Regresión Linear

https://goo.gl/VjSvKF

Aprendizaje Supervisado

Aprendizaje Supervisado

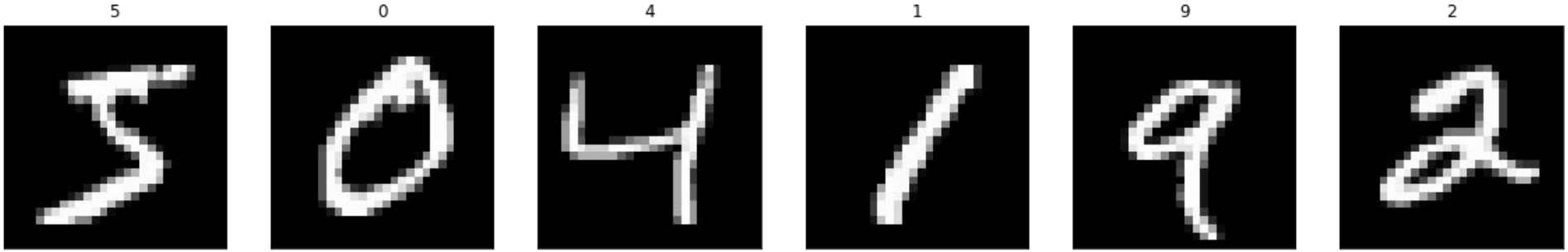
- Problema más común
- Damos datasets diciendo **x** (features/entradas) y **y** (lo que queremos predecir).
- Regresión
  - Predice un valor continuo
  - Estimar probabilidad que a partir del tamaño de un tumor sea maligno.
- Clasificación
  - Predice un valor discreto
  - Ejemplo: Clasificación binaria.
  - El tumor es maligno o no es maligno.

¿Qué preguntas responde el aprendizaje supervisado?

- ¿Cuánto dinero haremos invirtiendo X dinero en publicidad?
- ¿Este aplicante a un préstamo pagará de vuelta?
- ¿Cómo se comportará el mercado mañana?

Problemas de Aprendizaje Supervisado

- Tenemos un **dataset** con **training samples** que tienen su **labels** asociados.
- Ejemplo: reconocer dígitos escritos a mano



Modelo

$$Y = f(X) + \epsilon$$

- Un modelo es una abstracción del mundo real.
- Este modelo predice es salario anual a partir del número de estudios.

¿Qué es x?

Modelo

$$Y = f(X) + \epsilon$$

- Un modelo es una abstracción del mundo real.
- Este modelo predice es salario anual a partir del número de estudios.

¿Qué es x? El input. Años de estudios

¿Qué es Y?

Modelo

$$Y = f(X) + \epsilon$$

- Un modelo es una abstracción del mundo real.
- Este modelo predice es salario anual a partir del número de estudios.

¿Qué es x? Años de estudios

¿Qué es Y? El output - el salario anual.

¿Qué es f?

Modelo

$$Y = f(X) + \epsilon$$

- Un modelo es una abstracción del mundo real.
- Este modelo predice es salario anual a partir del número de estudios.

¿Qué es x? Años de estudios

¿Qué es Y? El output - el salario anual.

¿Qué es f? Una función que describe la relación entre X y Y.

¿Qué es ε?

Modelo

$$Y = f(X) + \epsilon$$

- Un modelo es una abstracción del mundo real.
- Este modelo predice es salario anual a partir del número de estudios.

¿Qué es x? Años de estudios

¿Qué es Y? El output - el salario anual.

¿Qué es f? Una función que describe la relación entre X y Y.

¿Qué es ε?

Modelo

$$Y = f(X) + \epsilon$$

- Un modelo es una abstracción del mundo real.
- Este modelo predice es salario anual a partir del número de estudios.

¿Qué es x? Años de estudios

¿Qué es Y? El output - el salario anual.

¿Qué es f? Una función que describe la relación entre X y Y.

¿Qué es Ɛ? Un error aleatorio con promedio de 0.

¿Cómo pueden predecir el salario a partir de años de educación

¿Cómo pueden predecir el salario a partir de años de educación

- Yo estimo que por cada año de educación, el promedio aumenta $5,000.

- Este modelo no es perfecto. Rara vez un modelo lo es.

  income = ($5,000 * years_of_education) + baseline_income

- En este método nosotros diseñamos la solución. Nosotros hicimos la ingeniería de la solución. Es diferente a una solución en la que se **aprende**.

¿Cómo pueden predecir el salario a partir de años de educación

- Si completó su carrera, agrega un multiplicador de 1.5x.

- Utilizar reglas explícitas no suele funcionar bien.

- ¿Por qué?

Reto

- ¿Cómo programarían un programa que puede detectar la cara de una persona con programación convencional a partir de los pixeles?

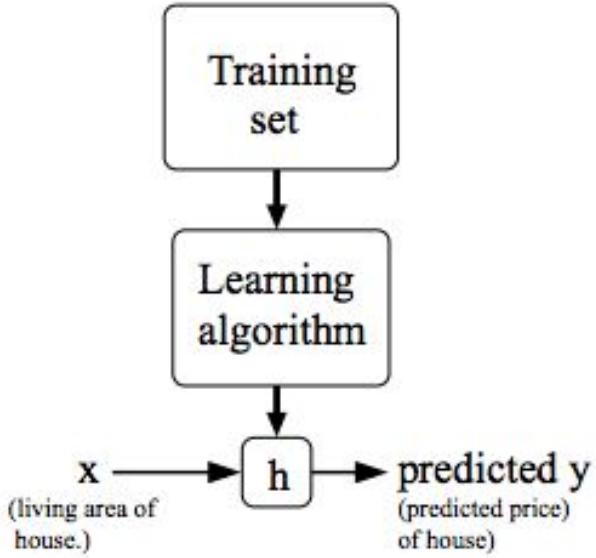
- ¿Qué pasa si hay luz diferente?

- ¿Qué pasa si la cabeza está inclinada con un ángulo?

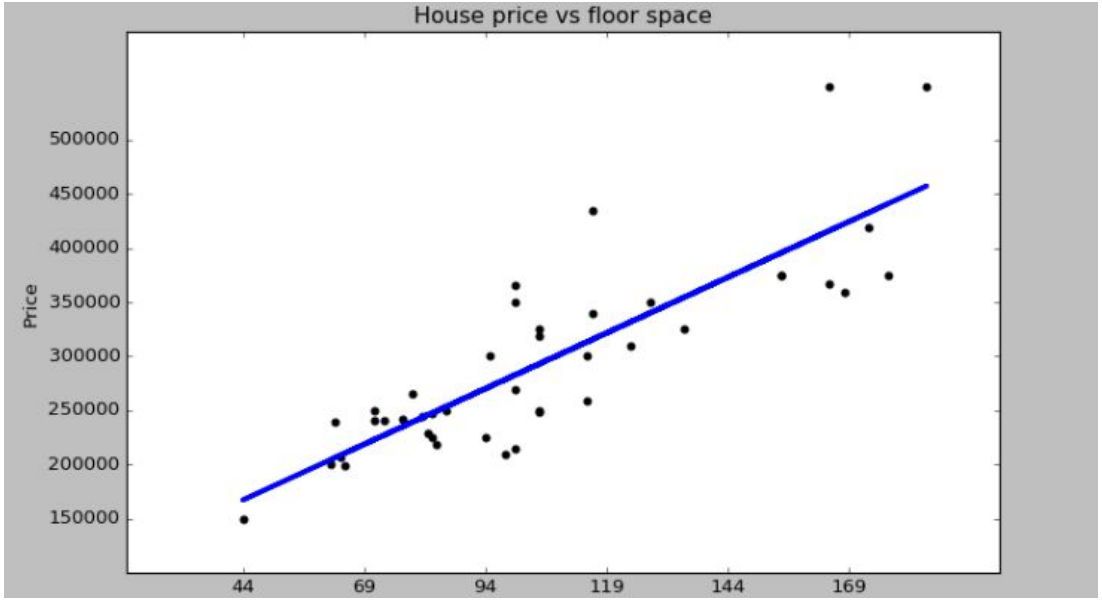
Aprendizaje Supervisado

- La computadora hará el trabajo por ti.
- El modelo genera heurísticas encontrando los patrones en la información.
- La computadora utiliza **labeled training data** a través de un **algoritmo de aprendizaje** para aprender una función. Una vez entrenado, podemos utilizar esta función para nuevos casos (**unlabeled testing data**).
- Objetivo: Predecir Y con la mayor precisión posible cuando se dan nuevos ejemplos de X donde Y no es conocida.

Supervised Learning

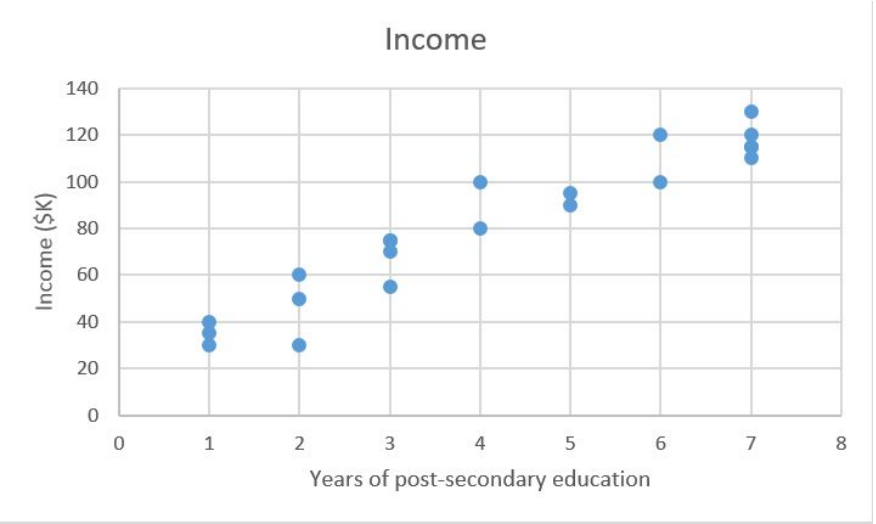


Regresión


House price vs floor space
Price
500000
450000
400000
350000
300000
250000
200000
150000
44
69
94
119
144
169

Regresión

- Predecir un valor real (continuo)
- Training set
  - m = # de training samples
  - x = variables de entrada o features
  - y = variable de salida
  - (x, y) un ejemplo de entrenamiento
  - $(x^{(i)}, y^{(i)})$ - i^{th} training sample



Features para predecir salario

- Feature
  - Atributos.
  - Las entradas (X).
  - Pueden ser numéricos o categóricos

- ¿Cuáles son nuestros features?

Features para predecir salario

- Feature
  - Atributos.
  - Las entradas (X).
  - Pueden ser numéricos o categóricos

- ¿Cuáles son nuestros features?
  - Años de educación.
  - Nombre de la universidad (con un id para identificarlo).
  - Carrera que estudió.

Retos

- Para que funcione, necesitamos un número alto de observaciones de entrenamiento.
- Entrenamiento (training set)
  - Tenemos los valores de X y los valores de Y.
- Pruebas (testing set)
  - Tenemos los valores de X y no tenemos los valores de Y.
  - Nos permite saber que nuestro modelo puede generalizar para nuevos casos

Supervised Learning: Regression

training set

Observation #	Years of Higher Education (X)	Income (Y)
1	4	$80,000
2	5	$91,500
3	0	$42,000
4	2	$55,000
…	…	…
N	6	$100,000

test set

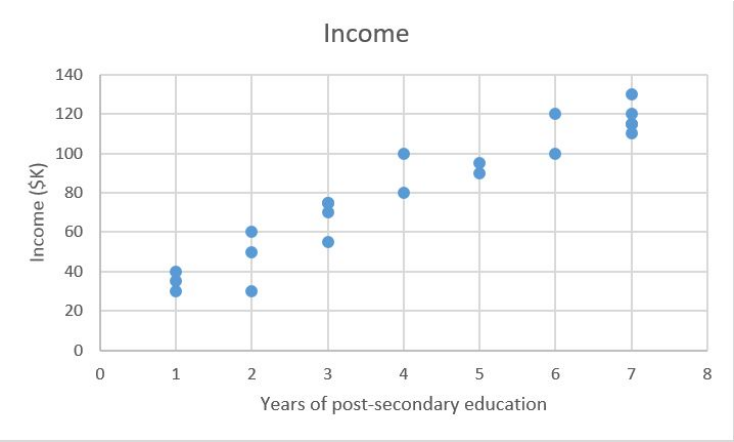
1	4	???
2	6	???

Machine Learning for Humans

Regresión Linear

Regresión Linear *"Draw the line. Yes, this counts as machine learning."*

- Regresión Linear con una variable



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

¿Cómo elegimos los parámetros?

- Proponemos diferentes hipótesis
- Vienen del training set
  - Propón valores qué sean un buen fit.
  - Elegirá θ_0 y θ_1 de manera que $h_\theta(x)$ se acerque a la y real en nuestros ejemplos de entrenamiento (x,y)
  - Es un problema de **optimización.**
  - Buscamos parámetros que minimicen el **error**.

Problema

X_train = [4, 5, 0, 2, ..., 6]

Y_train = [80, 91.5, 42, 55, ..., 100]

- Problema de optimización.
- Método paramétrico: Asume la función que relaciona X con Y.

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

- Objetivo: Aprender los parámetros del modelo que minimicen el error de las predicciones del modelo.

¿Cómo elegimos los parámetros

1. Definimos una función de pérdida (**Loss Function**) o de pérdida que va a medir qué tan inexacto es nuestro modelo.
2. Encontramos los parámetros que minimicen la pérdida (hagan nuestro modelo lo más exacto posible).

Dimensiones

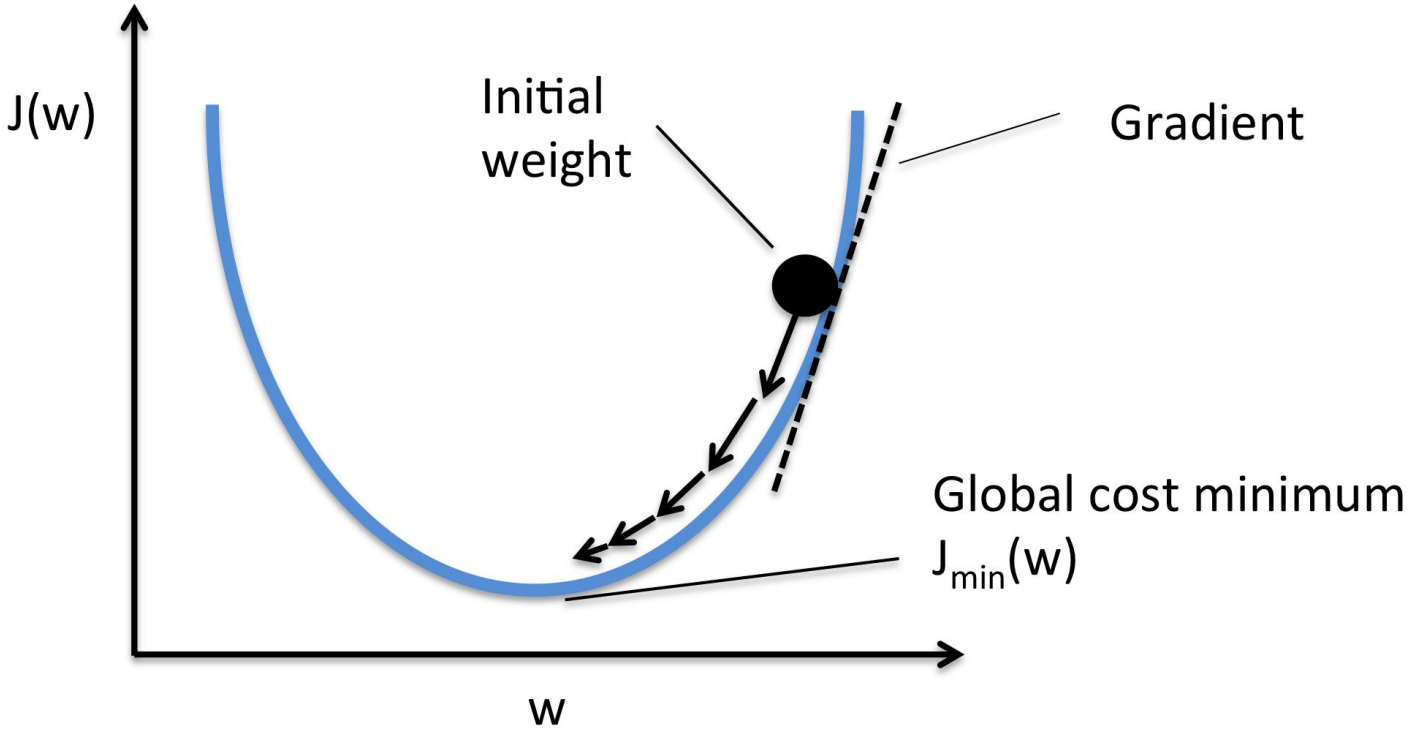
- Si tenemos dos dimensiones, usamos una línea.
- Si tenemos tres dimensiones, usamos un plano.
- Normalmente tenemos muchos features (muchas dimensiones) y coeficientes. Los mismos principios en dos dimensiones rigen para más.
- Por simplicidad trabajamos el caso que tenemos un feature (años de estudios).

Loss Function (1)

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(\hat{y}_i - y_i \right)^2 = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x_i) - y_i \right)^2$$

- Objetivo: Elegir θ_0 y θ_1 de manera que $h_\theta(x)$ se acerque a la *y* real en nuestros ejemplos de entrenamiento (x,y), es decir, que minimicen la función de pérdida J.
- En este caso usamos Mean Squared Error, pero hay otras.
- Lo podemos hacer con cálculo para un problema sencillo, pero las funciones de error aumentan en complejidad. Esto motiva un procedimiento iterativo que vaya minimizando el error.

Gradient Descent



Gradient Descent

- Tenemos una función $J(\theta_0$ y $\theta_1)$
- Algoritmo
  - Inicia con parámetros iniciales (se suelen iniciar en 0)
  - Se cambian los parámetros para reducir J hasta llegar a un mínimo.

Gradient Descent (2)

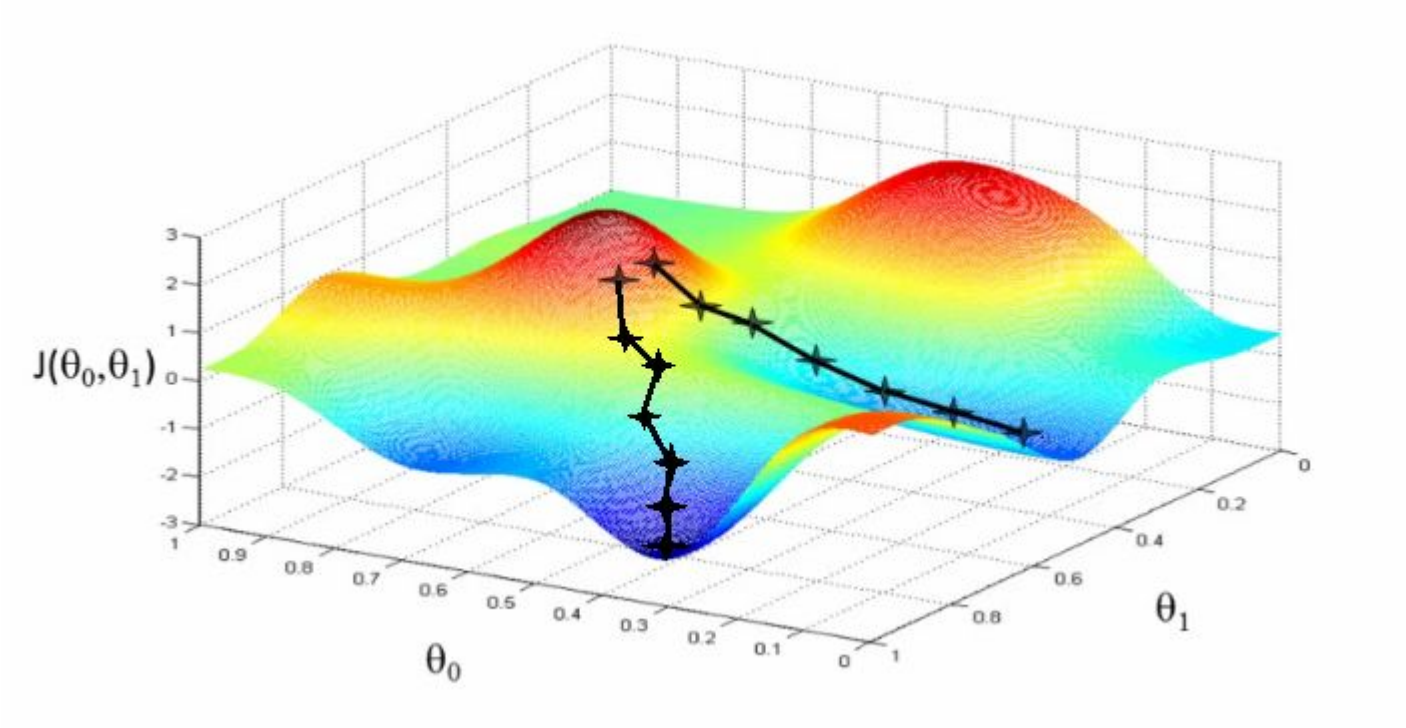
- Tenemos una función $J(\theta_0$ y $\theta_1)$
- Algoritmo
  - Inicia con parámetros iniciales (se suelen iniciar en 0)
  - Se cambian los parámetros para reducir J hasta llegar a un mínimo.
- Repetir hasta converger para cada parámetro:

$$\text{Repeat until convergence } \{$$

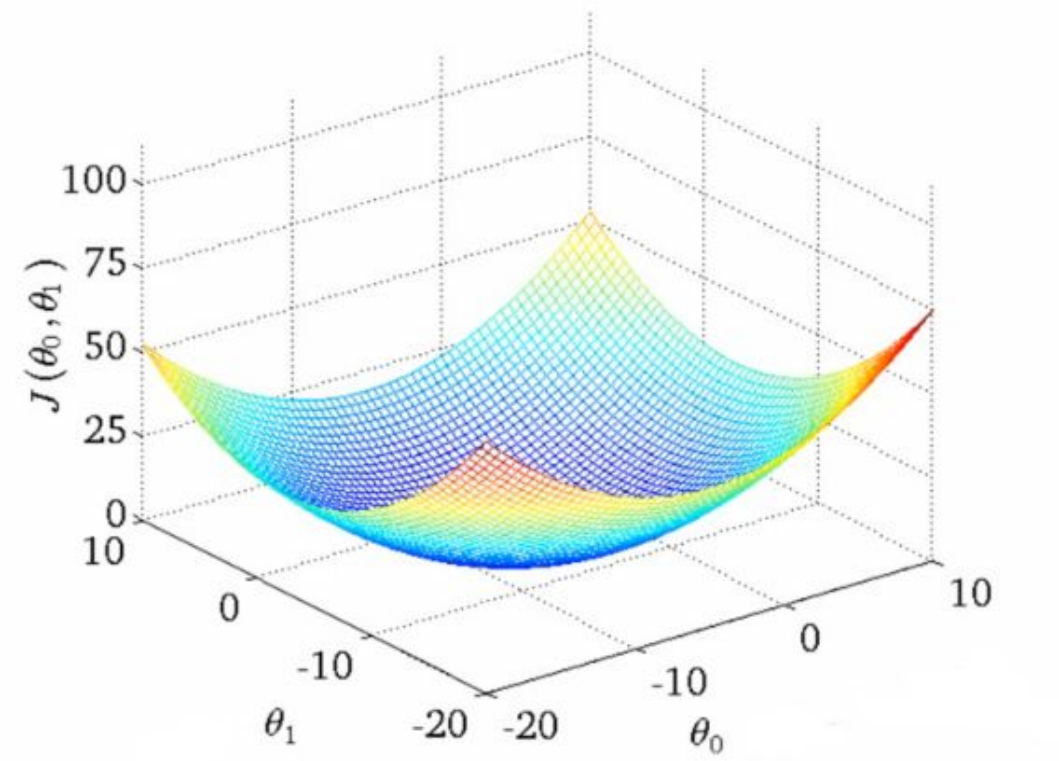
$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

$$\}$$

Gradient Descent



Loss Function - Linear Regression - Convex Function



¿Qué implica el algoritmo?

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(\hat{y}_i - y_i \right)^2 = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x_i) - y_i \right)^2$$

Repeat until convergence {

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

}

Derivadas parciales

- ¿Cuánto cambia la función de pérdida si hacemos un cambio **muy** pequeño en el parámetro?

$$\text{Repeat until convergence } \{$$

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

$$\}$$

Learning Rate

- Tenemos un **learning rate**.
- El learning rate es qué tan rápido o lento aprende nuestro modelo.
- ¿Qué pasa cuando el learning rate es muy grande?

$$\text{Repeat until convergence } \{$$

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

$$\}$$

Learning Rate

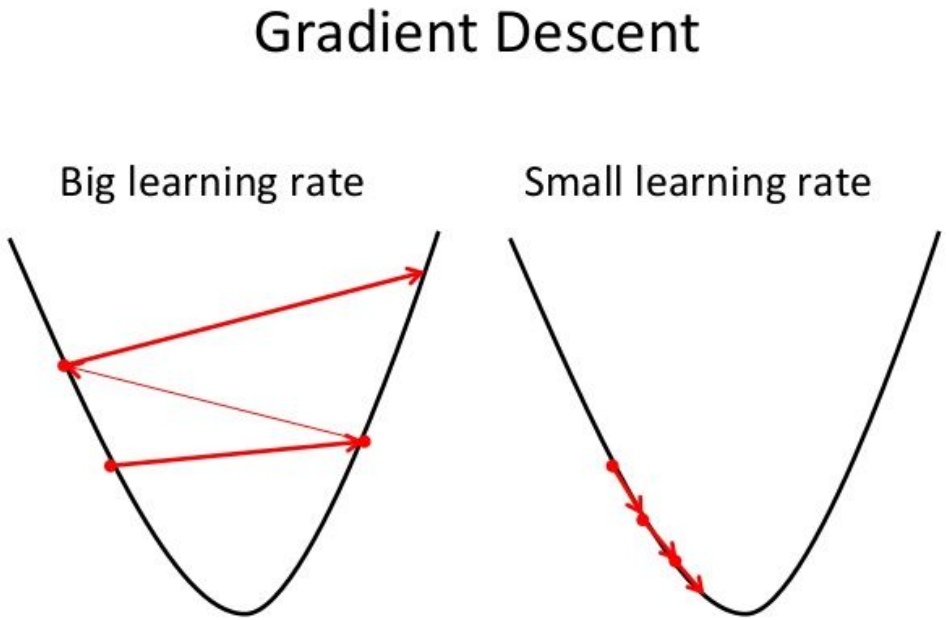
- Tenemos un **learning rate**.
- El learning rate es qué tan rápido o lento aprende nuestro modelo.
- ¿Qué pasa cuando el learning rate es muy chico?

$$\text{Repeat until convergence } \{$$

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

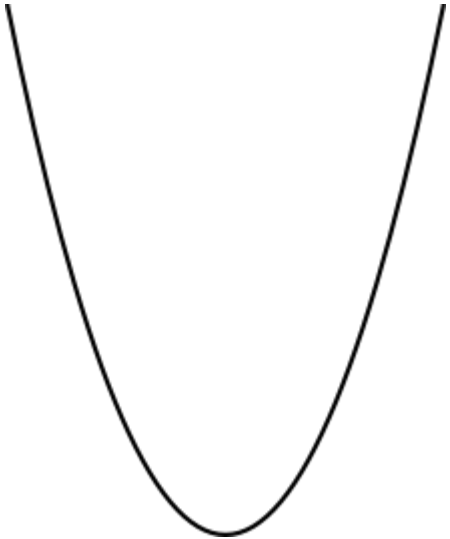
$$\}$$

Learning Rate

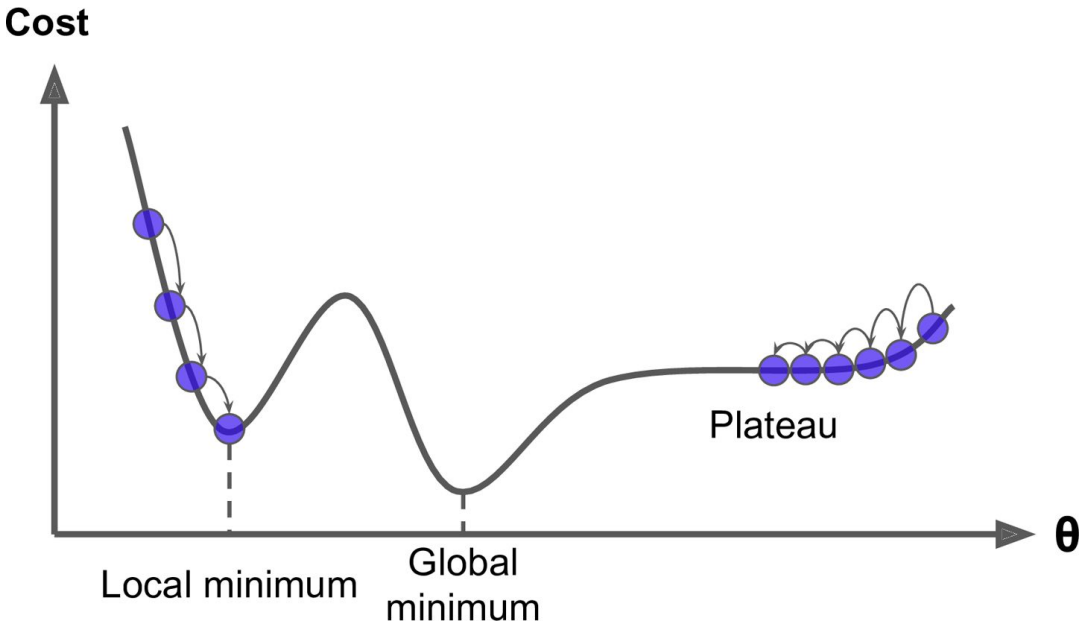


Sobre la derivada...

- ¿Es positiva o negativa?



Problemas con Gradient Descent



Las matemáticas de Gradient Descent (3 y 4)

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(\hat{y}_i - y_i \right)^2 = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x_i) - y_i \right)^2$$

$$\frac{\partial}{\partial \mathrm{m}} = \frac{2}{N} \sum_{i=1}^{N} - x_i (y_i - (mx_i + b))$$

$$\frac{\partial}{\partial \mathrm{b}} = \frac{2}{N} \sum_{i=1}^{N} - (y_i - (mx_i + b))$$

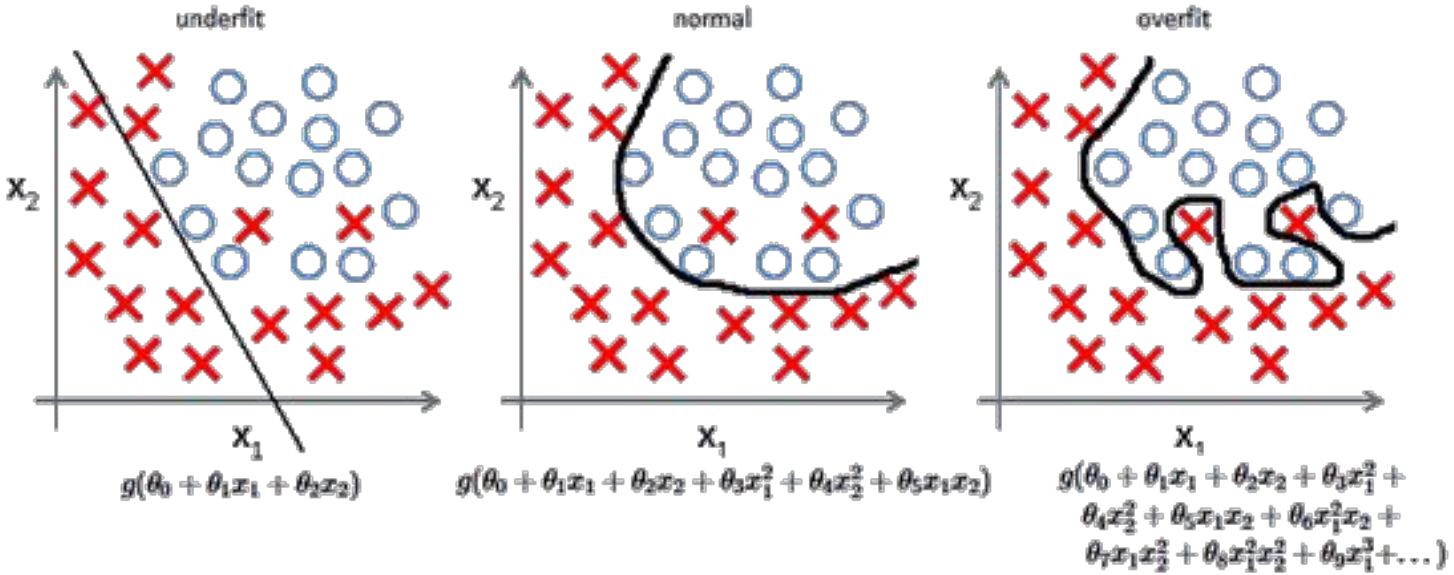
Overfitting

Conceptos

- **Overfitting:** Tu explicación es muy específica para este caso.
  - Su error con el training set es muy bajo.
  - Su error en el training set es alto.
  - Aprende la función perfectamente. Es como memorizar en vez de aprender.
  - No generaliza para nuevos casos.

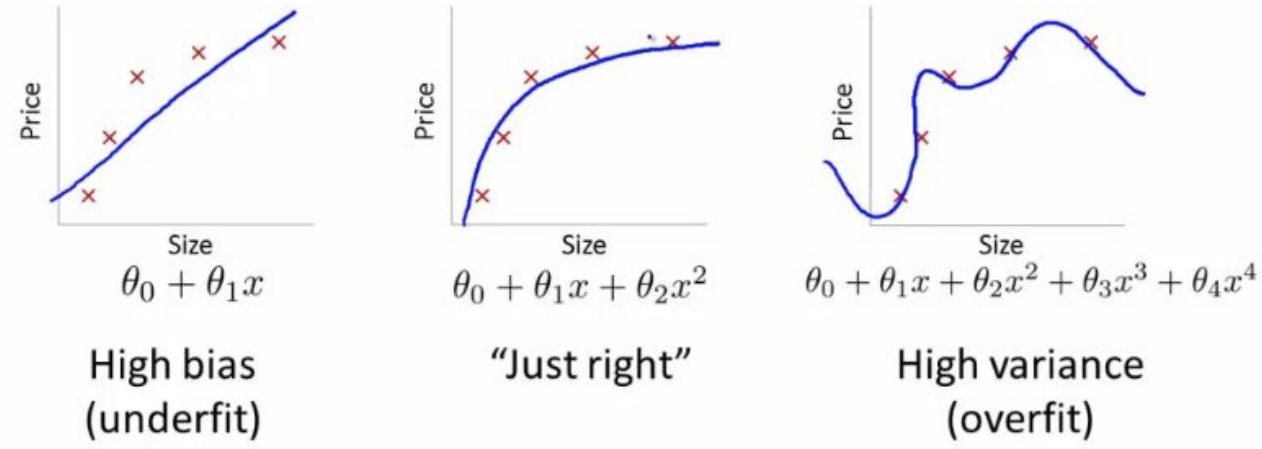
- **Regularization:** No compliques las cosas de más. Te castigaré por ir más allá de lo que debías.

Overfitting y Underfitting



Overfitting y Underfitting

- Bias: Error introducido por aproximar un fenómeno real con un modelo simple.
- Variance: Qué tanto cambia el error en el testing set cuando cambiamos la información de entrenamiento



Overfitting y Underfitting

- Si nuestro error en el training set es 0, normalmente es porque aprendió perfecto.
- Recuerden, el objetivo es generalizar para casos que nuestro modelo nunca ha visto.
- Para combatirlo
  - Usa más información de entrenamiento
  - Usa técnicas de **regularización** para combatirlo. Esto es una penalización en la función de pérdida por construir un modelo que da mucha importancia a un feature específico.

$$Cost = \frac{\sum_1^n ((\beta_1 x_i + \beta_0) - y_i)^2}{2 * n} + \lambda \sum_{i=0}^{1} \beta_i^2$$

Resumen

- Supervised Learning
- Regresión vs Clasificación
- Regresión Lineal - un algoritmo paramétrico
- Gradient Descent
- Overfitting y Regularización