el	:	0
la	:	0
elección:		0
...		

Pesos iniciales: W

W^*

W'

Ejemplo: Perceptron Multiclase

$$y = w_0 * BIAS + w_1 * f(x_1) + w_2 * f(x_2) \dots$$

y BIAS = 1

X_1 : “ganar el voto” $\rightarrow 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \rightarrow 1*1 + 0*1 + 0*0 + 0*1 + 0*1 + 0*0 + 0*0 + \dots = 1$

$\rightarrow 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \rightarrow 0*1 + 0*1 + 0*0 + 0*1 + 0*1 + 0*0 + 0*0 + \dots = 0$

$\rightarrow 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \rightarrow 0*1 + 0*1 + 0*0 + 0*1 + 0*1 + 0*0 + 0*0 + \dots = 0$

Nos dice que la clase es Deporte, pero sabemos que es incorrecto! Por lo que modificaremos los pesos de Deporte y Política (que es la clase correcta)

Deporte

BIAS	:	1
ganar	:	0
juego	:	0
voto	:	0
el	:	0
la	:	0
eleccion:		0
...		

Política

BIAS	:	0
ganar	:	0
juego	:	0
voto	:	0
el	:	0
la	:	0
eleccion:		0
...		

Tecnología

BIAS	:	0
ganar	:	0
juego	:	0
voto	:	0
el	:	0
la	:	0
elección:		0
...		

Pesos iniciales: W

W*

W'

Ejemplo: Perceptron Multiclase

➤ Ajuste de los pesos de la clase Política: $w^*_0 = w^*_0 + \text{clase} * \text{bias} \dots w^*_n = w^*_n + \text{clase} * f(x_n)$

Deporte

$$\begin{aligned}w_{\text{Bias}} &= 1 + (-1) * 1 = 0 \\w_{\text{ganar}} &= 0 + (-1) * 1 = -1 \\w_{\text{juego}} &= 0 + (-1) * 0 = 0 \\w_{\text{voto}} &= 0 + (-1) * 1 = -1 \\w_{\text{el}} &= 0 + (-1) * 1 = -1 \\w_{\text{la}} &= 0 + (-1) * 0 = 0 \\w_{\text{eleccion}} &= 0 + (-1) * 0 = 0 \\&\dots\end{aligned}$$

Política

$$\begin{aligned}w^*_{\text{Bias}} &= 0 + (+1) * 1 = 1 \\w^*_{\text{ganar}} &= 0 + (+1) * 1 = 1 \\w^*_{\text{juego}} &= 0 + (+1) * 0 = 0 \\w^*_{\text{voto}} &= 0 + (+1) * 1 = 1 \\w^*_{\text{el}} &= 0 + (+1) * 1 = 1 \\w^*_{\text{la}} &= 0 + (+1) * 0 = 0 \\w^*_{\text{eleccion}} &= 0 + (+1) * 0 = 0 \\&\dots\end{aligned}$$

Ejemplo: Perceptron Multiclase

$$y = w_0 * BIAS + w_1 * f(x_1) + w_2 * f(x_2) \dots \quad \text{y } BIAS = 1$$

X_1 : “ganar el voto” $\rightarrow 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \rightarrow 0*1 + -1*1 + 0*0 + -1*1 + -1*1 + 0*0 + 0*0 + \dots = -3$

$\rightarrow 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \rightarrow 1*1 + 1*1 + 0*0 + 1*1 + 1*1 + 0*0 + 0*0 + \dots = 4$

$\rightarrow 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \rightarrow 0*1 + 0*1 + 0*0 + 0*1 + 0*1 + 0*0 + 0*0 + \dots = 0$

La clase predecida es correcta

Deporte

BIAS	:	0
ganar	:	-1
juego	:	0
voto	:	-1
el	:	-1
la	:	0
eleccion:		0
...		

Política

BIAS	:	1
ganar	:	1
juego	:	0
voto	:	1
el	:	1
la	:	0
eleccion:		0
...		

Tecnología

BIAS	:	0
ganar	:	0
juego	:	0
voto	:	0
el	:	0
la	:	0
elección:		0
...		

Pesos iniciales: W

W^*

W'

Ejemplo: Perceptron Multiclase

$$y = w_0 * BIAS + w_1 * f(x_1) + w_2 * f(x_2) + \dots \quad \text{y } BIAS = 1$$

X_2 : “ganar la elección”

→ 1 1 0 0 0 1 1 → $0*1 + -1*1 + 0*0 + -1*0 + -1*0 + 0*1 + 0*1 + \dots = -1$

→ 1 1 0 0 0 1 1 → $1*1 + 1*1 + 0*0 + 1*0 + 1*0 + 0*1 + 0*1 + \dots = 2$

→ 1 1 0 0 0 1 1 → $0*1 + 0*1 + 0*0 + 0*0 + 0*0 + 0*1 + 0*1 + \dots = 0$

La clase predecida es correcta

Deporte

BIAS	:	0
ganar	:	-1
juego	:	0
voto	:	-1
el	:	-1
la	:	0
eleccion:	:	0
...		

Política

BIAS	:	1
ganar	:	1
juego	:	0
voto	:	1
el	:	1
la	:	0
eleccion:	:	0
...		

Tecnología

BIAS	:	0
ganar	:	0
juego	:	0
voto	:	0
el	:	0
la	:	0
elección:	:	0
...		

Pesos iniciales: W

W^*

W'

Ejemplo: Perceptron Multiclase

$$y = w_0 * BIAS + w_1 * f(x_1) + w_2 * f(x_2) + \dots \quad \text{y } BIAS = 1$$

X_3 : “ganar el juego”

$$\begin{aligned} \rightarrow 1 \ 1 \ 1 \ 0 \ 1 \ 0 \ 0 &\rightarrow 0*1 + -1*1 + 0*1 + -1*0 + -1*1 + 0*0 + 0*0 + \dots = -2 \\ \rightarrow 1 \ 1 \ 1 \ 0 \ 1 \ 0 \ 0 &\rightarrow 1*1 + 1*1 + 0*1 + 1*0 + 1*1 + 0*0 + 0*0 + \dots = 3 \\ \rightarrow 1 \ 1 \ 1 \ 0 \ 1 \ 0 \ 0 &\rightarrow 0*1 + 0*1 + 0*1 + 0*0 + 0*1 + 0*0 + 0*0 + \dots = 0 \end{aligned}$$

Nos dice que la clase es Política, pero sabemos que es incorrecto! Por lo que modificaremos los pesos de Política y Deporte (que es la clase correcta)

Deporte

BIAS	:	0
ganar	:	-1
juego	:	0
voto	:	-1
el	:	-1
la	:	0
eleccion:		0
...		

Política

BIAS	:	1
ganar	:	1
juego	:	0
voto	:	1
el	:	1
la	:	0
eleccion:		0
...		

Tecnología

BIAS	:	0
ganar	:	0
juego	:	0
voto	:	0
el	:	0
la	:	0
elección:		0
...		

Pesos iniciales: W

W^*

W'

Ejemplo: Perceptron Multiclase

➤ Ajuste de los pesos: $w_0 = w_0 + \text{clase} * \text{bias}$

$$w_n = w_n + \text{clase} * f(x_n)$$

Deporte

$$\begin{aligned}w_{\text{Bias}} &= 0 + (+1) * 1 = 1 \\w_{\text{ganar}} &= -1 + (+1) * 1 = 0 \\w_{\text{juego}} &= 0 + (+1) * 1 = 1 \\w_{\text{voto}} &= -1 + (+1) * 0 = -1 \\w_{\text{el}} &= -1 + (+1) * 1 = 0 \\w_{\text{la}} &= 0 + (+1) * 0 = 0 \\w_{\text{eleccion}} &= 0 + (+1) * 0 = 0 \\&\dots\end{aligned}$$

Política

$$\begin{aligned}w_{\text{Bias}}^* &= 1 + (-1) * 1 = 0 \\w_{\text{ganar}}^* &= 1 + (-1) * 1 = 0 \\w_{\text{juego}}^* &= 0 + (-1) * 1 = -1 \\w_{\text{voto}}^* &= 1 + (-1) * 0 = 1 \\w_{\text{el}}^* &= 1 + (-1) * 1 = 0 \\w_{\text{la}}^* &= 0 + (-1) * 0 = 0 \\w_{\text{eleccion}}^* &= 0 + (-1) * 0 = 0 \\&\dots\end{aligned}$$

Ejemplo: Perceptron Multiclase

$$y = w_0 * BIAS + w_1 * f(x_1) + w_2 * f(x_2) + \dots \quad \text{y } BIAS = 1$$

X_3 : “ganar el juego”

$$\begin{aligned} \rightarrow 1 \ 1 \ 1 \ 0 \ 1 \ 0 \ 0 &\rightarrow 1*1 + 0*1 + 1*1 + -1*0 + 0*1 + 0*0 + 0*0 + \dots = 2 \\ \rightarrow 1 \ 1 \ 1 \ 0 \ 1 \ 0 \ 0 &\rightarrow 0*1 + 0*1 + -1*1 + 1*0 + 0*1 + 0*0 + 0*0 + \dots = -1 \\ \rightarrow 1 \ 1 \ 1 \ 0 \ 1 \ 0 \ 0 &\rightarrow 0*1 + 0*1 + 0*1 + 0*0 + 0*1 + 0*0 + 0*0 + \dots = 0 \end{aligned}$$

La clase predecida es correcta

Deporte

BIAS	:	1
ganar	:	0
juego	:	1
voto	:	-1
el	:	0
la	:	0
eleccion:	:	0
...		

Política

BIAS	:	0
ganar	:	0
juego	:	-1
voto	:	1
el	:	0
la	:	0
eleccion:	:	0
...		

Tecnología

BIAS	:	0
ganar	:	0
juego	:	0
voto	:	0
el	:	0
la	:	0
elección:	:	0
...		

Pesos iniciales: W

W^*

W'

Ejemplo: Perceptron Multiclase

➤ Comprobamos los ajustes de los pesos: $y = w_0 * BIAS + w_1 * f(x_1) + w_2 * f(x_2) + \dots$

X_1 : “ganar el voto” $\Rightarrow 1\ 1\ 0\ 1\ 1\ 0\ 0 \Rightarrow 1*1 + 0*1 + 1*0 + -1*1 + 0*1 + 0*0 + 0*0 + \dots = 0$
 $\Rightarrow 1\ 1\ 0\ 1\ 1\ 0\ 0 \Rightarrow 0*1 + 0*1 + -1*0 + 1*1 + 0*1 + 0*0 + 0*0 + \dots = 1$
 $\Rightarrow 1\ 1\ 0\ 1\ 1\ 0\ 0 \Rightarrow 0*1 + 0*1 + 0*0 + 0*1 + 0*1 + 0*0 + 0*0 + \dots = 0$



X_2 : “ganar la elección” $\Rightarrow 1\ 1\ 0\ 0\ 0\ 1\ 1 \Rightarrow 1*1 + 0*1 + 1*0 + -1*0 + 0*0 + 0*1 + 0*1 + \dots = 1$
 $\Rightarrow 1\ 1\ 0\ 0\ 0\ 1\ 1 \Rightarrow 0*1 + 0*1 + -1*0 + 1*0 + 0*0 + 0*1 + 0*1 + \dots = 0$
 $\Rightarrow 1\ 1\ 0\ 0\ 0\ 1\ 1 \Rightarrow 0*1 + 0*1 + 0*0 + 0*0 + 0*0 + 0*1 + 0*1 + \dots = 0$



Nos dice que la clase es Deporte, pero sabemos que es incorrecto! Por lo que modificaremos los pesos de Deporte y Política (que es la clase correcta)

Deporte

BIAS	:	1
ganar	:	0
juego	:	1
voto	:	-1
el	:	0
la	:	0
eleccion:	:	0
...		

Política

BIAS	:	0
ganar	:	0
juego	:	-1
voto	:	1
el	:	0
la	:	0
eleccion:	:	0
...		

Tecnología

BIAS	:	0
ganar	:	0
juego	:	0
voto	:	0
el	:	0
la	:	0
elección:	:	0
...		

Ejemplo: Perceptron Multiclase

➤ Ajuste de los pesos: $w_0 = w_0 + \text{clase} * \text{bias}$

$$w_n = w_n + \text{clase} * x_n$$

Deporte

$$\begin{aligned}w_{\text{Bias}} &= 1 + (-1) * 1 = 0 \\w_{\text{ganar}} &= 0 + (-1) * 1 = -1 \\w_{\text{juego}} &= 1 + (-1) * 0 = 1 \\w_{\text{voto}} &= -1 + (-1) * 0 = -1 \\w_{\text{el}} &= 0 + (-1) * 0 = 0 \\w_{\text{la}} &= 0 + (-1) * 1 = -1 \\w_{\text{eleccion}} &= 0 + (-1) * 1 = -1 \\&\dots\end{aligned}$$

Política

$$\begin{aligned}w_{\text{Bias}}^* &= 0 + (+1) * 1 = 1 \\w_{\text{ganar}}^* &= 0 + (+1) * 1 = 1 \\w_{\text{juego}}^* &= -1 + (+1) * 0 = -1 \\w_{\text{voto}}^* &= 1 + (+1) * 0 = 1 \\w_{\text{el}}^* &= 0 + (+1) * 0 = 0 \\w_{\text{la}}^* &= 0 + (+1) * 1 = 1 \\w_{\text{eleccion}}^* &= 0 + (+1) * 1 = 1 \\&\dots\end{aligned}$$

Ejemplo: Perceptron Multiclase

➤ Comprobamos los ajustes de los pesos: $y = w_0 * BIAS + w_1 * f(x_1) + w_2 * f(x_2) + \dots$

X_1 : “ganar el voto” $\Rightarrow 1\ 1\ 0\ 1\ 1\ 0\ 0 \Rightarrow 0*1 + -1*1 + 1*0 + -1*1 + 0*1 + -1*0 + -1*0 + \dots = -2$
 $\Rightarrow 1\ 1\ 0\ 1\ 1\ 0\ 0 \Rightarrow 1*1 + 1*1 + -1*0 + 1*1 + 0*1 + 1*0 + 1*0 + \dots = 3$
 $\Rightarrow 1\ 1\ 0\ 1\ 1\ 0\ 0 \Rightarrow 0*1 + 0*1 + 0*0 + 0*1 + 0*1 + 0*0 + 0*0 + \dots = 0$



X_2 : “ganar la elección” $\Rightarrow 1\ 1\ 0\ 0\ 0\ 1\ 1 \Rightarrow 0*1 + -1*1 + 1*0 + -1*0 + 0*0 + -1*1 + -1*1 + \dots = -2$
 $\Rightarrow 1\ 1\ 0\ 0\ 0\ 1\ 1 \Rightarrow 1*1 + 1*1 + -1*0 + 1*0 + 0*0 + 1*1 + 1*1 + \dots = 4$
 $\Rightarrow 1\ 1\ 0\ 0\ 0\ 1\ 1 \Rightarrow 0*1 + 0*1 + 0*0 + 0*0 + 0*0 + 0*1 + 0*1 + \dots = 0$



X_3 : “ganar el juego” $\Rightarrow 1\ 1\ 1\ 0\ 1\ 0\ 0 \Rightarrow 0*1 + -1*1 + 1*1 + -1*0 + 0*1 + -1*0 + -1*0 + \dots = 0$
 $\Rightarrow 1\ 1\ 1\ 0\ 1\ 0\ 0 \Rightarrow 1*1 + 1*1 + -1*1 + 1*0 + 0*1 + 1*0 + 1*0 + \dots = 1$
 $\Rightarrow 1\ 1\ 1\ 0\ 1\ 0\ 0 \Rightarrow 0*1 + 0*1 + 0*1 + 0*0 + 0*1 + 0*0 + 0*0 + \dots = 0$



Deporte

BIAS	:	0
ganar	:	-1
juego	:	1
voto	:	-1
el	:	0
la	:	-1
eleccion:	:	-1
...		

Política

BIAS	:	1
ganar	:	1
juego	:	-1
voto	:	1
el	:	0
la	:	1
eleccion:	:	1
...		

Tecnología

BIAS	:	0
ganar	:	0
juego	:	0
voto	:	0
el	:	0
la	:	0
elección:	:	0
...		

Ejemplo: Perceptron Multiclase

➤ Ajuste de los pesos: $w_0 = w_0 + \text{clase} * \text{bias}$

$$w_n = w_n + \text{clase} * x_n$$

Deporte

$$\begin{aligned}w_{\text{Bias}} &= 0 + (+1) * 1 = 1 \\w_{\text{ganar}} &= -1 + (+1) * 1 = 0 \\w_{\text{juego}} &= 1 + (+1) * 1 = 2 \\w_{\text{voto}} &= -1 + (+1) * 0 = -1 \\w_{\text{el}} &= 0 + (+1) * 1 = 1 \\w_{\text{la}} &= -1 + (+1) * 0 = -1 \\w_{\text{eleccion}} &= -1 + (+1) * 0 = -1 \\&\dots\end{aligned}$$

Política

$$\begin{aligned}w_{\text{Bias}}^* &= 1 + (-1) * 1 = 0 \\w_{\text{ganar}}^* &= 1 + (-1) * 1 = 0 \\w_{\text{juego}}^* &= -1 + (-1) * 1 = -2 \\w_{\text{voto}}^* &= 1 + (-1) * 0 = 1 \\w_{\text{el}}^* &= 0 + (-1) * 1 = -1 \\w_{\text{la}}^* &= 1 + (-1) * 0 = 1 \\w_{\text{eleccion}}^* &= 1 + (-1) * 0 = 1 \\&\dots\end{aligned}$$

Ejemplo: Perceptron Multiclase

➤ Comprobamos los ajustes de los pesos: $y = w_0 * BIAS + w_1 * f(x_1) + w_2 * f(x_2) + \dots$

X_1 : “ganar el voto” → 1 1 0 1 1 0 0 → $1*1 + 0*1 + 2*0 + -1*1 + 1*1 + -1*0 + -1*0 + \dots = 1$

→ 1 1 0 1 1 0 0 → $0*1 + 0*1 + -2*0 + 1*1 + -1*1 + 1*0 + 1*0 + \dots = 0$

→ 1 1 0 1 1 0 0 → $0*1 + 0*1 + 0*0 + 0*1 + 0*1 + 0*0 + 0*0 + \dots = 0$



Deporte

BIAS	:	1
ganar	:	0
juego	:	2
voto	:	-1
el	:	1
la	:	-1
eleccion:	:	-1
...		

Política

BIAS	:	0
ganar	:	0
juego	:	-2
voto	:	1
el	:	-1
la	:	1
eleccion:	:	1
...		

Tecnología

BIAS	:	0
ganar	:	0
juego	:	0
voto	:	0
el	:	0
la	:	0
elección:	:	0
...		

Ejemplo: Perceptron Multiclase

➤ Ajuste de los pesos: $w_0 = w_0 + \text{clase} * \text{bias}$

$$w_n = w_n + \text{clase} * x_n$$

Deporte

$$\begin{aligned}w_{\text{Bias}} &= 1 + (-1) * 1 = 0 \\w_{\text{ganar}} &= 0 + (-1) * 1 = -1 \\w_{\text{juego}} &= 2 + (-1) * 0 = 2 \\w_{\text{voto}} &= -1 + (-1) * 1 = -2 \\w_{\text{el}} &= 1 + (-1) * 1 = 0 \\w_{\text{la}} &= -1 + (-1) * 0 = -1 \\w_{\text{eleccion}} &= -1 + (-1) * 0 = -1 \\&\dots\end{aligned}$$

Política

$$\begin{aligned}w_{\text{Bias}}^* &= 0 + (+1) * 1 = 1 \\w_{\text{ganar}}^* &= 0 + (+1) * 1 = 1 \\w_{\text{juego}}^* &= -2 + (+1) * 0 = -2 \\w_{\text{voto}}^* &= 1 + (+1) * 1 = 2 \\w_{\text{el}}^* &= -1 + (+1) * 1 = 0 \\w_{\text{la}}^* &= 1 + (+1) * 0 = 1 \\w_{\text{eleccion}}^* &= 1 + (+1) * 0 = 1 \\&\dots\end{aligned}$$

Ejemplo: Perceptron Multiclase

➤ Comprobamos los ajustes de los pesos: $y = w_0 * BIAS + w_1 * f(x_1) + w_2 * f(x_2) + \dots$

X_1 : “ganar el voto” $\Rightarrow 1\ 1\ 0\ 1\ 1\ 0\ 0 \Rightarrow 0*1 + -1*1 + 2*0 + -2*1 + 0*1 + -1*0 + -1*0 + \dots = -3$
 $\Rightarrow 1\ 1\ 0\ 1\ 1\ 0\ 0 \Rightarrow 1*1 + 1*1 + -2*0 + 2*1 + 0*1 + 1*0 + 1*0 + \dots = 4$
 $\Rightarrow 1\ 1\ 0\ 1\ 1\ 0\ 0 \Rightarrow 0*1 + 0*1 + 0*0 + 0*1 + 0*1 + 0*0 + 0*0 + \dots = 0$



X_2 : “ganar la elección” $\Rightarrow 1\ 1\ 0\ 0\ 0\ 1\ 1 \Rightarrow 0*1 + -1*1 + 2*0 + -2*0 + 0*0 + -1*1 + -1*1 + \dots = -3$
 $\Rightarrow 1\ 1\ 0\ 0\ 0\ 1\ 1 \Rightarrow 1*1 + 1*1 + -2*0 + 2*0 + 0*0 + 1*1 + 1*1 + \dots = 4$
 $\Rightarrow 1\ 1\ 0\ 0\ 0\ 1\ 1 \Rightarrow 0*1 + 0*1 + 0*0 + 0*0 + 0*0 + 0*1 + 0*1 + \dots = 0$



X_3 : “ganar el juego” $\Rightarrow 1\ 1\ 1\ 0\ 1\ 0\ 0 \Rightarrow 0*1 + -1*1 + 2*1 + -2*0 + 0*1 + -1*0 + -1*0 + \dots = 1$
 $\Rightarrow 1\ 1\ 1\ 0\ 1\ 0\ 0 \Rightarrow 1*1 + 1*1 + -2*1 + 2*0 + 0*1 + 1*0 + 1*0 + \dots = 0$
 $\Rightarrow 1\ 1\ 1\ 0\ 1\ 0\ 0 \Rightarrow 0*1 + 0*1 + 0*1 + 0*0 + 0*1 + 0*0 + 0*0 + \dots = 0$



Deporte

BIAS	:	0
ganar	:	-1
juego	:	2
voto	:	-2
el	:	0
la	:	-1
eleccion:	:	-1
...		

Política

BIAS	:	1
ganar	:	1
juego	:	-2
voto	:	2
el	:	0
la	:	1
eleccion:	:	1
...		

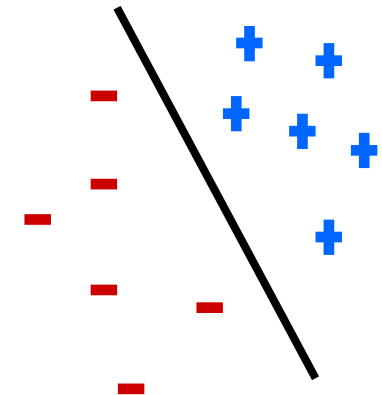
Tecnología

BIAS	:	0
ganar	:	0
juego	:	0
voto	:	0
el	:	0
la	:	0
elección:	:	0
...		

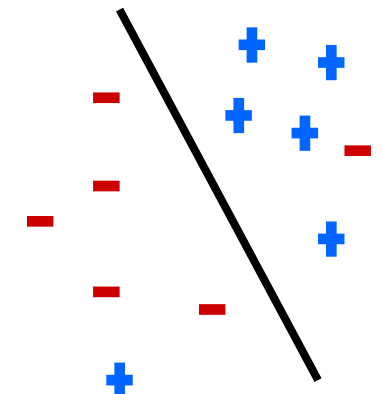
Propiedades del Perceptron

- Separabilidad: Existen vectores de pesos tal que los ejemplos del entrenamiento queden correctamente clasificados
- Convergencia: Los ejemplos del conjunto de entrenamiento son separables, el perceptron convergerá
- Limite de error: El número máximo de errores asociados al margen o grado de separabilidad

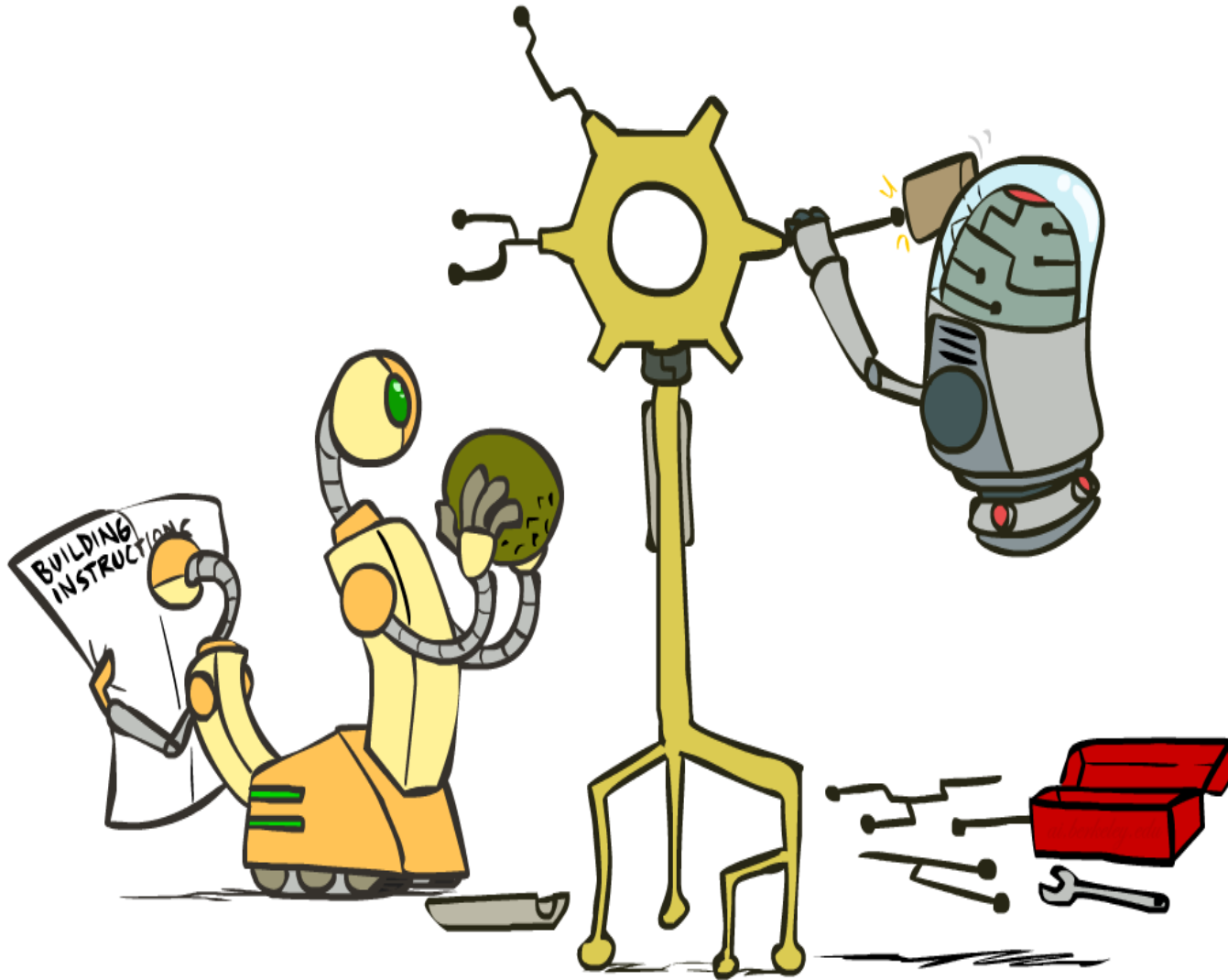
Separable



No-Separable



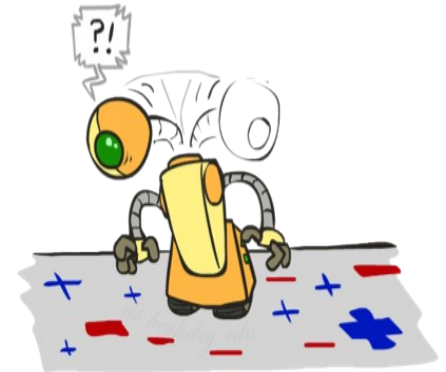
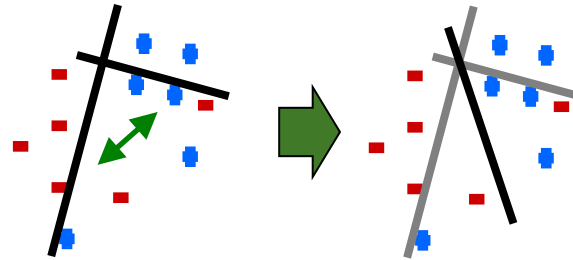
Mejorando el Perceptron



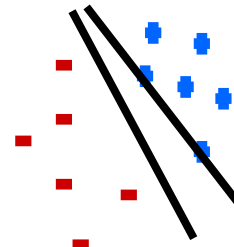
Problemas con el Perceptron

➤ Ruido: si los datos no son separables, pesos basura

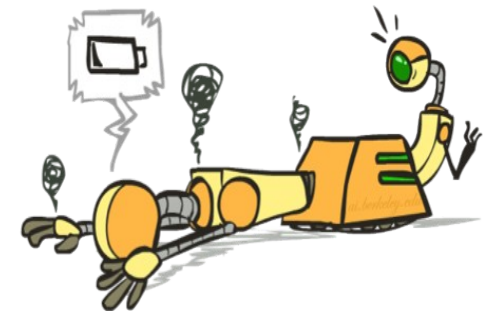
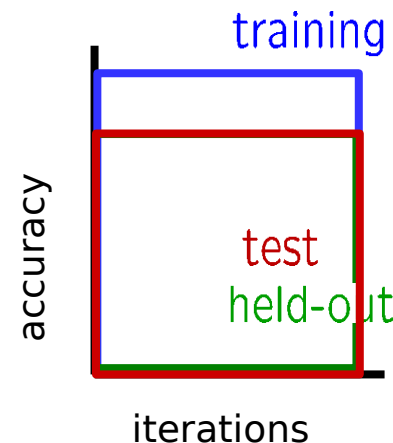
- Obtener medias de los vectores de pesos puede ayudar (averaged perceptron)



➤ Generalization Mediocre: Encuentra “una” solución separable



➤ Sobreajuste: Los resultados en el test suelen ser peores



Averaged Perceptron

➤ Teniendo en cuenta los pesos que se han calculado para el ejercicio (Deporte, Política, Tecnología):

– Pesos iniciales: $w_D=(1,0,0,0,0,0,0)$ $w_P=(0,0,0,0,0,0,0)$ $w_T=(0,0,0,0,0,0,0)$

- Tras primera iteración: **predicción errónea de Y_D**

$$w_D=(0,-1,0,-1,-1,0,0) \quad w_P=(1,1,0,1,1,0,0) \quad w_T=(0,0,0,0,0,0,0)$$

- Se almacenan las actualizaciones y al final de cada vuelta
- Pesos media:

$$W_D=((1+0+1)/3, (0-1+0)/3, (0+0+1)/3, (0-1-1)/3, (0-1+0)/3, (0-0+0)/3, (0-0+0)/3) =$$

$$w_D=(0.66,-0.33,0.33,-0.66,-0.33,0,0)$$

➤ Este proceso necesita guardar todos los w calculados en cada iteración: uso excesivo de memoria

– **Solución:** guardar solamente la media anterior:

- $w_D=(0.5,-0.5,0,-0.5,-0.5,0,0)$ $w'_D = w_D + f(x)$

$$w_D=((0.5+1)/2, (-0.5+0)/2, (0+1)/2, (-0.5-1)/2, (-0.5+0)/2, (0+0)/2, (0+0)/2) =$$

$$\Rightarrow w_D=(0.75,-0.25,0.5,-0.75,-0.25,0,0)$$

Arreglando el Perceptron

- Idea: ajustar la actualización de pesos para mitigar estos efectos adversos
- MIRA*: elegir una actualización que arregle la errónea clasificación del ejemplo actual de entrenamiento...
- ... pero, que minimice el cambio sobre w

$$\min_w \frac{1}{2} \sum_y \|w_y - w'_y\|^2$$

$$w_{y^*} \cdot f(x) \geq w_y \cdot f(x) + 1$$

- El +1 ayuda a generalizar (luchar contra el sobre ajuste)

w_y Vector de pesos asociado a la clase predecida incorrectamente
 w_{y^*} Vector de pesos asociado a la clase correcta

$$w_y = w'_y - \tau f(x)$$

$$w_{y^*} = w'_{y^*} + \tau f(x)$$

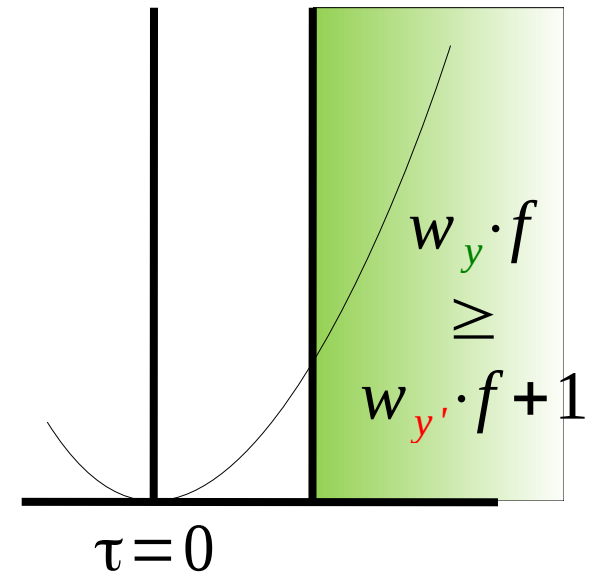
* Margin Infused Relaxed Algorithm

Actualización

Hemos clasificado el ejemplo como y' cuando la correcta debería haber sido y

1) Queremos minimizar la actualización

2) Queremos que en el siguiente paso, la clase correcta y^* (o mejor dicho el coseno entre y , f) tome mayor valor que la incorrecta y' (o mejor dicho el coseno entre y^* , f)



Min $\tau \neq 0$, o no hubiera cometido un error, así el min estará donde surge la igualdad+1

$$1) \min_w \frac{1}{2} \sum_y \|w_y - w_y^{ant}\|^2$$

$$2) w_y \cdot f(x) \geq w_{y'} \cdot f(x) + 1$$



$$\min_{\tau} \|\tau f\|^2$$

$$w_y \cdot f \geq w_{y'} \cdot f + 1$$



$$(w_y^{ant} + \tau f) \cdot f = (w_{y'}^{ant} - \tau f) \cdot f + 1$$

$$\tau = \frac{(w_{y'}^{ant} - w_y^{ant}) \cdot f + 1}{2 f \cdot f}$$

$$w_y = w_y^{ant} - \tau f(x)$$

$$w_{y'} = w_{y'}^{ant} + \tau f(x)$$



$$w_y - w_y^{ant} = -\tau f(x)$$

Actualización

¿De dónde sale la formula? $\min_w \frac{1}{2} \sum_y \|w_y - w_y^{ant}\|^2$

Asumamos que al clasificar cometemos un error $\epsilon \sim N(0,1)$ así pues $y_j + \epsilon = y_j'$ que sigue una distribución Gaussiana

$$p(y_j) = \frac{1}{\sigma \sqrt{2\pi}} \exp\left(-\frac{(y_j' - \mu)^2}{2\sigma^2}\right)$$

Siendo $\mu = y_j$ y $\sigma = 1$:

$$p(y_j) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(y_j' - y_j)^2}{2}\right)$$

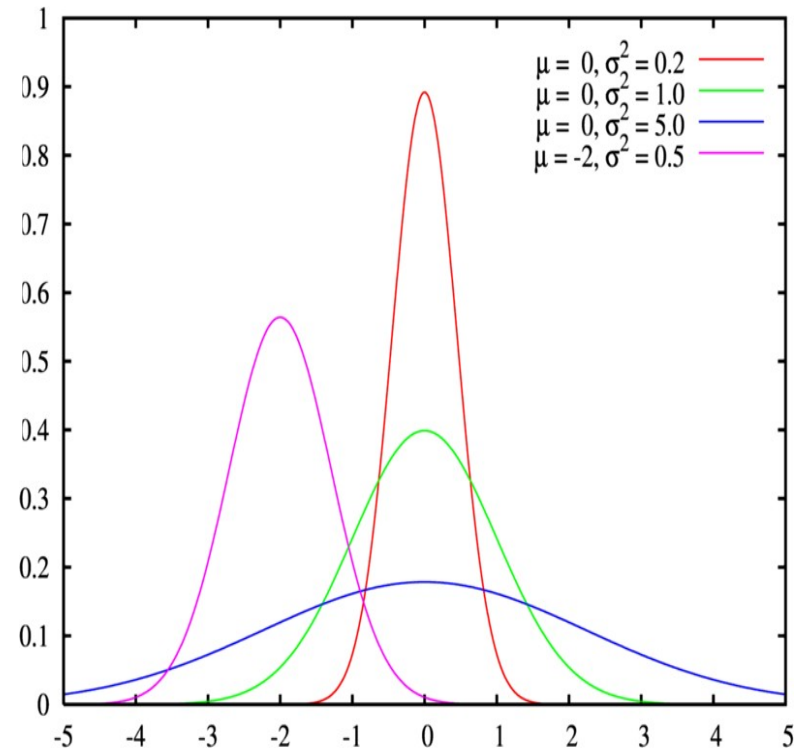
Aplico el log a ambos lados

$$\log p(y_j) = -\frac{(y_j' - y_j)^2}{2} + const$$

Esto es la probabilidad de un ejemplo,

Generalizando para todos sería la media de todos

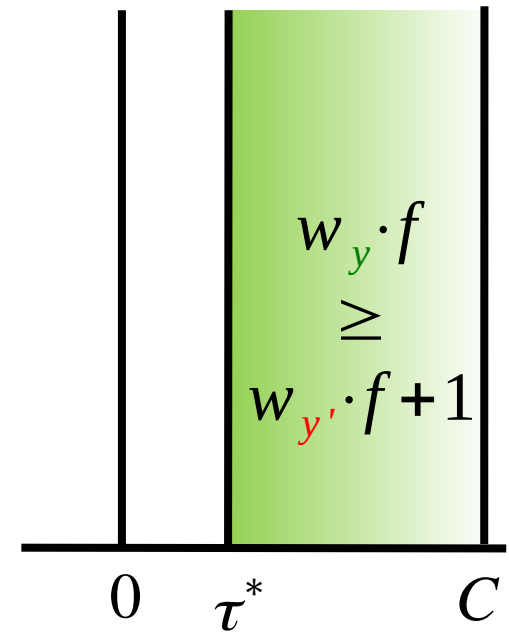
los ejemplos $MSE(y, y') = \frac{1}{2m} \sum_i (y - y')^2$



La actualización Máxima

- En la práctica hacer actualizaciones demasiado generosas no es apropiado
 - El ejemplo puede ser incorrectamente etiquetado
 - Puede que no tengas suficientes rasgos
 - Solución: capar el valor máximo de τ con alguna constante C

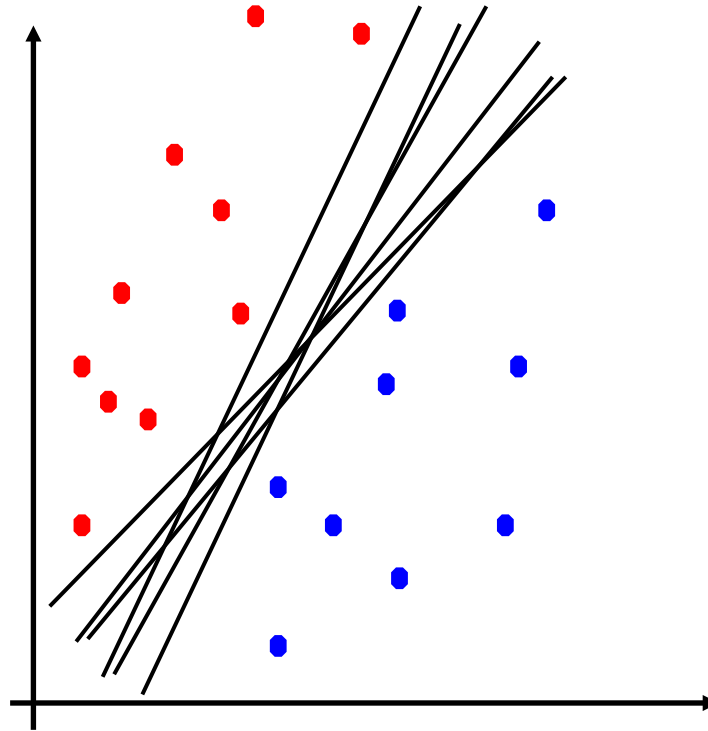
$$\tau^* = \min\left(\frac{(w_{\textcolor{red}{y}}^{\text{ant}} - w_{\textcolor{green}{y}}^{\text{ant}}) \cdot f + 1}{2f \cdot f}, C\right)$$



- Normalmente converge antes que el perceptron
- Normalmente es mejor, especialmente frente a ejemplos ruidosos

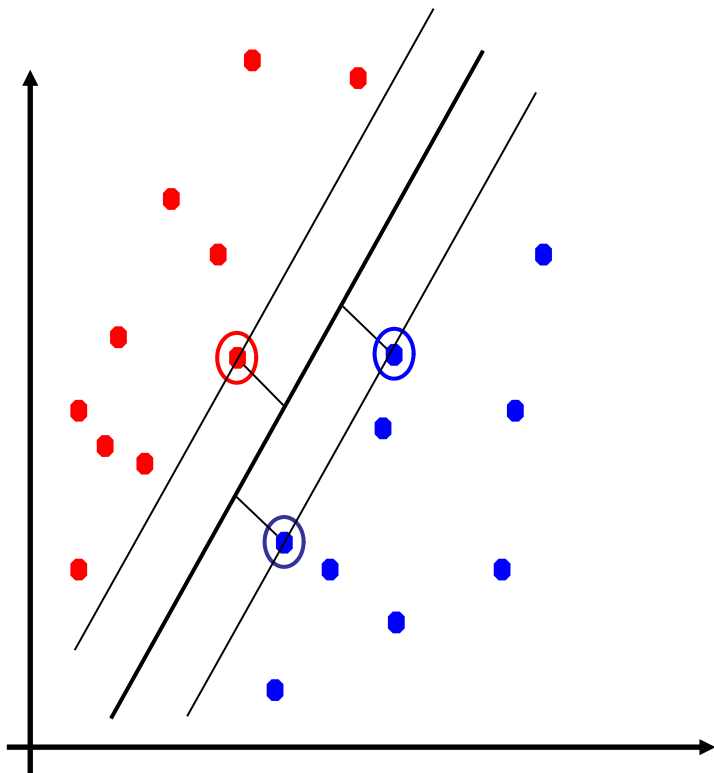
Separadores lineales

➤ ¿Cual de estos separadores lineales es el óptimo?



Support Vector Machines (SVM)

- Maximizar el margen: es intuitivo
- Solo importan los support vectors; el resto de los ejemplos de entrenamiento se ignoran
- Support vector machines (SVMs) buscan también maximizar el margen
- Básicamente, los SVM son MIRA donde la optimización (buscar el mayor margen de clasificación) se realiza sobre todos los ejemplos del entrenamiento a la vez.



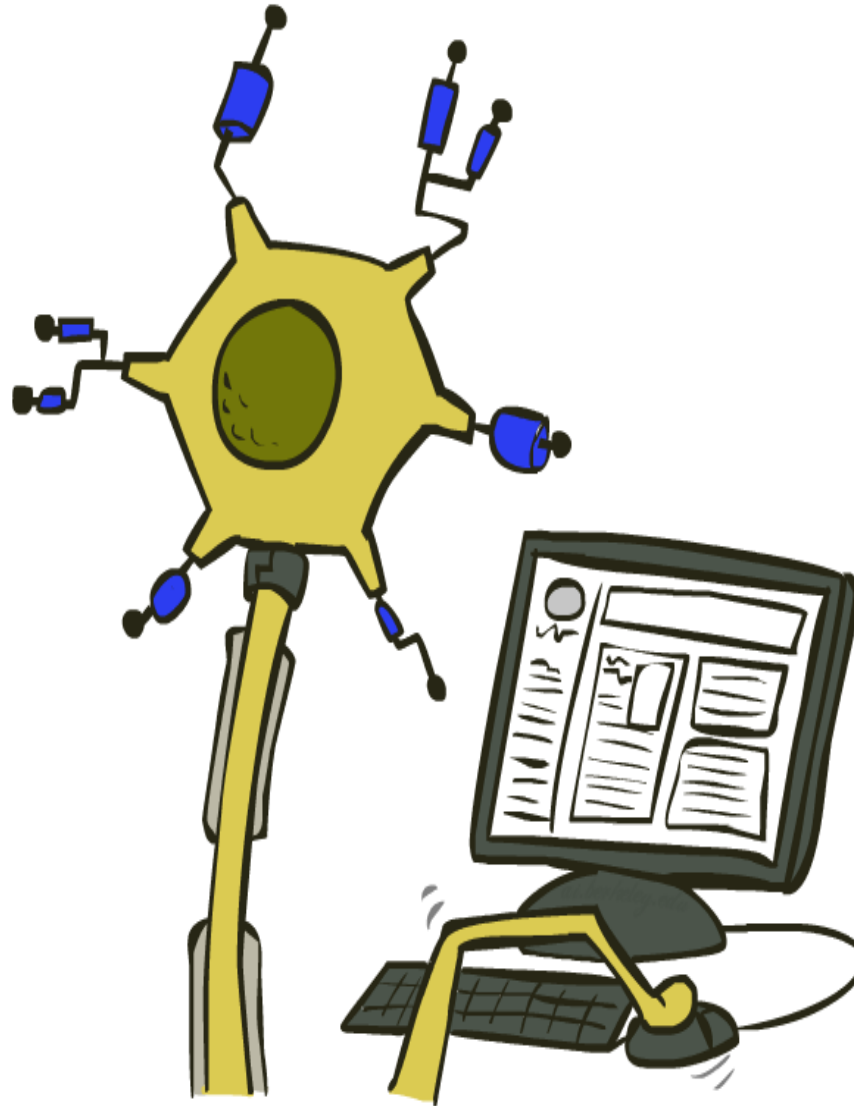
MIRA

$$\min_w \frac{1}{2} \sum_y \|w_y - w'_y\|^2$$
$$w_{y^*} \cdot f(x_i) \geq w_y \cdot f(x_i) + 1$$

SVM

$$\min_w \frac{1}{2} \|w\|^2$$
$$\forall i, y \quad w_{y^*} \cdot f(x_i) \geq w_y \cdot f(x_i) + 1$$

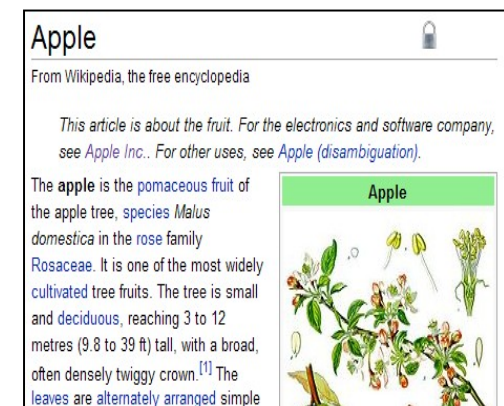
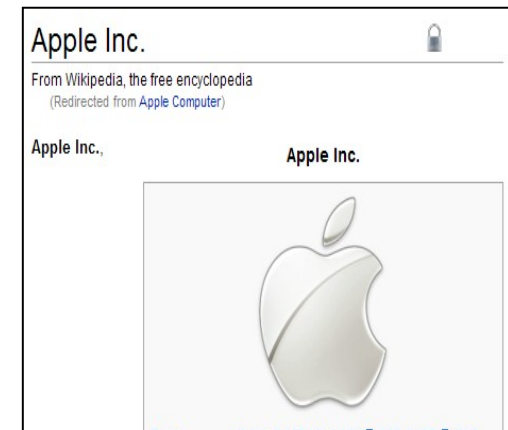
Búsqueda Web



Extensión: Búsqueda Web

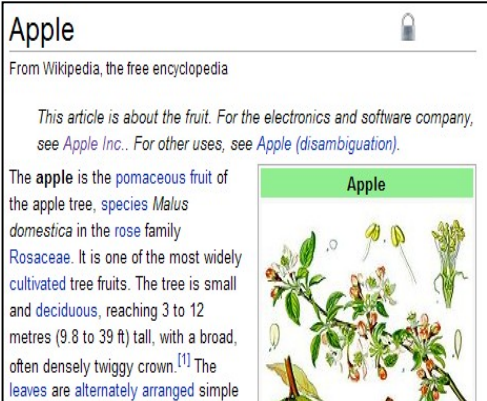
- Recuperación de Información (Information retrieval):
 - Dada una consulta, producir información relevante a esa consulta
 - Incluye, búsqueda web, question answering, y recuperación de información clásica (IR).
- Búsqueda Web: no es exactamente clasificación, sino ranking


$x = \text{“Apple Computers”}$



Ranking basado en rasgos

$x = \text{“Apple Computer”}$

$$f(x, \text{Apple}) = [0.3 \ 5 \ 0 \ 0 \ \dots]$$


$$f(x, \text{Apple Inc.}) = [0.8 \ 4 \ 2 \ 1 \ \dots]$$


Perceptron para Ranking

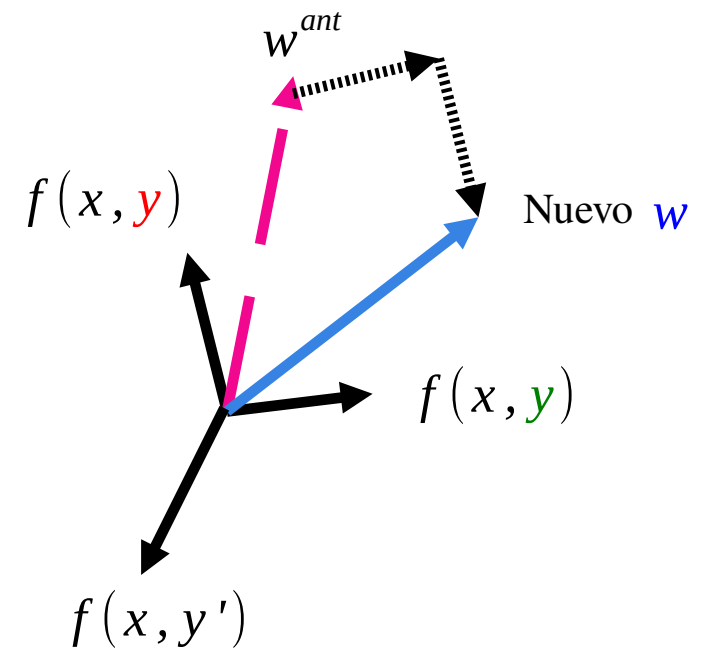
- Inputs: x
- Candidatos: y
- Muchos vectores de rasgos: $f(x, y)$
- Un vector de pesos: w

– Predicción:

$$y = \operatorname{argmax}_y w \cdot f(x, y)$$

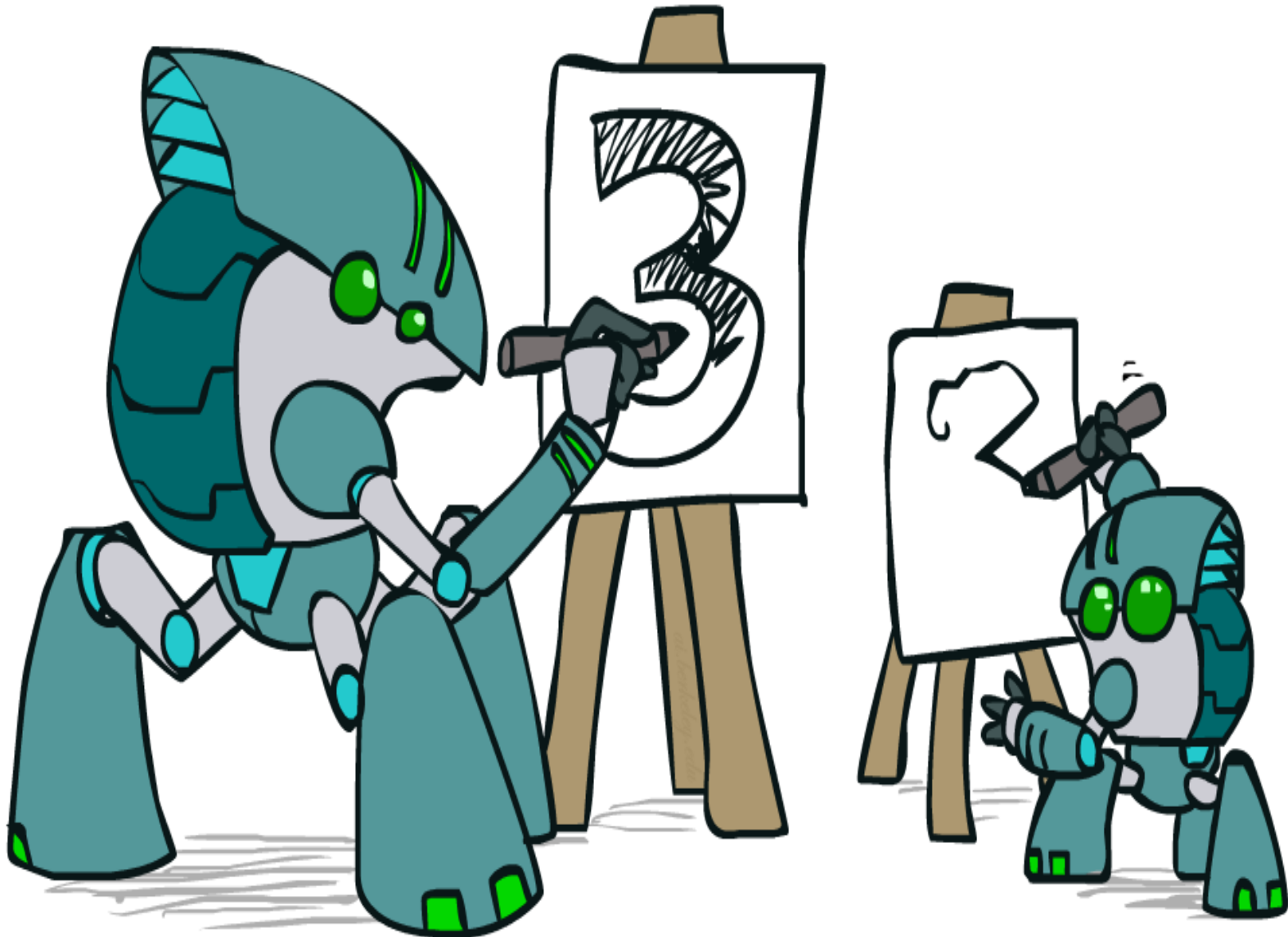
– actualización (si incorrecto):

$$w = w + f(x, y) - f(x, y')$$



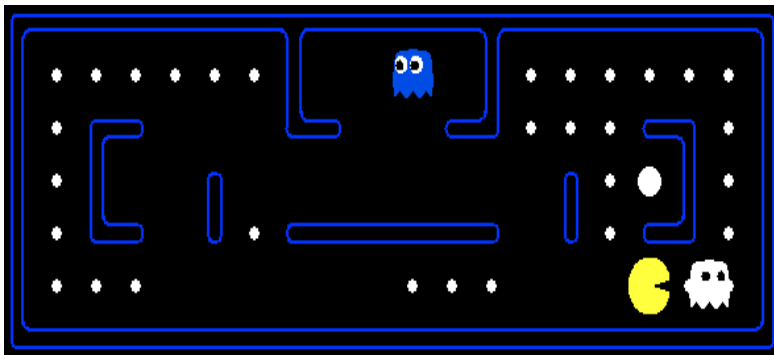
Hay solo un vector de pesos porque se aplica el mismo a todos los ejemplos dado que no hay una clase correcta o una incorrecta sino cuanto se parece cada ejemplo a la solución deseada

Aprendices



¡Aprendiz de Pacman!

- Los items son estados: S



- Los candidatos son pares estado, acción: (S, a)
- Las acciones “Correctas”: las realizadas por un experto (grabadas) $(S, \textcolor{green}{a}): f(S, \textcolor{green}{a})$
- Los rasgos se calculan por cada par (S, a)
- El score/puntuación del par se obtiene:

$$w \cdot f(S, a)$$

- Hay que realizar un ranking de las acciones, así que hay solo un vector de pesos. $w = w + f(S, \textcolor{green}{a}) - f(S, \textcolor{red}{a})$



$$\forall a \neq \textcolor{green}{a}, \\ w \cdot f(\textcolor{green}{a}) > w \cdot f(\textcolor{red}{a})$$