UNIVERSIDAD
NACIONAL
DE COLOMBIA

Universidad Nacional de Colombia
Sede Medellín

Aprendizaje supervisado

Profesora: Patricia Jaramillo A. Ph.D

Métodos de optimización

- **De primer orden:** Consideran la aproximación de primer orden de la superficie de la función objetivo en el punto en el que se está evaluando.

- **De segundo orden:** Consideran la aproximación de segundo orden de la superficie de la función objetivo en el punto en el que se está evaluando.

El método de gradiente descendente, que vimos la clase pasada, es de primer orden.

Método de gradiente descendente

Es el método mas comun: de forma iterativa se actualiza el punto de búsqueda y la dirección de cambio hacia el nuevo punto depende del gradiente en ese punto.

Recuerde la regla de actualización :

$$\mathbf{x}(k+1) = \mathbf{x}(k) + \lambda(k)\Delta(k) \quad = \mathbf{x}(k) - \lambda(k)\nabla \mathrm{f}(\mathrm{x})$$

Donde:

- f(x) = función a optimizar
- X= variables a las que se les desea encontrar el valor para optimizar f(x)
- k= iteración
- λ(k) =tamaño de paso
- ∇f(x) gradiente de la función f(x) en el punto actual

Aprendizaje supervisado de un modelo de ajuste

- **Entrenar** (aprender, calibrar, ajustar): Encontrar los valores de los pesos que **minimicen** la función de costo E en un modelo de ajuste, que depende de diferencia entre los valores de salida de los datos observados y los que reproduce el modelo.

- Métodos mas comunes
  - Gradient descent (GD)
  - Stochastic gradient descent (SGD)
  - Newton
  - Momentum
  - Rprop y RMSprop
  - Adagrad
  - Adam

Aprendizaje supervisado con GD

- Las variables ha encontrar son los parámetros θ que hacen que la calidad del modelo f(θ) sea óptima, asi que el método del gradiente se interpreta ahora como:

$$\theta(k+1) = \theta(k) + \lambda(k)\Delta(k) \quad = \theta\,(k) - \lambda(k)\nabla \mathrm{E}(\theta)$$

Donde: θ = parámetros del modelo, por ejemplo pesos w (son los valores que se van ha encontrar)

E= función de error

$$\mathrm{E}(\theta) = \frac{1}{2}\sum_{i=1}^{n}\left(f_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

Por la alta dimensionalidad de la función objetivo (depende de n) vamos a usar una versión simplificada de GD: el SGD.

Gradiente Descente Estocástico (SGD)

- El tamaño de la muestra de datos puede ser muy grande (muchos datos) y calcular cada uno de los gradientes puede llevar demasiado tiempo.

- Como el gradiente es un promedio, se puede aproximar tomando una muestra mas pequeña.

- SGD usa, en cada iteración, no todo el conjunto de datos de entrenamiento, sino un conjunto de n datos, elegido aleatoriamente. Preferiblemente n=1

- Esto hace que el costo computacional sea menor y se evite usar información redundante.

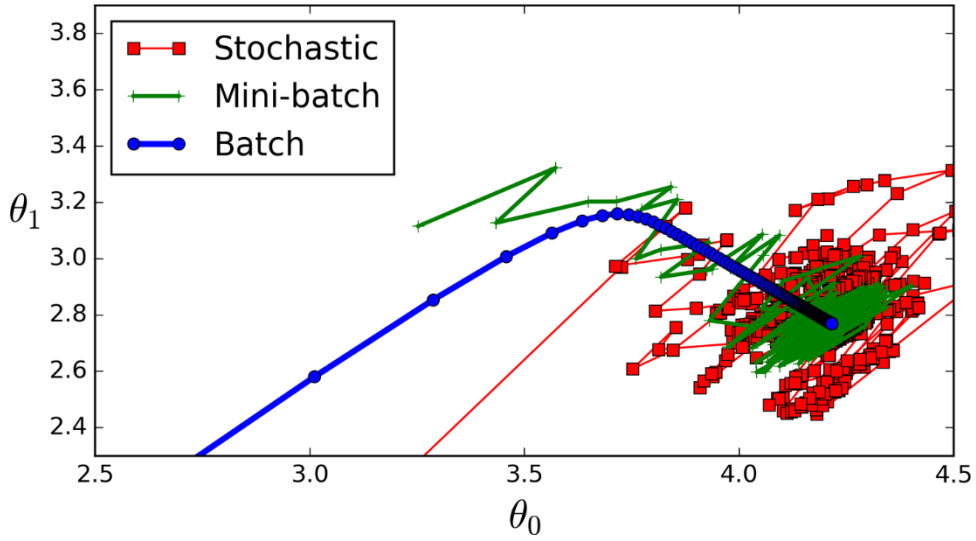
- SGD suele ser mas rápido que si se usa en cada iteración toda la muestra de entrenamiento (denominado Batch gradient descent BGD).

- Sin embargo, SGD no es capaz de llegar al mínimo sino a un punto muy cercano al óptimo y su búsqueda suele ser errática (u oscilatoria).

- Es decir, sacrifica exactitud pero se obtiene una solución relativamente buena rapidamente

Por ahora, trabajaremos con GD y SGD. Más adelante veremos los otros métodos.

Diferentes tipos de GD



- **Batch — GD**: es el método genérico en el que en cada iteración se incluye en la función de costo, los errores de todos los ejemplos de entrenamiento.

- **Mini — Batch — GD**: en cada iteracion no se usan todos los ejemplos de entrenamiento, puede usarse diferentes n ejemplos debido a que las bases de datos pueden ser muy grandes.

- **Estocástico — GD**: En cada iteracion se usa solo una muestra (ejemplo un solo caso de los ejemplos de entrenamiento) elegida aleatoriamente. Con este método el recorrido de búsqueda del mínimo suele ser errático .

Ejemplo: Entrenamiento en Regresión lineal

Llamando θ = b_0, w_j a los parámetros a calibrar

Y= $f(x, \theta)$

$$f_\theta(x) = b_0 + w_1 x_1 + w_2 x_2 \ldots$$

De forma vectorial (sin considerer θ_0)

$$f(x) = \sum_i w_i x_i = \mathbf{w}^T \mathbf{x}$$

Función de costo (error)

$$E(\theta) = \frac{1}{2} \sum_{i=1}^{n} \left(f_\theta\left(x^{(i)}\right) - y^{(i)} \right)^2$$

(aunque se puede usar otras funciones de costo)

Entrenar = Elegir los valores θ = b_0, w_j que minimicen $E(\theta)$

Algoritmo: Comienza con unos valores iniciales de prueba θ y se hacen repetidas actualizaciones para ir haciendo $E(\theta)$ cada vez menor hasta que se logre convergencia.

Recuerde la regla de actualización en cada iteración (por facilidad, supongamos que se usa SGD)

$$\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} E(\theta) = \theta_j - \alpha \frac{\partial}{\partial \theta_j} \left\{ \frac{(f_\theta(x) - y)^2}{2} \right\}$$

$$\frac{\partial}{\partial \theta_j} E(\theta) = \frac{\partial}{\partial \theta_j} \frac{1}{2} (f_\theta(x) - y)^2$$

$$= 2 \cdot \frac{1}{2} (f_\theta(x) - y) \cdot \frac{\partial}{\partial \theta_j} (f_\theta(x) - y)$$

$$= (f_\theta(x) - y) \cdot \frac{\partial}{\partial \theta_j} \left(\sum_{i=0}^{d} \theta_i x_i - y \right)$$

$$= (f_\theta(x) - y)\, x_j$$

Entonces:

$$\theta_j := \theta_j + \alpha \left(y^{(i)} - f_\theta\left(x^{(i)}\right) \right) x_j^{(i)}$$

Si se consideraran los n casos (x_i,y_i) de entrenamiento (método BGD):

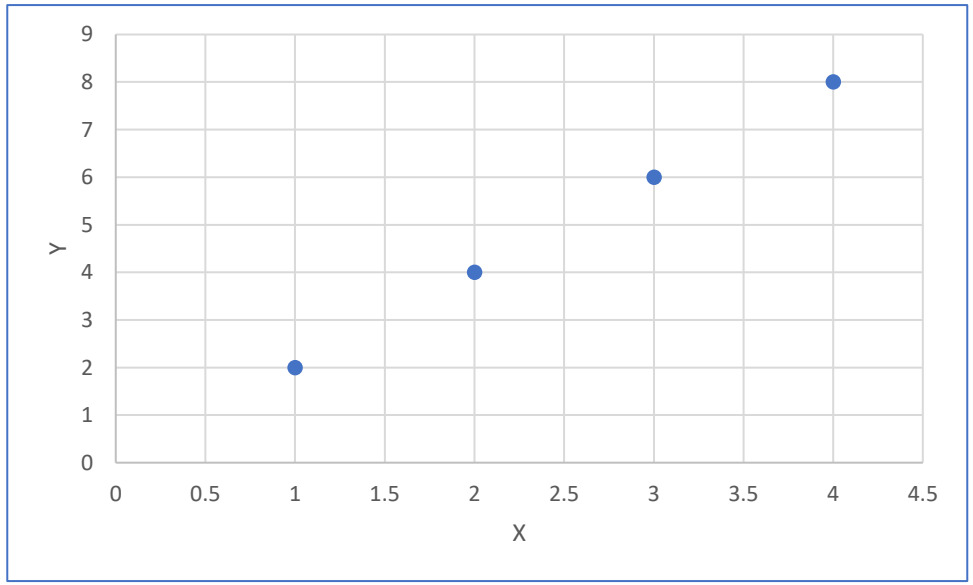
$$\theta_j := \theta_j + \alpha \sum_{i=1}^{n} \left(y^{(i)} - f_\theta\left(x^{(i)}\right) \right) x_j^{(i)}, \qquad (para\ cada\ variable\ xj)$$

En Regresión lineal este método converge a un óptimo global (no tiene óptimos locales)

Ejemplo sencillo: Entrenamiento de Regresión lineal mediante SGD

Conjunto de entrenamiento

X	Y
1	2
2	4
3	6
4	8



$Y = f(X) = w X$

$E = (w X - y)^2$

$\partial E/\partial w = 2X (wX - y)$

Regla de actualización: $w = w - \alpha \, \partial E/\partial w$

Suponga w inicial=0.5; α=0.1

Utilizaremos un punto por iteración, elegidos aleatoriamente:

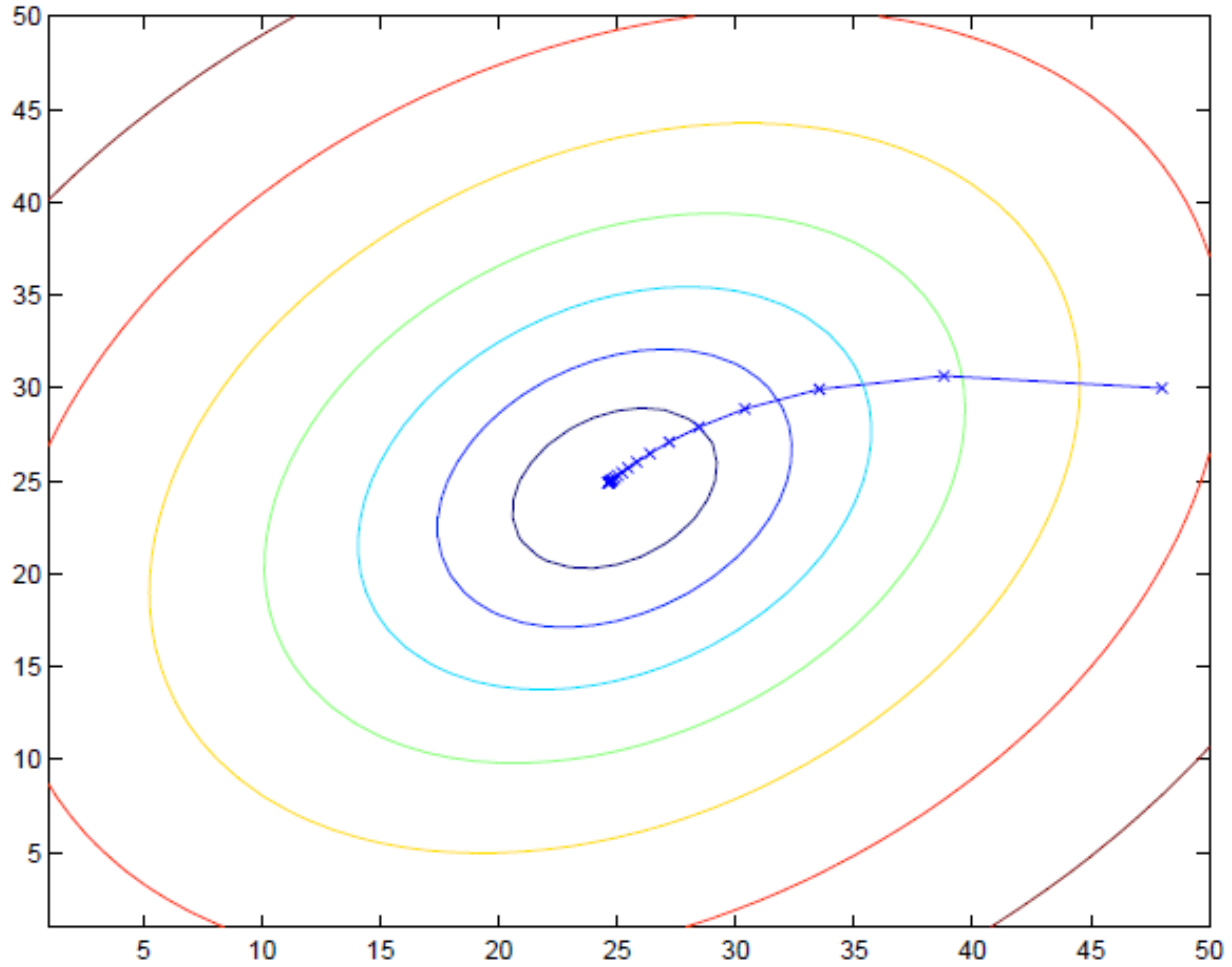
(2,4) w=0.5-2*0.1*2(2*0.5 -4)=1.7

(1,2) w=1.7-2*0.1*1(1*1.7 - 2)=1.76

(3,6) w=1.76 – 2 *0.1(3*1.76-6)=2.192

(4,8) ?

Gráficamente: E es función cuadrática convexa



Para regresión logística se usa la Derivada de la función sigmoidea

$$f(x) = \frac{1}{1+e^{-x}} = (1+e^{-x})^{-1} \qquad \rightarrow$$

$$\frac{\partial f(x)}{\partial x} = -(1+e^{-x})^{-2}(-e^{-x}) = -\frac{1}{(1+e^{-x})^2}(-e^{-x})$$

$$= \left(\frac{1}{1+e^{-x}}\right)\left(\frac{1+e^{-x}-1}{1+e^{-x}}\right)$$

$$\boxed{\frac{\partial f(x)}{\partial x} = \mathrm{f(x)(1-f(x))}}$$

Lo anterior hace que sea fácilmente derivable.