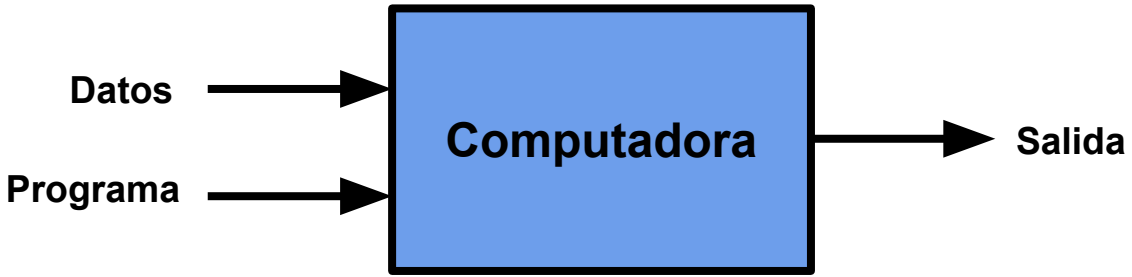
Introducción al aprendizaje automático

•••

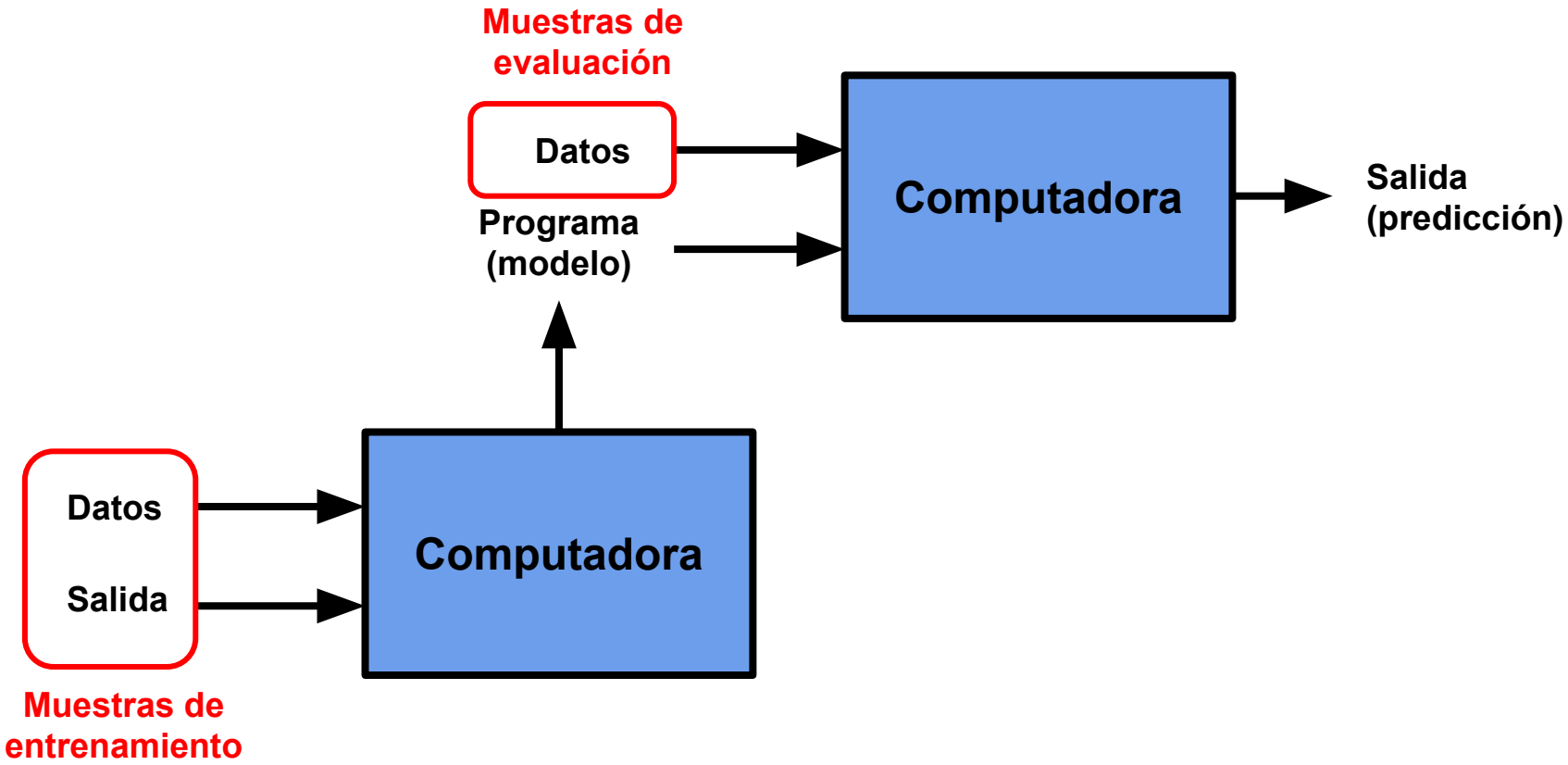
#1. Introducción al aprendizaje automático.

Programación tradicional



Aprendizaje automático

Aprendizaje automático: entrenamiento vs. evaluación



Sobre "aprendizaje"

- Se puede ver como la utilización directa o indirecta de la experiencia para aproximar una determinada función.

- La aproximación de dicha función corresponde a una búsqueda en un espacio de hipótesis (espacio de funciones) por aquella que mejor prediga el comportamiento de **datos nuevos**.

- Distintos métodos de aprendizaje automático asumen distintos espacios de hipótesis o utilizan distintas estrategias de búsqueda.

Tipos de problemas

- **Aprendizaje supervisado (inductivo)**
  Datos de entrenamiento + salida esperada

- **Aprendizaje no supervisado**
  Datos de entrenamiento (sin salida esperada)

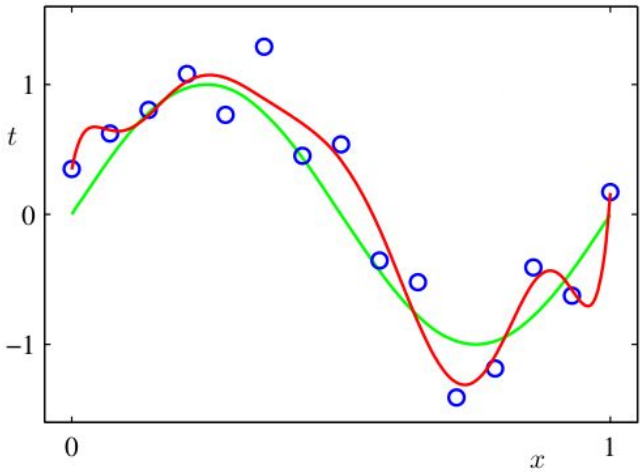
- **Aprendizaje semi-supervisado**
  Datos de entrenamiento + **pocas** salida esperadas

- **Aprendizaje auto-supervisado**
  Datos de entrenamiento auto generados (*tareas pretexto*)

- **Aprendizaje por refuerzo**
  "Recompensas" por secuencias de acciones

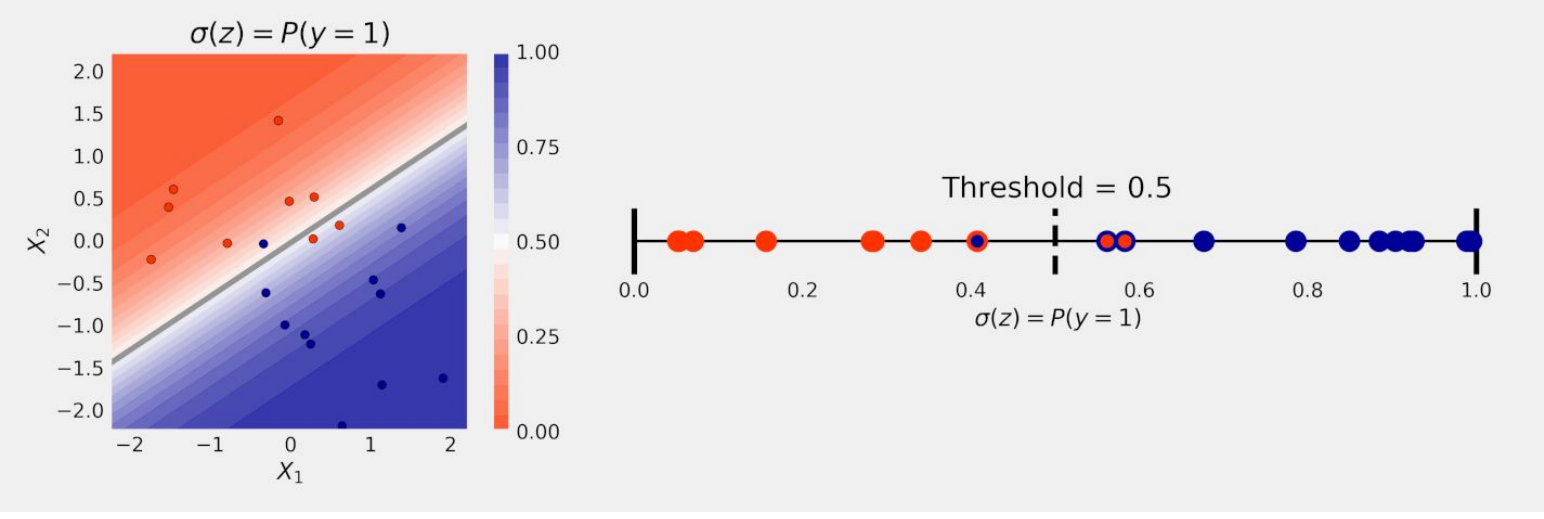
Aprendizaje supervisado: regresión

- Dados $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$
- Aprender una $f(x)$ que permita predecir y a partir de x
    - Si y está en $\mathbb{R}^n$ → **regresión**



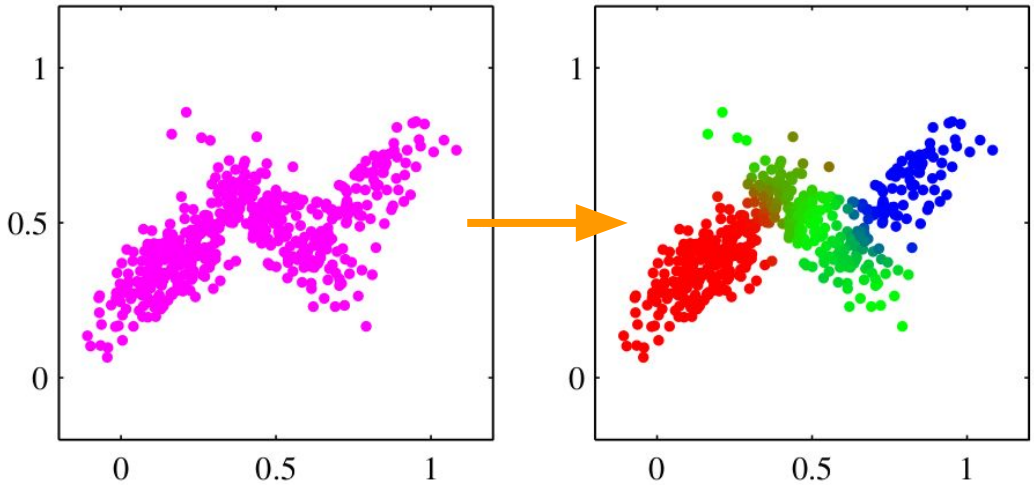
Aprendizaje supervisado: clasificación

- Dados $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$
- Aprender una $f(x)$ que permita predecir y a partir de x
    - Si y es categórica → **clasificación**



Aprendizaje no supervisado

- Dados $x_1, x_2, ..., x_n$
- Aprender la estructura interna de los datos
    - p.ej. *clustering*

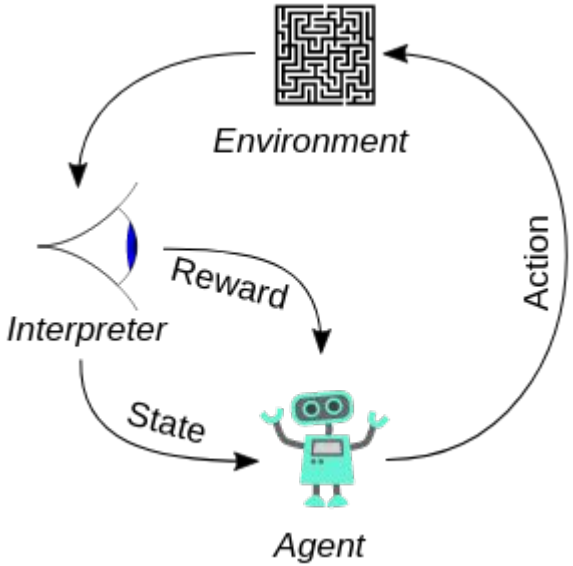


Aprendizaje auto supervisado

- Dados $x_1, x_2, ..., x_n$
- Utilizar estructura interna para generar *tareas pretexto*
    - p.ej.: predecir siguiente elemento en una secuencia
- (pre)entrenar para aprender a representar bien los datos
- Adaptar a la tarea de interés (regresión, clasificación, …)

Aprendizaje por refuerzo

- Dada una secuencia de estados y acciones con recompensa (*reward*), generar una política (*policy*) (secuencia de acciones) que nos indique qué hacer ante un determinado estado



Aprendizaje (supervisado)

Funciones $\mathcal{F}$

$f : \mathcal{X} \to \mathcal{Y}$

Datos de entrenamiento

$\{(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}\}$

Aprendizaje

find $\hat{f} \in \mathcal{F}$
s.t. $y_i \approx \hat{f}(x_i)$

Predicción

$y = \hat{f}(x)$

x

Nuevo dato

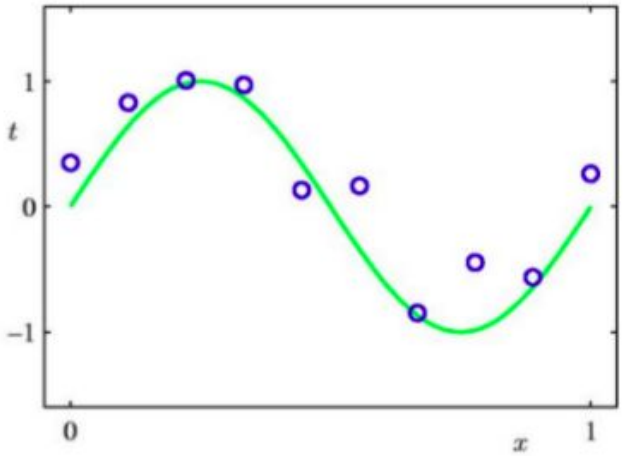
Regresión

Regresión

- Disponemos de N pares de entrenamiento (observaciones)

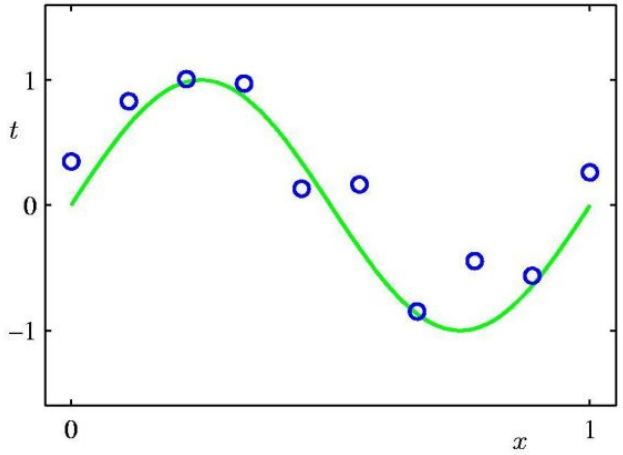
$$\{(x_i, y_i)\}_{i=1}^{N} = \{(x_1, y_1), \cdots, (x_N, y_N)\}$$

- El problema de regresión consiste en estimar *f(x)* a partir de estos datos



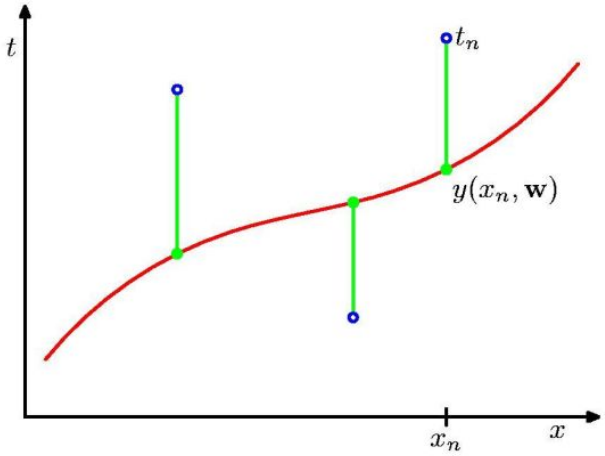
Regresión polinomial

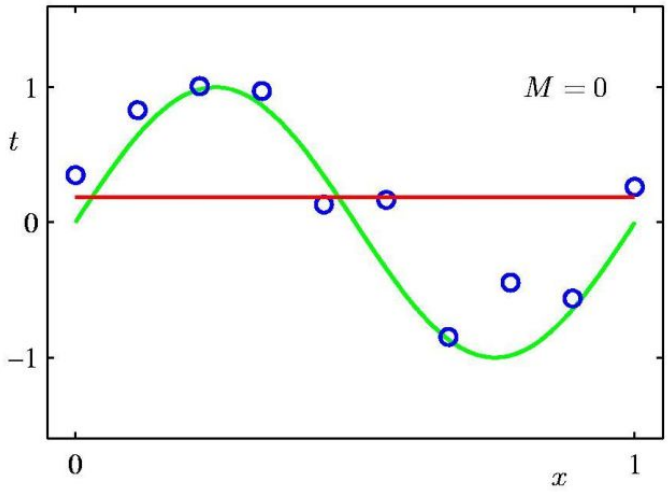
- En verde se ilustra la función "verdadera" (inaccesible)
- Las muestras son uniformes en x y poseen ruido en y
- Utilizaremos una **<u>función de costo</u>** (error cuadrático) para medir el error en la predicción de y mediante $f(x)$

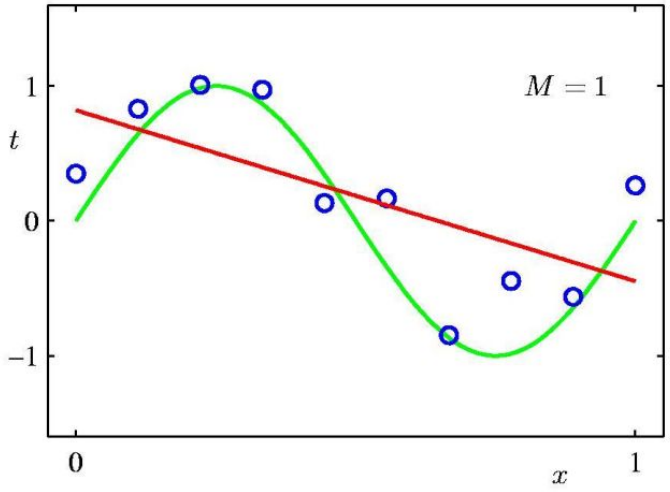


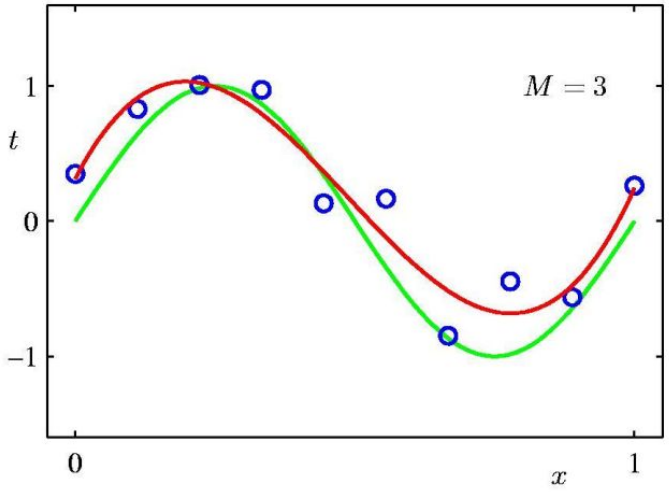
$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2$$

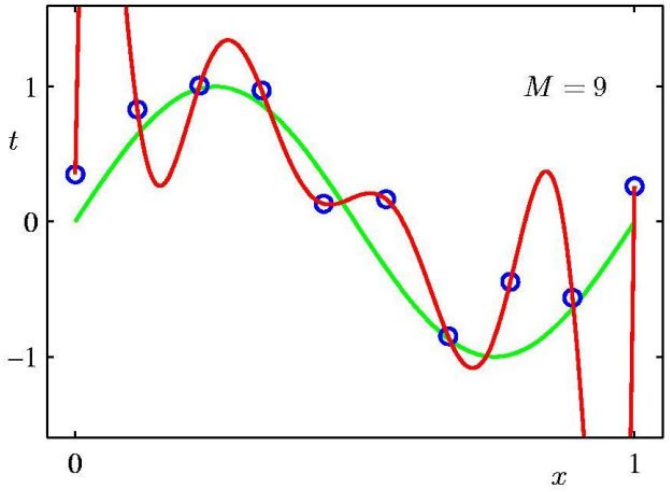
$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$




M = 0
t
x
1
0
−1
0
1

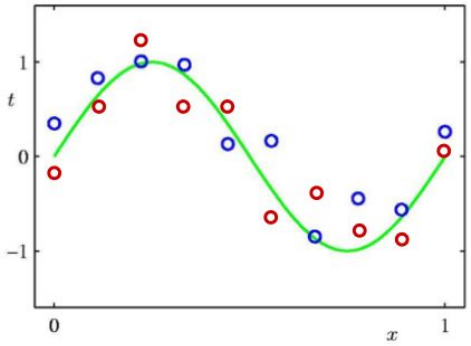

M = 1
t
x
1
0
−1
0
1

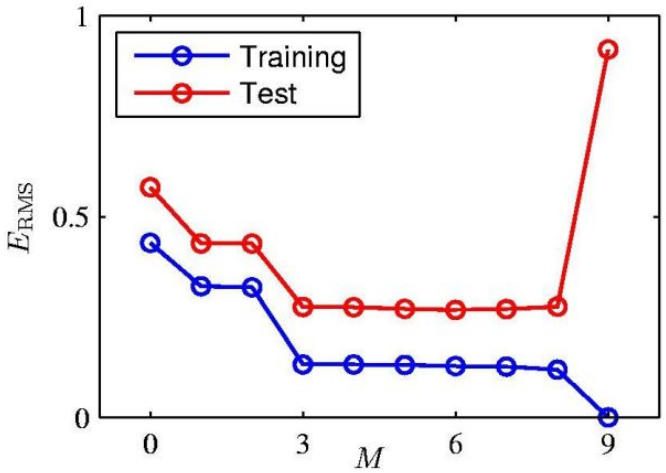

M = 3
t
x
1
0
−1
0
1


M = 9
t
x
1
0
−1
0
1

Sobreajuste (*overfitting*)

- Datos de test: otra muestra de los misma función subyacente
- El error de entrenamiento se hace cero, pero el de test crece con M





Root-Mean-Square (RMS) Error: $E_{\mathrm{RMS}} = \sqrt{2E(\mathbf{w}^{\star})/N}$

Bondad de ajuste vs. complejidad de modelo

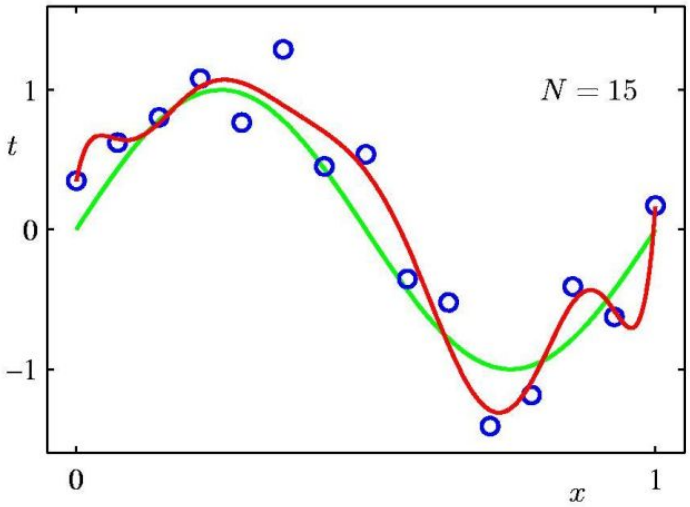
- Si el modelo tiene tantos grados de libertad como los presentes en los datos de entrenamiento, puede ajustarlos perfectamente

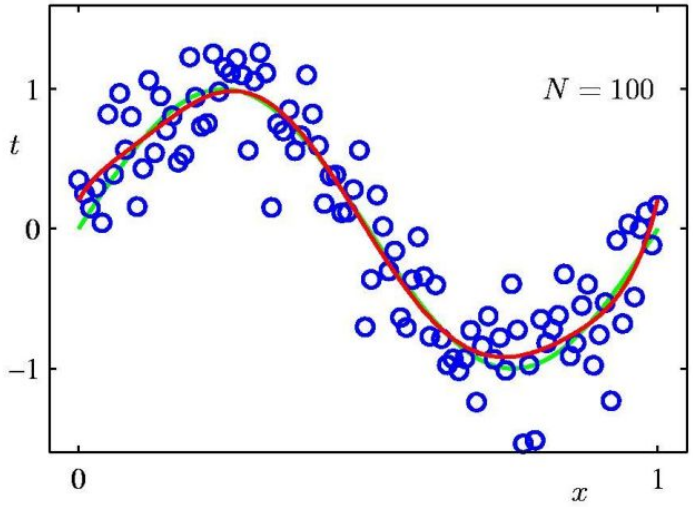
- El objetivo en aprendizaje automático no es el ajuste perfecto, sino la **generalización** a conjuntos nuevos (no vistos en entrenamiento)

- Podemos decir que un modelo generaliza, si puede explicar los datos empleando una complejidad acotada

Prevenir el sobreajuste (I)

- Agregar más datos (más que la "complejidad" del modelo)





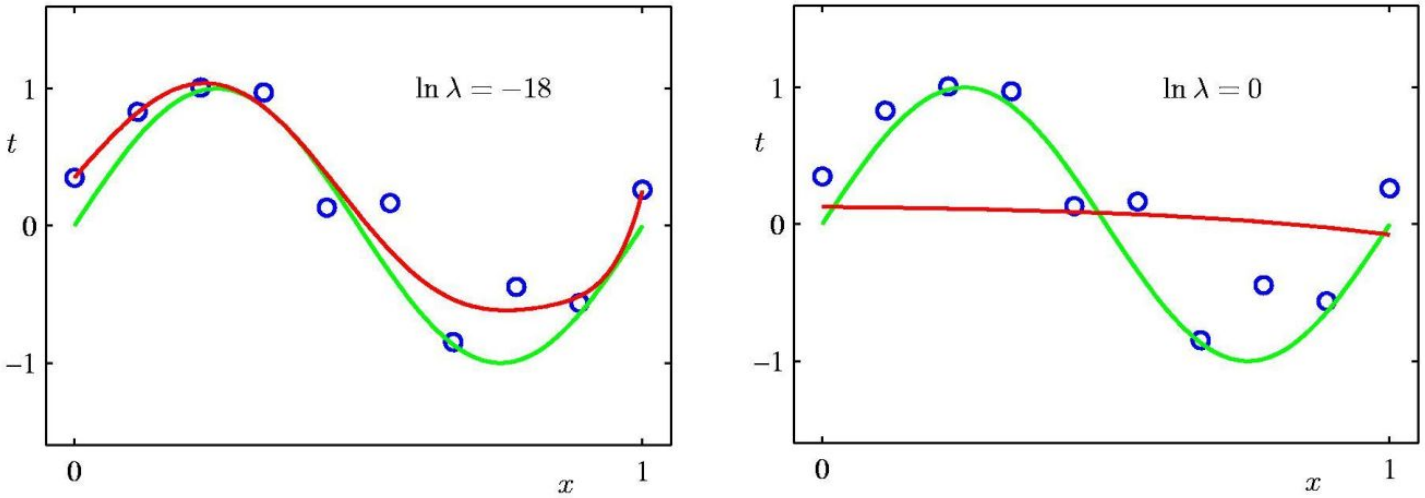
Prevenir el sobreajuste (II)

	$M=0$	$M=1$	$M=3$	$M=9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

Prevenir el sobreajuste (II)

- Regularización: penalizar valores grandes de los coeficientes

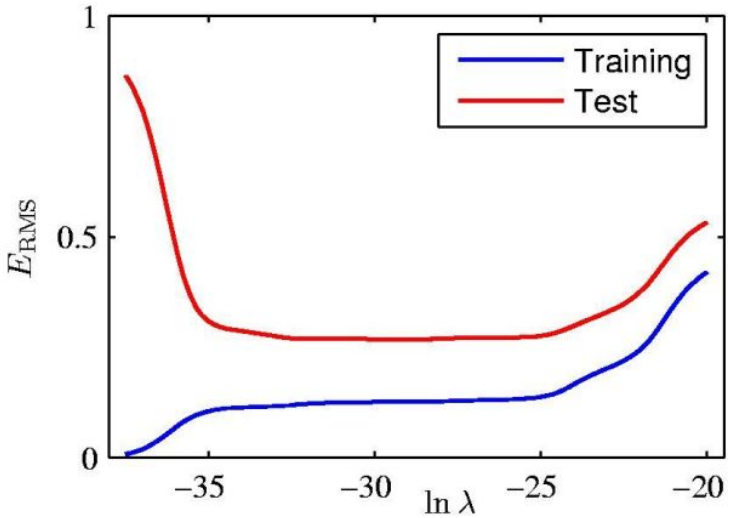
$$\widetilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$$



Prevenir el sobreajuste (II)

- Regularización: penalizar valores grandes de los coeficientes

$$\widetilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n,\mathbf{w}) - t_n\}^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$$



Término de regularización (*ridge*)

λ = hiperparámetro

Prevenir el sobreajuste (II)

	$\ln \lambda = -\infty$	$\ln \lambda = -18$	$\ln \lambda = 0$
$w_0^\star$	0.35	0.35	0.13
$w_1^\star$	232.37	4.74	-0.05
$w_2^\star$	-5321.83	-0.77	-0.06
$w_3^\star$	48568.31	-31.97	-0.05
$w_4^\star$	-231639.30	-3.89	-0.03
$w_5^\star$	640042.26	55.28	-0.02
$w_6^\star$	-1061800.52	41.32	-0.01
$w_7^\star$	1042400.18	-45.95	-0.00
$w_8^\star$	-557682.99	-91.53	0.00
$w_9^\star$	125201.43	72.68	0.01

Regresión polinomial como regresión lineal

$$x \mapsto \mathbf{z} = \begin{pmatrix} x \\ x^2 \\ \vdots \\ x^M \end{pmatrix} \qquad \mathbf{w} = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_M \end{pmatrix}$$

$$\begin{aligned} y(x; \mathbf{w}) &= w_0 + w_1 x + w_2 x^2 + \cdots + w_M x^M \\ &= w_0 + \sum_{j=1}^{M} w_j x^j = w_0 + \sum_{j=1}^{M} w_j z_j \\ &= w_0 + \mathbf{w}^T \mathbf{z} \end{aligned}$$

$$\mathbf{a}^T \mathbf{b} \equiv \langle \mathbf{a}, \mathbf{b} \rangle = \sum_{i=1}^{D} a_i b_i$$

Regresión polinomial como regresión lineal

$$x \mapsto \mathbf{z} = \begin{pmatrix} 1 \\ x \\ x^2 \\ \vdots \\ x^M \end{pmatrix} \qquad \mathbf{w} = \begin{pmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_M \end{pmatrix}$$

$$\begin{aligned} y(x; \mathbf{w}) &= w_0 + w_1 x + w_2 x^2 + \cdots + w_M x^M \\ &= \sum_{j=0}^{M} w_j x^j \\ &= \mathbf{w}^T \mathbf{z} \end{aligned}$$

Regresión lineal: solución de mínimos cuadrados

- Dataset: $\{(x_1, t_1), \cdots, (x_N, t_N)\} \mapsto \{(\mathbf{z}_1, t_1), \cdots, (\mathbf{z}_N, t_N)\}$
- Función de costo: $E(\mathbf{W}) = \frac{1}{2}\sum_{i=1}^{N}(y(x_i; \mathbf{w}) - t_i)^2 = \frac{1}{2}\sum_{i=1}^{N}(\mathbf{w}^T\mathbf{z}_i - t_i)^2$

$$\mathbf{Z} = \begin{pmatrix} — & \mathbf{z}_1^T & — \\ & \vdots & \\ — & \mathbf{z}_N^T & — \end{pmatrix} \in \mathbb{R}^{N \times M} \qquad \mathbf{y} = \begin{pmatrix} t_1 \\ \vdots \\ t_N \end{pmatrix} \in \mathbb{R}^{N \times 1} \qquad \|\mathbf{w}\|^2 = \mathbf{w}^T\mathbf{w}$$

$$E(\mathbf{w}) = (\mathbf{Z}\mathbf{w} - \mathbf{y})^T(\mathbf{Z}\mathbf{w} - \mathbf{y}) \quad \rightarrow \quad \mathbf{w}^* = (\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T\mathbf{y}$$

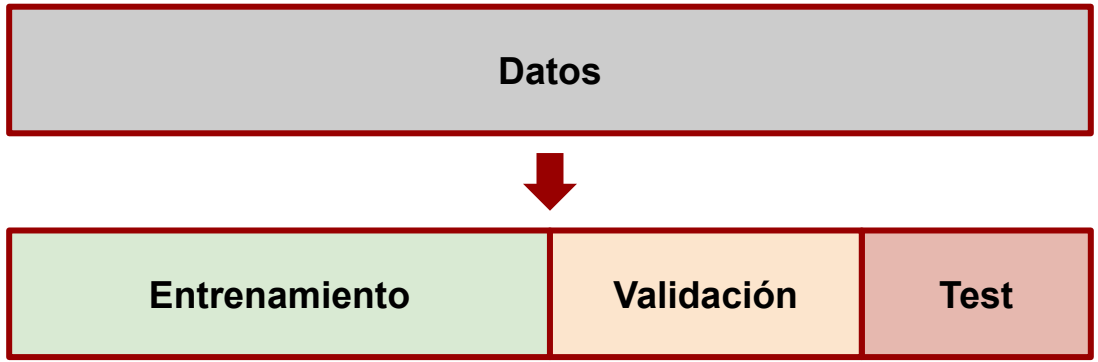
$$E(\mathbf{w}) = (\mathbf{Z}\mathbf{w} - \mathbf{y})^T(\mathbf{Z}\mathbf{w} - \mathbf{y}) + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w} \quad \rightarrow \quad \mathbf{w}^* = (\mathbf{Z}^T\mathbf{Z} + \lambda\mathbf{I})^{-1}\mathbf{Z}^T\mathbf{y}$$

Petersen & Pedersen, The matrix cookbook (2012)

Elección de *hiperparámetros*

Dividir el conjunto total de ejemplos en tres subconjuntos

- **Entrenamiento**: aprendizaje de variables del modelo
- **Validación**: ajuste/elección de hiperparámetros
- **Test**: estimación final de la performance del modelo entrenado (y con hiperparámetros elegidos adecuadamente



Clasificación

Clasificación binaria

- Disponemos de N pares de entrenamiento (observaciones)

$$\{(x_i, y_i)\}_{i=1}^{N} = \{(x_1, y_1), \cdots, (x_N, y_N)\}$$

con $x_i \epsilon \mathbb{R}^n$, $y_i \epsilon \{-1, +1\}$.

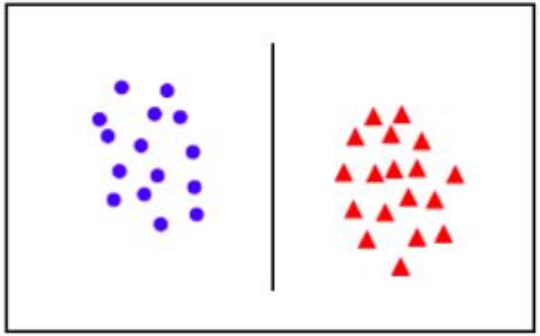
- Aprender una $f(x)$ tal que

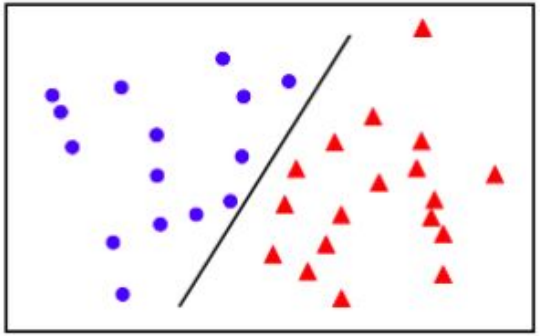
$$f(\mathbf{x}_i) \begin{cases} \geq 0 & y_i = +1 \\ < 0 & y_i = -1 \end{cases}$$

es decir: $y_i f(x_i) > 0$ para una clasificación correcta.

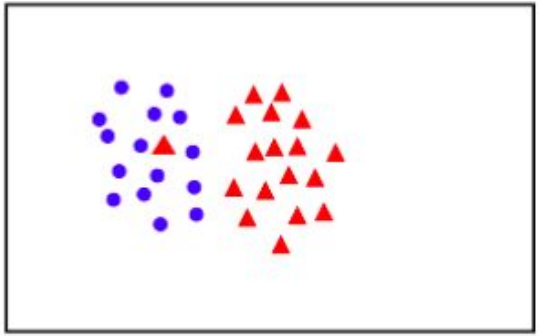
Separabilidad lineal

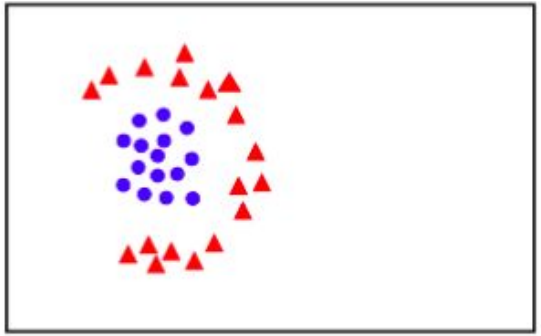
linealmente separable





no linealmente separable





Clasificadores lineales

- La entrada es un vector $\mathbf{x}_i$ de dimensionalidad n
- La salida es una etiqueta $y_i \epsilon \{-1, +1\}$
- Clasificador = función de predicción + función de decisión

$$g(f(\mathbf{x})) \rightarrow \{-1, +1\}$$

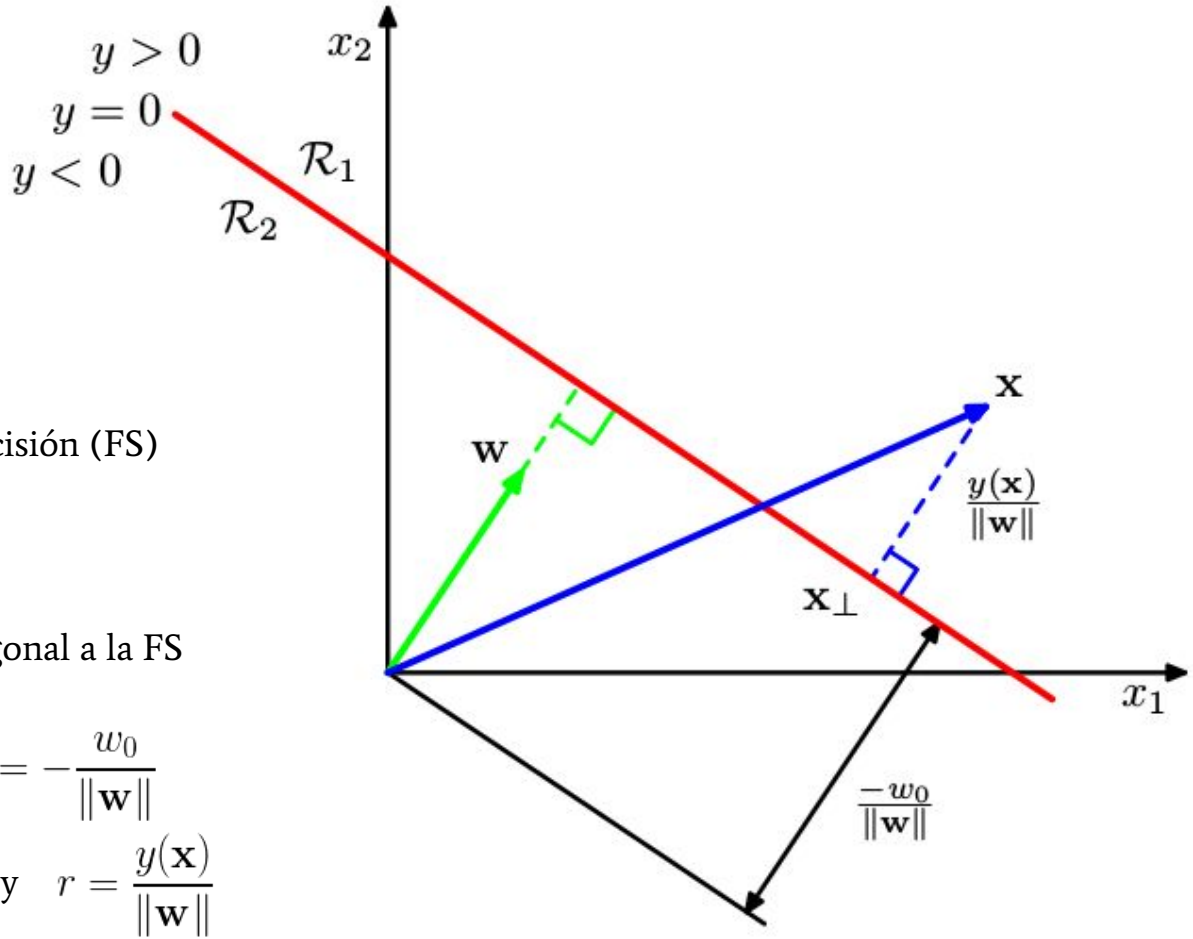
- Función de predicción **lineal**

$$f(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0$$

- Función de decisión

$$g(z) = sign(z)$$

$$g(f(\mathbf{x})) = sign(\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0)$$



$$y(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0$$

$\mathbf{x}_A, \mathbf{x}_B$: puntos en la frontera de decisión (FS)

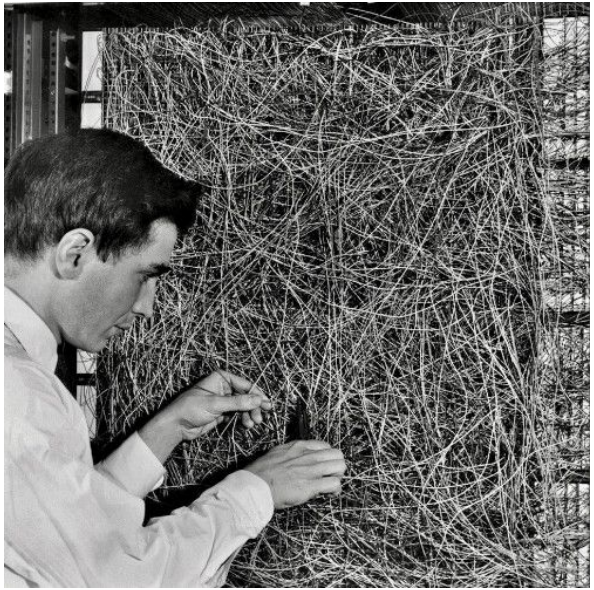
$$y(\mathbf{x}_A) = y(\mathbf{x}_B) = 0$$

$\mathbf{w}^T(\mathbf{x}_A - \mathbf{x}_B) = 0 \implies \mathbf{w}$ es ortogonal a la FS

si $\mathbf{x}$ en la FS, $\quad y(\mathbf{x}) = 0 \rightarrow \dfrac{\mathbf{w}^T\mathbf{x}}{\|\mathbf{w}\|} = -\dfrac{w_0}{\|\mathbf{w}\|}$

si $\mathbf{x} = \mathbf{x}_\perp + r\dfrac{\mathbf{w}}{\|\mathbf{w}\|} \rightarrow y(\mathbf{x}_\perp) = 0$ y $\quad r = \dfrac{y(\mathbf{x})}{\|\mathbf{w}\|}$

El algoritmo del "perceptrón"



- Propuesto por Rosemblatt en 1958
- El objetivo es encontrar un hiperplano de separación. **Si los datos son linealmente separables, lo encuentra.**
- Es un algoritmo *online* (procesa un ejemplo a la vez)
- Muchas variantes ...

El algoritmo del "perceptrón"

Entrada:

- una secuencia de pares de entrenamiento $(\mathbf{x}_1, y_1)$, $(\mathbf{x}_2, y_2)$ ...
- Una tasa de aprendizaje r (número pequeño y menor a 1)

Algoritmo:

- Inicializar $\mathbf{w}^{(0)} \epsilon \mathbb{R}^n$
- Para cada ejemplo $(\mathbf{x}_i, y_i)$
  - Predecir $y_i' = sign(\mathbf{w}^T \mathbf{x}_i)$
  - Si $y_i' \neq y_i$:

    $\mathbf{w}^{(t+1)} \leftarrow \mathbf{w}^{(t)} + r\,(y_i\,\mathbf{x}_i)$

El algoritmo del "perceptrón"

Entrada:

- una secuencia de pares de entrenamiento $(\mathbf{x}_1, y_1)$, $(\mathbf{x}_2, y_2)$ ...
- Una tasa de aprendizaje r (número pequeño y menor a 1)

Algoritmo:

- Inicializar $\mathbf{w}^{(0)} \epsilon \mathbb{R}^n$
- Para cada ejemplo $(\mathbf{x}_i, y_i)$
  - Predecir $y_i' = sign(\mathbf{w}^T \mathbf{x}_i)$
  - Si $y_i' \neq y_i$:

    $\mathbf{w}^{(t+1)} \leftarrow \mathbf{w}^{(t)} + r\,(y_i\,\mathbf{x}_i)$

El algoritmo del "perceptrón"

Entrada:

- una secuencia de pares de entrenamiento $(\mathbf{x}_1, y_1)$, $(\mathbf{x}_2, y_2)$ ...
- Una tasa de aprendizaje r (número pequeño y menor a 1)

Algoritmo:

- Inicializar $\mathbf{w}^{(0)} \epsilon \mathbb{R}^n$
- Para cada ejemplo $(\mathbf{x}_i, y_i)$
  - Predecir $y_i' = sign(\mathbf{w}^T \mathbf{x}_i)$
  - Si $y_i' \neq y_i$:

    $\mathbf{w}^{(t+1)} \leftarrow \mathbf{w}^{(t)} + r\,(y_i\,\mathbf{x}_i)$

Actualiza solo cuando comete un error

Error en positivos:

$$\mathbf{w}^{(t+1)} \leftarrow \mathbf{w}^{(t)} + r\,\mathbf{x}_i$$

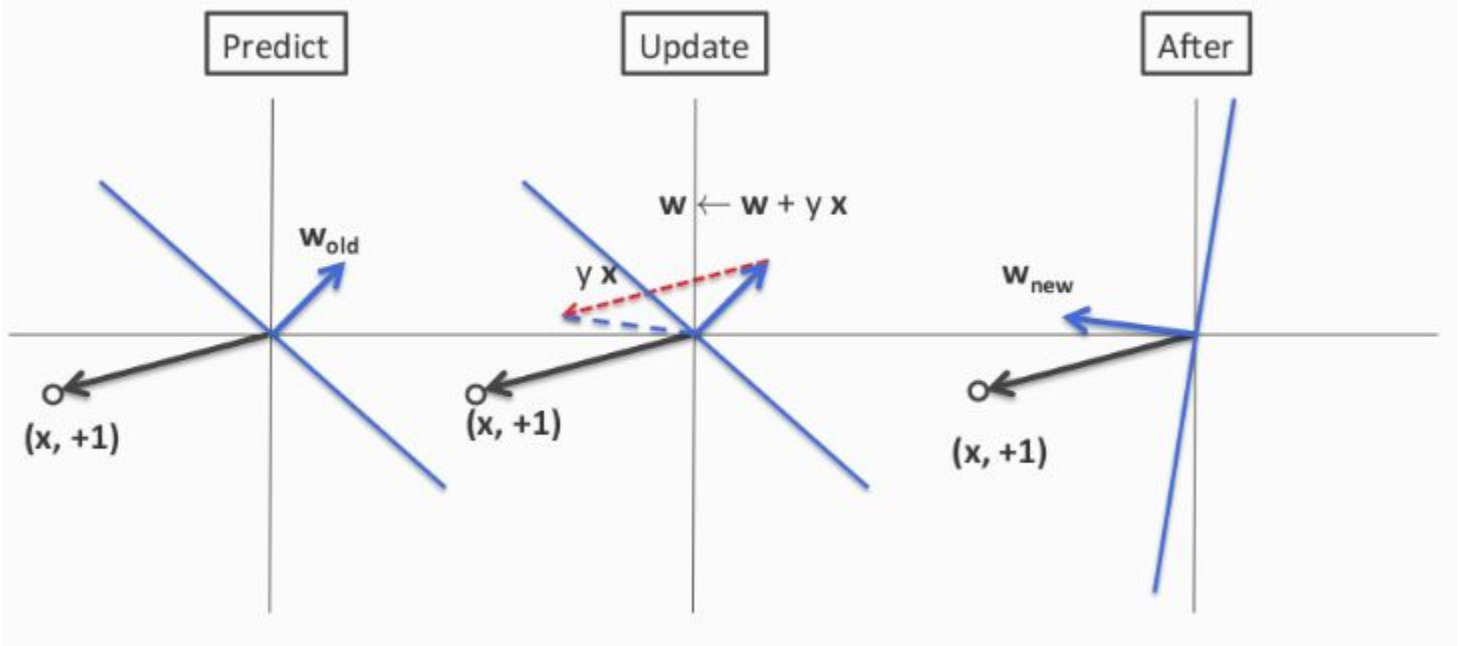
Error en negativos:

$$\mathbf{w}^{(t+1)} \leftarrow \mathbf{w}^{(t)} - r\,\mathbf{x}_i$$

Si $y_i \mathbf{w}^T \mathbf{x}_i \leq 0 \rightarrow$ error

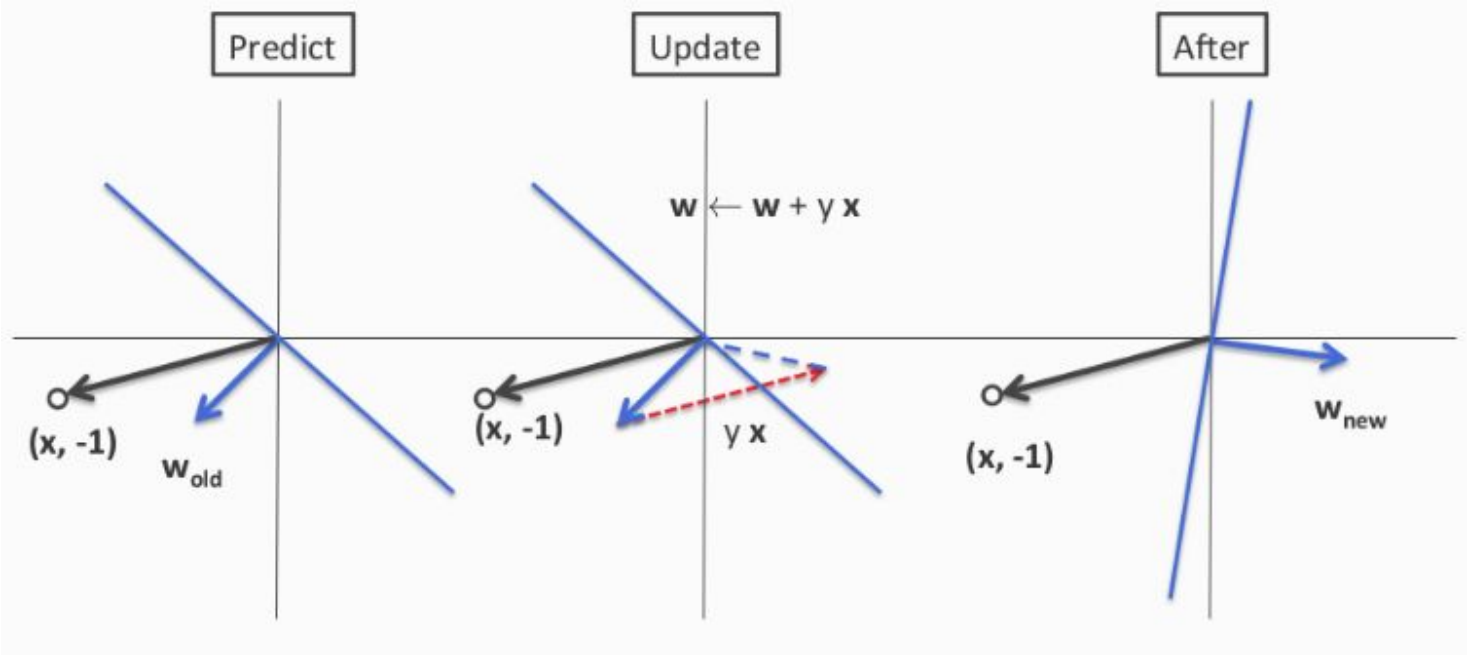
Dinámica de actualización

Error en ejemplo **positivo**:



Dinámica de actualización

Error en ejemplo **negativo**:



El algoritmo "estándar"

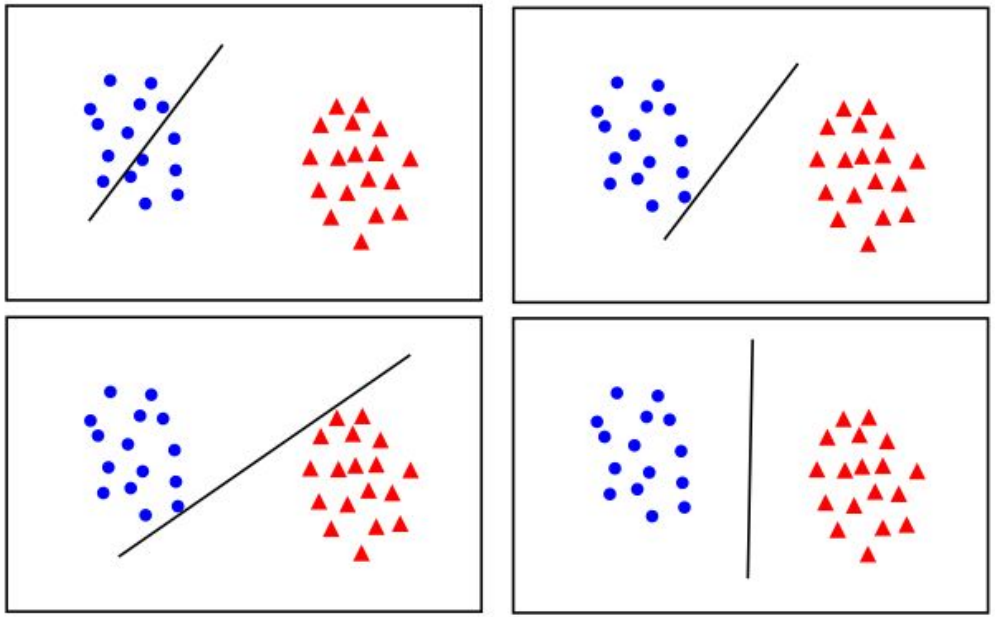
Dado un conjunto $\mathrm{D}=\{(\mathbf{x}_i, y_i)$, i=1, ..., N$\}$, $y_i \in \{-1, +1\}$, taza de entrenamiento r y número de épocas T

1. Inicializar $\mathbf{w}^{(0)}$
2. Para época t=1, …, T
   a. *barajar* el conjunto de entrenamiento D
   b. Para cada muestra de entrenamiento $(\mathbf{x}_i, y_i) \in \mathrm{D}$
      - si $y_i \mathbf{w}^{(t)\mathrm{T}} \mathbf{x}_i \leq 0$, actualizar $\mathbf{w}^{(t+1)} \leftarrow \mathbf{w}^{(t)} + r\,(y_i\,\mathbf{x}_i)$
3. Retornar $\mathbf{w}^{(\mathrm{T})}$

r, T : hiperparámetros

Predicción: *sgn*$(\mathbf{w}^{\mathrm{T}}\mathbf{x})$

¿Cuál es el mejor w?



Solución de **margen máximo**: el hiperplano más estable ante perturbaciones de la entrada

Generalización en clasificación

- Complejidad del modelo ⇔ complejidad de la frontera de decisión

Regresión

Clasificación