

## 5. Fundamentos de la Regresión Lineal

Copyright (c) 2025 Adrián Quiroga Linares Lectura y referencia permitidas; reutilización y plagio prohibidos

### 5.1 ¿Qué es el Aprendizaje Automático?

Un programa informático **aprende** de la **experiencia E** en relación a una **tarea T**, utilizando una **medida de rendimiento P**, si mejora sus prestaciones, medidas mediante **P**, en la realización de la tarea **T** a través de la experiencia **E**.

Esto suena complicado, pero es simple. Vamos a desglosarlo:

#### Ejemplo: Programa que juega a las damas

Componente	Significado	En el ejemplo de damas
<b>T (Tarea)</b>	Lo que queremos que haga el programa	Jugar a las damas
<b>E (Experiencia)</b>	Los datos o situaciones con las que aprende	Jugar muchas partidas
<b>P (Performance/Rendimiento)</b>	Cómo medimos si lo hace bien	Probabilidad de ganar la próxima partida

**En resumen:** El programa **aprende a jugar mejor a las damas** (T) jugando muchas partidas (E), y sabemos que aprende porque cada vez gana más veces (P).

### 5.2 ¿Por qué es importante el Aprendizaje Automático?

1. **Hacer viables ciertas aplicaciones** que serían imposibles de programar manualmente
  - Ejemplo: Reconocimiento facial con millones de variaciones
2. **Construir una IA de propósito general**
  - En lugar de programar reglas específicas, el sistema aprende de forma general
3. **Avances tecnológicos actuales:**
  - Mayor potencia de cálculo
  - Mayor capacidad de almacenamiento
  - Disponibilidad masiva de datos
  - Mejores algoritmos

## 5.3 Estrategias de Aprendizaje

Imagina que estás enseñando a un niño a identificar frutas:

### 5.3.1 Aprendizaje Supervisado

**Definición:** Durante el entrenamiento, le dices al sistema **exactamente qué respuesta es correcta** para cada ejemplo.

**Analogía:** Como un profesor que corrige un examen mostrando la respuesta correcta.

**Ejemplo:**

```
Entrada: [Imagen de manzana] → Etiqueta: "Manzana" ✓
Entrada: [Imagen de naranja] → Etiqueta: "Naranja" ✓
Entrada: [Imagen de plátano] → Etiqueta: "Plátano" ✓
```

El sistema aprende: "Cuando vea esta forma y color → es una manzana"

### 5.3.2 Aprendizaje No Supervisado

**Definición:** Le das datos al sistema **SIN etiquetas**, y él debe encontrar patrones por sí mismo.

**Analogía:** Como darle a un niño una caja de botones y pedirle que los agrupe como quiera.

**Ejemplo:**

Le das: [🍎, 🍊, 🍏, 🍌, 🍊, 🍏]

El sistema agrupa:

Grupo 1 (rojos, redondos): 🍏🍎🍏

Grupo 2 (naranjas, redondos): 🍊🍊

Grupo 3 (amarillos, alargados): 🍌🍌

**Aplicaciones reales:**

- Segmentación de clientes (agrupar clientes similares)
- Detección de anomalías
- Compresión de datos

### 5.3.3 Aprendizaje por Refuerzo

**Definición:** El sistema recibe **señales de recompensa o castigo** según sus acciones, pero no se le dice explícitamente qué hacer.

**Analogía:** Como entrenar a un perro con premios cuando hace algo bien.

**Ejemplo: Robot aprendiendo a caminar**

Acción: Da un paso hacia adelante

Resultado: No se cae

Señal: +10 puntos (recompensa)

Acción: Se inclina demasiado

Resultado: Se cae

Señal: -50 puntos (castigo)

**Aplicaciones reales:**

- Videojuegos (AlphaGo)
- Robots industriales
- Vehículos autónomos

## 5.4 Tipos de Problemas en Aprendizaje Supervisado

### 5.4 Problemas de Regresión

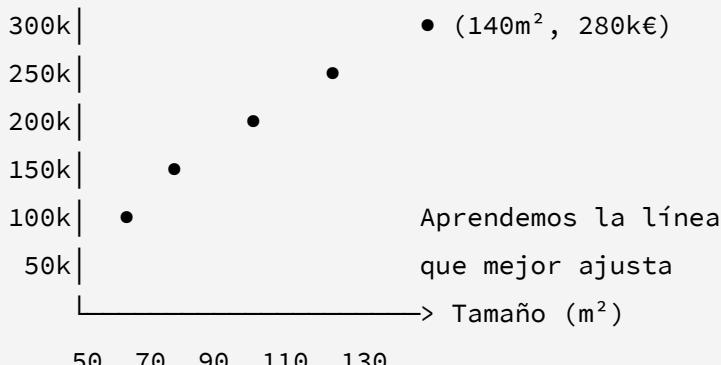
**Objetivo:** Predecir un **valor numérico continuo**.

**Definición simple:** Encontrar la función (curva/línea) que mejor se ajuste a los datos.

**Ejemplo visual:**

Precio de casas según tamaño

Precio (€)



**Lo que aprende el algoritmo:** La línea que mejor predice el precio según el tamaño.

**Pregunta típica:** "¿Cuánto costará una casa de 120m<sup>2</sup>?" → Respuesta: ~250,000€

**Otros ejemplos:**

- Predecir temperatura mañana
- Estimar ventas del próximo mes
- Predecir edad de una persona por su foto

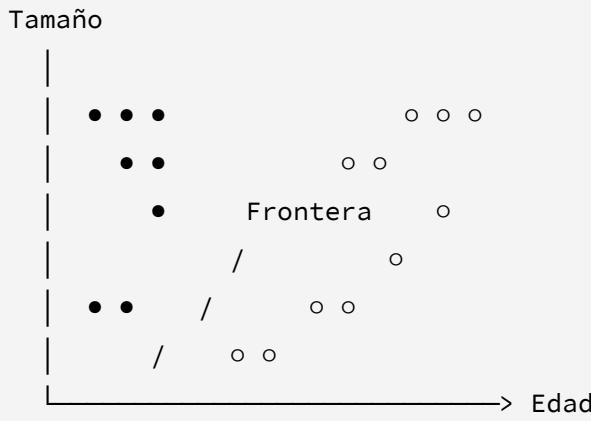
## 5.4.2 Problemas de Clasificación

**Objetivo:** Asignar datos a **categorías discretas** (clases).

**Definición simple:** Encontrar la frontera que separa diferentes grupos.

**Ejemplo visual:**

Clasificar tumores: benigno vs maligno



● = Benigno (Clase 0)

○ = Maligno (Clase 1)

**Pregunta típica:** "¿Este tumor es benigno o maligno?" → Respuesta: Clase (benigno/maligno)

**Otros ejemplos:**

- Email: spam/no spam
- Imagen: gato/perro/pájaro
- Transacción: fraude/legítima

## 5.4.3 Importancia de las Características

Aumentar el número de características **relevantes** mejora el aprendizaje.

**Ejemplo de clasificación de tumores:****Con 1 característica** (solo tamaño):

Difícil separar benignos de malignos

**Con 2 características** (tamaño + edad):

Mejor separación

**Con 3 características** (tamaño + edad + densidad):Separación aún mejor 

## 5.5 Regresión Lineal

La regresión lineal es el algoritmo más básico de aprendizaje supervisado.

### 5.5.1 Componentes del Sistema

**Notación estándar:**

Símbolo	Significado
<b>m</b>	Número de ejemplos de entrenamiento
<b>n</b>	Número de características (features)
<b>x</b>	Variables de entrada / características
<b>y</b>	Variable de salida / respuesta
<b>(x, y)</b>	Un par entrada-salida genérico
<b>(x<sup>(i)</sup>, y<sup>(i)</sup>)</b>	El i-ésimo par entrada-salida
<b>x<sup>(i)</sup><sub>j</sub></b>	Valor de la característica j en el ejemplo i

### 5.5.2 Modelo de Aprendizaje

**Flujo del proceso:**

1. Conjunto de Entrenamiento  
↓
2. Algoritmo de Aprendizaje  
↓
3. Hipótesis  $h$  (función aprendida)  
↓
4. Para nueva entrada  $x \rightarrow h$  predice  $y$

## 5.3 La Hipótesis $h$

**Forma de la hipótesis** (regresión lineal simple):

$$y = h_{\theta}(x) = \theta_0 + \theta_1 \cdot x$$

**Componentes:**

- $\theta_0$  (theta cero): Intercepto (valor cuando  $x=0$ )
- $\theta_1$  (theta uno): Pendiente (inclinación de la recta)

**Ejemplo numérico:**

$$h_{\theta}(x) = 50,000 + 2,000 \cdot x$$

Interpretación:

- Casa de 0 m<sup>2</sup>: 50,000€ (base)
- Por cada m<sup>2</sup> adicional: +2,000€
- Casa de 100 m<sup>2</sup>: 50,000 + 2,000×100 = 250,000€

## 5.5.4 Función de Coste $J(\theta)$

**Objetivo:** Medir qué tan bien se ajusta nuestra línea a los datos.

**Concepto:** Queremos que nuestra predicción  $h_{\theta}(x)$  esté lo más cerca posible del valor real  $y$ .

**Error Cuadrático Medio (MSE):**

$$J(\theta_0, \theta_1) = \frac{1}{2m} \times \sum (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

**Desglosando la fórmula:**

1.  $h_{\theta}(x^{(i)})$ : Predicción para el ejemplo i
2.  $y^{(i)}$ : Valor real del ejemplo i
3.  $(h_{\theta}(x^{(i)}) - y^{(i)})$ : Error en el ejemplo i
4. (...)<sup>2</sup>: Elevamos al cuadrado (penaliza errores grandes)

5.  $\Sigma$ : Sumamos errores de todos los ejemplos
6.  $\frac{1}{2m}$ : Promediamos (el  $\frac{1}{2}$  simplifica derivadas)

### Ejemplo numérico:

Datos reales:

```
x = [50, 100, 150]
y = [150k, 250k, 350k]
```

Nuestra línea:  $h_{\theta}(x) = 50k + 2k \cdot x$

Predicciones:

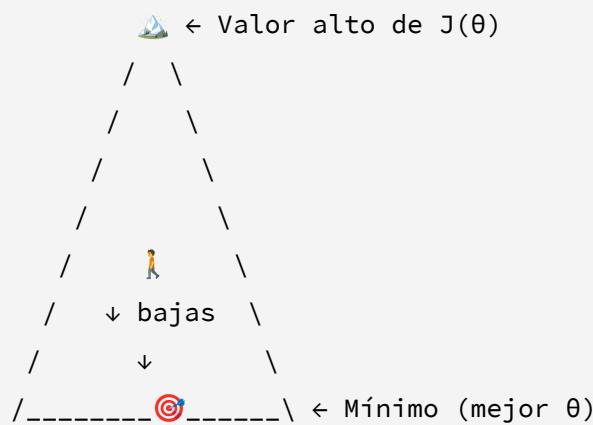
```
 $h_{\theta}(50) = 150k \rightarrow \text{Error} = (150k - 150k)^2 = 0$ 
 $h_{\theta}(100) = 250k \rightarrow \text{Error} = (250k - 250k)^2 = 0$ 
 $h_{\theta}(150) = 350k \rightarrow \text{Error} = (350k - 350k)^2 = 0$ 
```

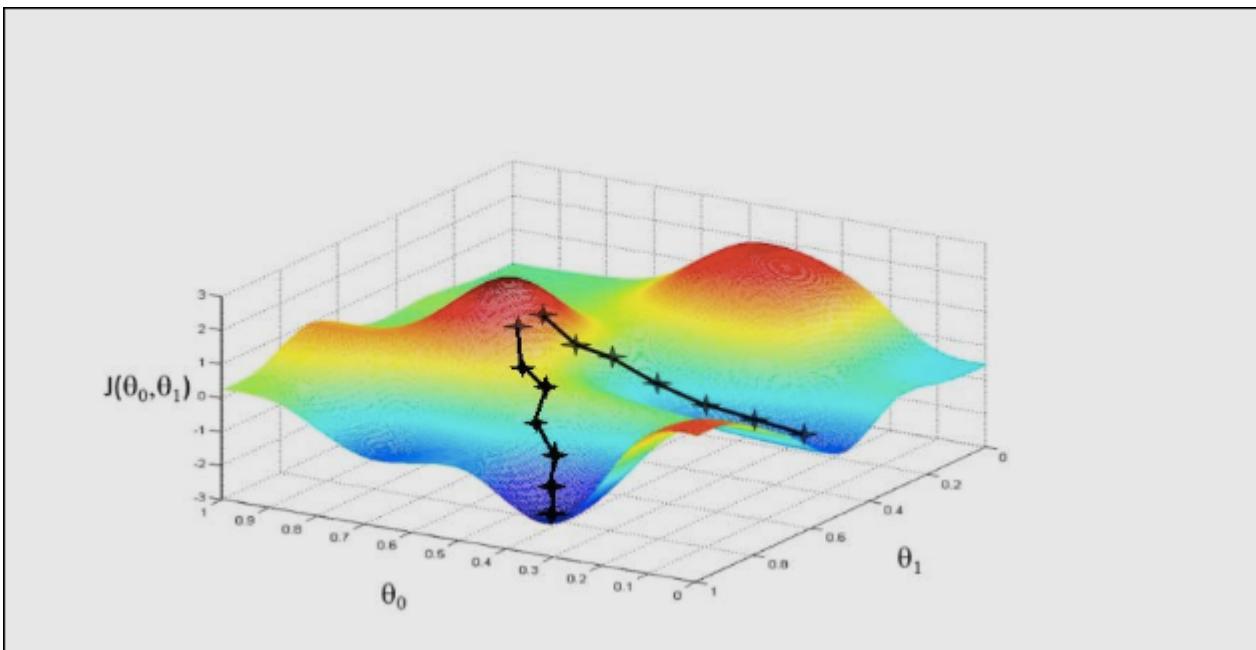
$J(\theta) = 0$  ✓ ¡Ajuste perfecto!

## 5.5.5 Descenso de Gradiente

**Objetivo:** Encontrar los valores de  $\theta$  que **minimizan  $J(\theta)$** .

**Concepto visual:** Imagina que estás en una montaña y quieres bajar al valle (mínimo).





**Algoritmo** (repetir hasta convergencia):

$$\theta_j := \theta_j - \alpha \times \frac{\partial J}{\partial \theta_j}$$

Donde:

- $\alpha$  (alpha): Tasa de aprendizaje (tamaño del paso)
- $\frac{\partial J}{\partial \theta_j}$ : Derivada parcial (dirección del descenso)

**Forma específica para regresión lineal:**

$$\theta_0 := \theta_0 - \alpha \times \frac{\partial}{\partial \theta_0} \times J(\theta_0, \theta_1)$$

$$\theta_1 := \theta_1 - \alpha \times \frac{\partial}{\partial \theta_1} \times J(\theta_0, \theta_1)$$

$$\frac{\partial}{\partial \theta_j} \times J(\theta_0, \theta_1) = J(\theta_0, \theta_1) = \frac{\partial}{\partial \theta_j} \times \frac{1}{2m} \times \sum (h_\theta(x^{(i)}) - y^{(i)})^2 = \frac{\partial}{\partial \theta_j} \times \frac{1}{2m} \times \sum (\theta_0 + \theta_1 x^{(i)} - y^{(i)})^2$$

$$j = 0 : \frac{\partial}{\partial \theta_0} \times J(\theta_0, \theta_1) = \frac{1}{m} \times \sum (h_\theta(x^{(i)}) - y^{(i)})$$

$$j = 1 : \frac{\partial}{\partial \theta_1} \times J(\theta_0, \theta_1) = \frac{1}{m} \times \sum (h_\theta(x^{(i)}) - y^{(i)}) \times x^{(i)}$$

## 5.5.6 Ejemplo Paso a Paso

**Problema:** Predecir precio de casas según tamaño.

**Datos de entrenamiento** ( $m=3$ ):

x (m <sup>2</sup> )	y (precio €)
50	150,000
100	250,000
150	350,000

**Paso 1: Inicializar parámetros**

$$\begin{aligned}\theta_0 &= 0 \\ \theta_1 &= 0 \\ \alpha &= 0.01 \text{ (tasa de aprendizaje)}\end{aligned}$$

**Paso 2: Primera predicción**

$$h_{\theta}(x) = \theta_0 + \theta_1 \cdot x = 0 \text{ (para todos los } x)$$

**Paso 3: Calcular coste inicial**

$$\begin{aligned}J(\theta) &= (1/6) \times [(0-150k)^2 + (0-250k)^2 + (0-350k)^2] \\ &= (1/6) \times [22,500M + 62,500M + 122,500M] \\ &= 34,583M \leftarrow \text{¡Muy alto! } \times\end{aligned}$$

**Paso 4: Calcular gradientes y actualizar θ**

$$\begin{aligned}\frac{\partial J}{\partial \theta_0} &= (1/3) \times [(0-150k) + (0-250k) + (0-350k)] \\ &= -250,000\end{aligned}$$

$$\begin{aligned}\frac{\partial J}{\partial \theta_1} &= (1/3) \times [(0-150k) \times 50 + (0-250k) \times 100 + (0-350k) \times 150] \\ &= -30,000,000\end{aligned}$$

$$\begin{aligned}\theta_0 &:= 0 - 0.01 \times (-250,000) = 2,500 \\ \theta_1 &:= 0 - 0.01 \times (-30,000,000) = 300,000\end{aligned}$$

**Paso 5: Nueva predicción**

$$h_{\theta}(x) = 2,500 + 300,000 \cdot x$$

**Repetir** pasos 3-4 hasta que  $J(\theta)$  deje de disminuir significativamente.

## 5.5.7 Tasa de Aprendizaje $\alpha$

**Concepto:** Controla el tamaño de los pasos al descender.

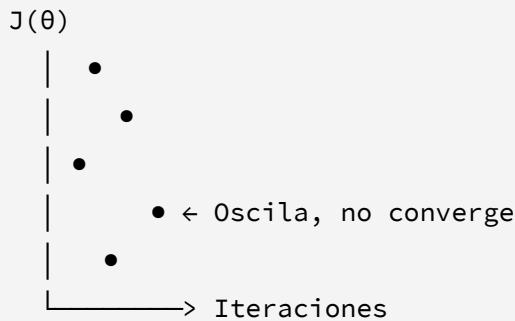
Valor de $\alpha$	Efecto	Problema
<b>Muy pequeño</b> (0.001)	Pasos muy pequeños	Convergencia MUY lenta 🐀
<b>Adecuado</b> (0.01-0.1)	Pasos balanceados	Convergencia óptima ✓
<b>Muy grande</b> (10)	Pasos muy grandes	Puede no converger (oscila) ✗

**Visualización del efecto de  $\alpha$ :**

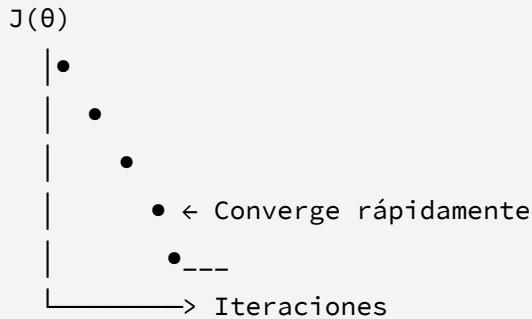
$\alpha$  muy pequeño:



$\alpha$  muy grande:



$\alpha$  adecuado:



## 5.6 Regresión Lineal Multivariable

Hasta ahora: 1 característica (tamaño de casa)

Ahora: Múltiples características (tamaño, habitaciones, edad, etc.)

## 5.6.1 Notación Extendida

### Nueva forma de $h$ :

$$h_{\theta}(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n$$

En forma vectorial:

$$h_{\theta}(x) = \theta^T \times x$$

Donde:

$$\theta = [\theta_0, \theta_1, \theta_2, \dots, \theta_n]^T$$

$$x = [1, x_1, x_2, \dots, x_n]^T \quad (x_0 = 1 \text{ por convención})$$

## 5.6.2 Ejemplo con 3 características

### Predicción de precio de casa:

Características:

$x_1$  = Tamaño ( $m^2$ )

$x_2$  = Número de habitaciones

$x_3$  = Edad (años)

$$h_{\theta}(x) = \theta_0 + \theta_1 \cdot \text{tamaño} + \theta_2 \cdot \text{habitaciones} + \theta_3 \cdot \text{edad}$$

Ejemplo con valores aprendidos:

$$h_{\theta}(x) = 80,000 + 2,000 \cdot \text{tamaño} + 10,000 \cdot \text{habitaciones} - 1,000 \cdot \text{edad}$$

Para una casa de  $100m^2$ , 3 habitaciones, 5 años:

$$\begin{aligned} h_{\theta}(x) &= 80,000 + 2,000 \cdot 100 + 10,000 \cdot 3 - 1,000 \cdot 5 \\ &= 80,000 + 200,000 + 30,000 - 5,000 \\ &= 305,000 \text{€} \end{aligned}$$

## 6.3 Normalización de Características

Si las características tienen rangos muy diferentes, el descenso de gradiente converge lentamente.

### Ejemplo:

$x_1$  = Tamaño: rango [50 - 200]  
 $x_2$  = Habitaciones: rango [1 - 5]

### Solución: Feature Scaling

$$x_j^*(i) := (x_j(i) - \mu_j) / s_j$$

Donde:

$\mu_j$  = media de la característica  $j$

$s_j$  = desviación estándar de la característica  $j$

### Ejemplo numérico:

Original  $x_1$  (tamaño): [50, 100, 150, 200]

$$\mu_1 = (50+100+150+200)/4 = 125$$

$$s_1 = \text{desviación} = 55.9$$

Normalizado:

$$x_1 = [50-125]/55.9 = -1.34$$

$$x_1 = [100-125]/55.9 = -0.45$$

$$x_1 = [150-125]/55.9 = 0.45$$

$$x_1 = [200-125]/55.9 = 1.34$$

Ahora todos están aproximadamente en el rango [-2, 2] ✓

## 5.6.4 Ecuación Normal (Método Analítico)

**Alternativa al descenso de gradiente:** Calcular  $\theta$  directamente con álgebra lineal.

### Fórmula:

$$\theta = (X^T X)^{-1} X^T y$$

Donde:

$X$  = matriz de características ( $m \times n+1$ )

$y$  = vector de salidas ( $m \times 1$ )

### Comparación con Descenso de Gradiente:

Descenso de Gradiente	Ecuación Normal
Necesita elegir $\alpha$	No necesita $\alpha$ ✓
Necesita muchas iteraciones	Sin iteraciones ✓
Funciona bien con $n$ grande (millones) ✓	Lento con $n > 10,000$ ✗
Necesita normalizar features	No necesita normalizar ✓
Complejidad $O(kn^2)$	Complejidad $O(n^3)$

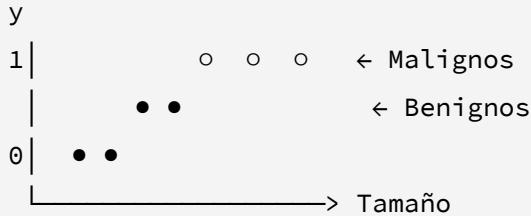
## 5.7 Regresión Logística

**Objetivo:** Resolver problemas de **clasificación** (no regresión, a pesar del nombre).

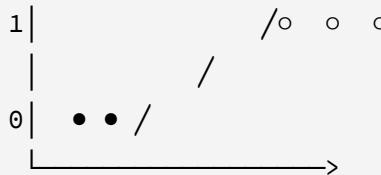
### 5.7.1 ¿Por qué no usar regresión lineal para clasificación?

**Problema visual:**

Clasificar tumor: 0 (benigno) o 1 (maligno)



Si usamos regresión lineal:



Problemas:

- $h(x)$  puede ser  $> 1$  o  $< 0$  ✗
- Un tumor MUY grande altera toda la línea ✗

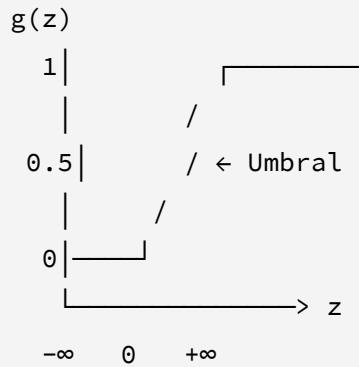
### 5.7.2 Función Sígmide (Logística)

**Solución:** Usar una función que siempre devuelva valores entre 0 y 1.

**Fórmula:**

$$g(z) = 1 / (1 + e^{-z})$$

$$h_{\theta}(x) = g(\theta^T x) = 1 / (1 + e^{-(\theta^T x)})$$

**Gráfica:**

$$z \gg 0 \rightarrow g(z) \approx 1$$

$$z = 0 \rightarrow g(z) = 0.5$$

$$z \ll 0 \rightarrow g(z) \approx 0$$

**Interpretación:**

$$h_{\theta}(x) = 0.7$$

Significa: "Hay un 70% de probabilidad de que  $y=1$ "

### 5.7.3 Frontera de Decisión

**Regla de decisión:**

Si  $h_{\theta}(x) \geq 0.5 \rightarrow$  predecir  $y=1$

Si  $h_{\theta}(x) < 0.5 \rightarrow$  predecir  $y=0$

Como  $g(z) \geq 0.5$  cuando  $z \geq 0$ :

Si  $\theta^T x \geq 0 \rightarrow$  predecir  $y=1$

Si  $\theta^T x < 0 \rightarrow$  predecir  $y=0$

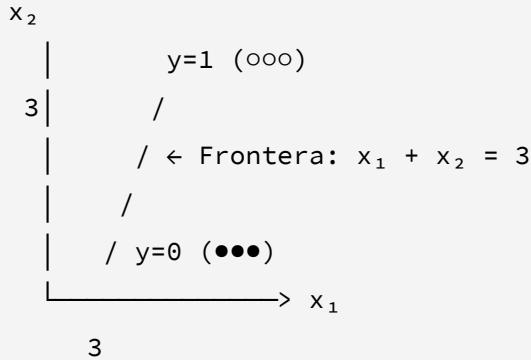
## Ejemplo 1: Frontera lineal

$$h_{\theta}(x) = g(-3 + x_1 + x_2)$$

Predecir  $y=1$  cuando:  $-3 + x_1 + x_2 \geq 0$

Es decir:  $x_1 + x_2 \geq 3$

Gráfica:

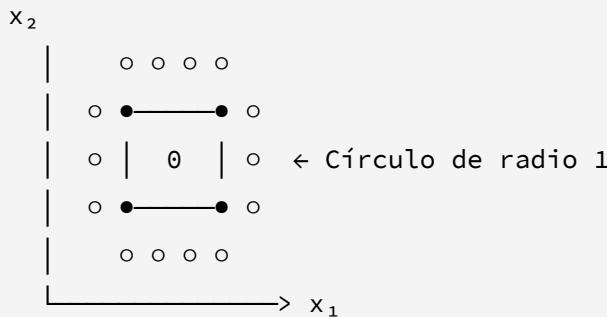


## Ejemplo 2: Frontera circular

$$h_{\theta}(x) = g(-1 + x_1^2 + x_2^2)$$

Predecir  $y=1$  cuando:  $x_1^2 + x_2^2 \geq 1$

Gráfica:



Interior (●):  $y=0$

Exterior (○):  $y=1$

## 5.7.4 Función de Coste para Regresión Logística

**Problema:** El error cuadrático hace que  $J(\theta)$  sea no convexa (múltiples mínimos).

**Solución:** Nueva función de coste:

$$\text{Cost}(h_{\theta}(x), y) = -\log(h_{\theta}(x)) \quad \text{si } y=1$$

$$\text{Cost}(h_{\theta}(x), y) = -\log(1 - h_{\theta}(x)) \quad \text{si } y=0$$

**Forma compacta:**

$$\text{Cost}(h_{\theta}(x), y) = -y \cdot \log(h_{\theta}(x)) - (1-y) \cdot \log(1-h_{\theta}(x))$$

**Función de coste total:**

$$J(\theta) = -(1/m) \sum [y^{(i)} \cdot \log(h_{\theta}(x^{(i)})) + (1-y^{(i)}) \cdot \log(1-h_{\theta}(x^{(i)}))]$$

## Intuición:

Si  $y=1$ :

- $h_{\theta}(x) = 1 \rightarrow \text{Cost} = -\log(1) = 0 \checkmark \text{ Sin penalización}$

- $h_{\theta}(x) = 0.5 \rightarrow \text{Cost} = -\log(0.5) = 0.69$

- $h_{\theta}(x) = 0.1 \rightarrow \text{Cost} = -\log(0.1) = 2.3 \times \text{Penalización alta}$

Si  $y=0$ :

- $h_{\theta}(x) = 0 \rightarrow \text{Cost} = -\log(1) = 0 \checkmark \text{ Sin penalización}$

- $h_{\theta}(x) = 0.5 \rightarrow \text{Cost} = -\log(0.5) = 0.69$

- $h_{\theta}(x) = 0.9 \rightarrow \text{Cost} = -\log(0.1) = 2.3 \times \text{Penalización alta}$

## 5.7.5 Algoritmo de Descenso de Gradiente

**Actualización** (¡idéntica en forma a regresión lineal!):

$$\theta_j := \theta_j - \alpha \times (1/m) \times \sum [h_{\theta}(x^{(i)}) - y^{(i)}] \times x_j^{(i)}$$

Pero recuerda:

$$h_{\theta}(x) = 1/(1 + e^{-\theta^T x}) \leftarrow \text{Diferente de regresión lineal}$$

## 7.6 Clasificación Multiclas

**Estrategia:** One-vs-All (uno contra todos)

**Proceso:**

Problema: Clasificar emails en 3 categorías

- Clase 1: Personal
- Clase 2: Trabajo
- Clase 3: Spam

Entrenamos 3 clasificadores:

Clasificador 1: ¿Es Personal? (sí vs no)

Clasificador 2: ¿Es Trabajo? (sí vs no)

Clasificador 3: ¿Es Spam? (sí vs no)

Para un nuevo email:

$h_1(x) = 0.2$  (20% probabilidad Personal)

$h_2(x) = 0.7$  (70% probabilidad Trabajo) ← ¡Máximo!

$h_3(x) = 0.1$  (10% probabilidad Spam)

Predicción final: Trabajo ✓

## 5.8 Regresión Polinómica

**Objetivo:** Ajustar curvas (no solo líneas rectas).

**Idea:** Crear nuevas características elevando las originales a potencias.

Original:

$$h_{\theta}(x) = \theta_0 + \theta_1 x$$

Cuadrática:

$$h_{\theta}(x) = \theta_0 + \theta_1 x + \theta_2 x^2$$

Cúbica:

$$h_{\theta}(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$$

**Ejemplo visual:**

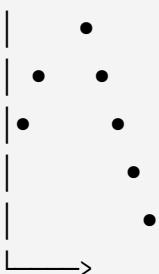
Precio de casa según tamaño

Lineal:



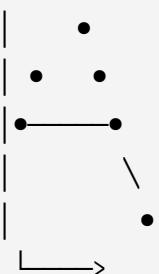
Simple

Cuadrática:



Mejor ajuste

Cúbica:



Puede sobreajustar

## 5.9 Overfitting (Sobreajuste)

### 5.9.1 Los Tres Escenarios

**Ejemplo con regresión:**

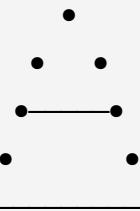
UNDERFITTING  
(Subajuste)



$$h(x) = \theta_0 + \theta_1 x$$

Sesgo alto  
No captura  
el patrón

GOOD FIT  
(Buen ajuste)



$$h(x) = \theta_0 + \theta_1 x + \theta_2 x^2$$

Balance   
Captura  
el patrón

OVERFITTING  
(Sobreajuste)



$$h(x) = \theta_0 + \dots + \theta_5 x^5$$

Varianza alta  
Captura el ruido  
también

**Ejemplo con clasificación:**

UNDERFITTING	GOOD FIT	OVERFITTING
Línea recta muy simple	Curva suave apropiada	Frontera errática se ajusta al ruido

## 5.9.2 Soluciones al Overfitting

### Solución 1: Reducir número de características

Manualmente:

- Eliminar características poco relevantes

Automáticamente:

- Algoritmos de selección de características

### Solución 2: Regularización

**Concepto:** Penalizar parámetros  $\theta$  muy grandes.

**Nueva función de coste:**

$$J(\theta) = [\text{coste original}] + \lambda/(2m) \times \sum \theta_j^2$$

└ Término de regularización

Donde:

$\lambda$  (lambda) = parámetro de regularización

**Efecto:**

- |                    |  |
|--------------------|--|
| $\lambda = 0$ :    | Sin regularización $\rightarrow$ Possible overfitting    |
| $\lambda$ pequeño: | Poca regularización $\rightarrow$ Balance                |
| $\lambda$ grande:  | Mucha regularización $\rightarrow$ Possible underfitting |

**Ejemplo:**

Si  $\lambda$  es muy grande, forzamos todos los  $\theta_j \approx 0$

Resultado:  $h(x) \approx \theta_0$  (función constante)  $\rightarrow$  Underfitting

**Regresión lineal regularizada:**

$$\theta_j := \theta_j - \alpha \times [(1/m) \sum (h_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)} + (\lambda/m) \theta_j]$$

└ término original └ └ regularización └

## 5.10 Flujo Completo de un Proyecto de ML

**1. DEFINIR EL PROBLEMA**

¿Regresión o Clasificación?

↓

**2. RECOPILAR DATOS**

Conjunto de entrenamiento ( $x$ ,  $y$ )

↓

**3. PREPROCESAR DATOS**

- Limpiar datos
- Normalizar características
- Dividir en entrenamiento/test

↓

**4. ELEGIR MODELO**

- Regresión lineal
- Regresión logística
- Otro algoritmo

↓

**5. ENTRENAR MODELO**

- Inicializar  $\theta$
- Minimizar  $J(\theta)$  con descenso de gradiente

↓

**6. EVALUAR MODELO**

- Probar en datos de test
- Verificar overfitting/underfitting

↓

**7. AJUSTAR Y MEJORAR**

- Cambiar  $\alpha$
- Añadir/quitar características
- Aplicar regularización

↓

**8. DESPLEGAR**

Usar el modelo en producción