UNIVERSIDAD
NACIONAL
DE COLOMBIA

Universidad Nacional de Colombia
Sede Medellín

Métodos basados en gradiente - parte II
Extensiones del método básico
Métodos de segundo orden

Profesora: Patricia Jaramillo A. Ph.D

- El método del gradiente y si variantes mini-Batch o estocástico, son eficientes, pero aun se sigue investigando como mejor su eficiencia y exactitud.

- Se han hecho algunas propuestas que amplian algunos de los pasos del algoritmo

Normalización

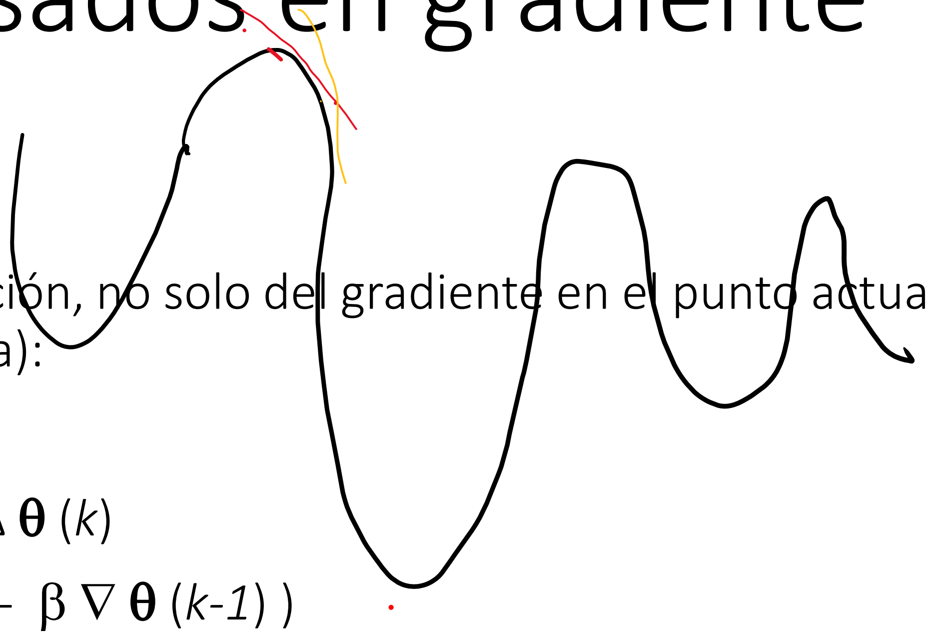
- Las variables de entrada pueden estar en unidades y escalas muy diferentes. Se sugiere normalizar (llevar a la misma escala y adimensionalidad).

- Esto disminuye problemas de propagación de grandes errores entre neuronas, permite un aprendizaje mas adecuado y un mayor entendimiento del resultado final.

- Ejemplos de normalización : (Valor(i)-Valor min)/(Valor maximo-Valor min)
- (Valor-E(Valores))/σ(valores)

Métodos adaptativos basados en gradiente

Momentum



- Cuando el método hace un movimiento utiliza información, no solo del gradiente en el punto actual, sino también del gradiente en el punto previo (memoria):

$$\boldsymbol{\theta}(k+1) = \boldsymbol{\theta}(k) + \Delta\boldsymbol{\theta}(k)$$

$$\Delta\boldsymbol{\theta}(k+1) = -\alpha\,(\nabla\boldsymbol{\theta}(k) - \beta\nabla\boldsymbol{\theta}(k-1))$$

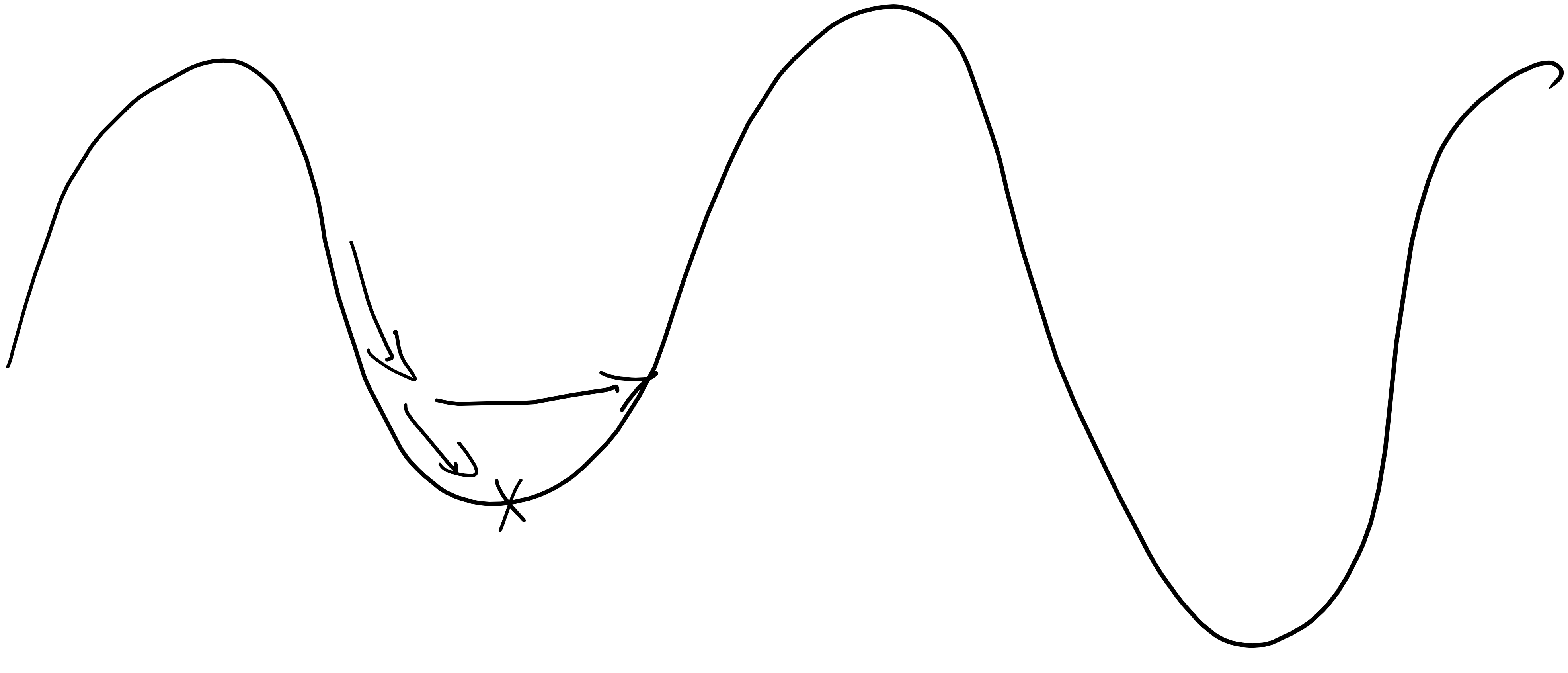
Donde $0 \leq \beta \leq 1$ momentum

Algunas versiones involucran más de un periodo de memoria (supermemoria).

$$\Delta\boldsymbol{\theta}(k+1) = -\alpha\,(\nabla\boldsymbol{\theta}(k) - \beta\nabla\boldsymbol{\theta}(k-1)) - \beta^2\nabla\boldsymbol{\theta}(k-2)\ \ldots)$$

- Esto hace que la trayectoria siga un decaimiento exponencial suave, a diferencia del movimiento caótico que se produce al usar el método de mini-lotes (mini-batches) o SGD.

EL momentum permite, cuando se llega a un gradiente cercano a 0, saltar en búsqueda de otros mínimos y así no quedar atrapado en óptimos locales.



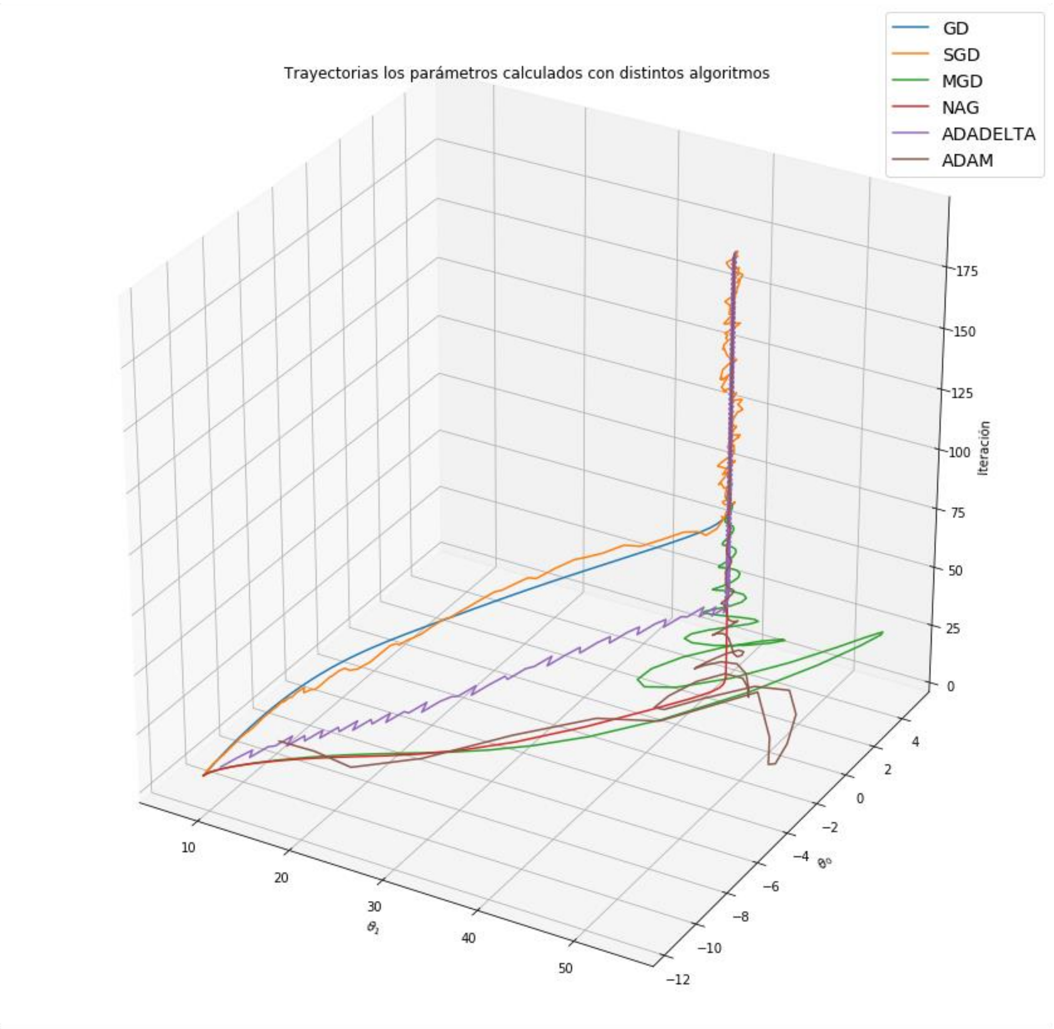
Adagrab

Puede usar diferentes α para cada parámetro. Lo actualiza así:

Si los últimos gradientes han sido altos, decrece α más rápido y así disminuye el Δ
Si los últimos gradientes han sido bajos , hace lo contrario:

La regla de actualización es:

$$\Delta \theta (k+1) = - \alpha \ (\nabla \theta (k) / \Sigma_t \nabla \theta (t))$$



De: http://personal.cimat.mx:8181/~mrivera/cursos/optimizacion/descenso_grad_estocastico/descenso_grad_estocastico.html#descenso-de-gradiente-adaptable-adagrad

Métodos Que utilizan el criterio de la segunda derivada

Se denominan Métodos de segundo orden

- La dirección de búsqueda depende también del Hessiano.
- Su ventaja es que es independente de cambios lineales en las coordenadas.
- Su desventaja: excesivo costo computacional en problemas de alta dimensionalidad, al invertir la matriz hessiana.

- Cualquier función suficientemente suave puede ser aproximada como una serie de Taylor de 2do orden:

$$f(\mathbf{x}+\boldsymbol{\delta}) = f(\mathbf{x}) + \mathbf{g}(\mathbf{x})^T\boldsymbol{\delta} + \frac{1}{2}\boldsymbol{\delta}^T\mathbf{H}_f(\mathbf{x})\boldsymbol{\delta} + H.O.T.$$

Donde H.O.T son los términos de orden superior (no se tendran en cuenta)

H es la matriz hessiana .

$$\mathbf{H}_f = \begin{bmatrix} \frac{\partial^2}{\partial x_1^2} f(\mathbf{x}) & \cdots & \frac{\partial^2}{\partial x_1 \partial x_n} f(\mathbf{x}) \\ \vdots & \vdots & \vdots \\ \frac{\partial^2}{\partial x_n \partial x_1} f(\mathbf{x}) & \cdots & \frac{\partial^2}{\partial x_n^2} f(\mathbf{x}) \end{bmatrix}$$

Método de Newton

Derivando:

$$\frac{d}{d\boldsymbol{\delta}} f(\mathbf{x}+\boldsymbol{\delta}) = \frac{d}{d\boldsymbol{\delta}}\left[f(\mathbf{x}) + \mathbf{g}(\mathbf{x})^T\boldsymbol{\delta} + \frac{1}{2}\boldsymbol{\delta}^T\mathbf{H}(\mathbf{x})\boldsymbol{\delta}\right] = \mathbf{g}(\mathbf{x}) + \mathbf{H}(\mathbf{x})\boldsymbol{\delta}$$

Igualando a cero:

$$\boldsymbol{\delta} = -\mathbf{H}(\mathbf{x})^{-1}\mathbf{g}(\mathbf{x})$$

Teniendo en cuenta que en el proceso iterativo $\mathbf{x} \rightarrow \mathbf{x}(k)$ y $\mathbf{x} + \boldsymbol{\delta} \rightarrow \mathbf{x}(k+1)$

- $\mathbf{x}(k+1) - \mathbf{x}(k) = -\mathbf{H}\big(\mathbf{x}(k)\big)^{-1}\mathbf{g}(\mathbf{x}(k)) = = -\mathbf{g}(\mathbf{x}(k)) \,/\, \mathbf{H}\big(\mathbf{x}(k)\big)$

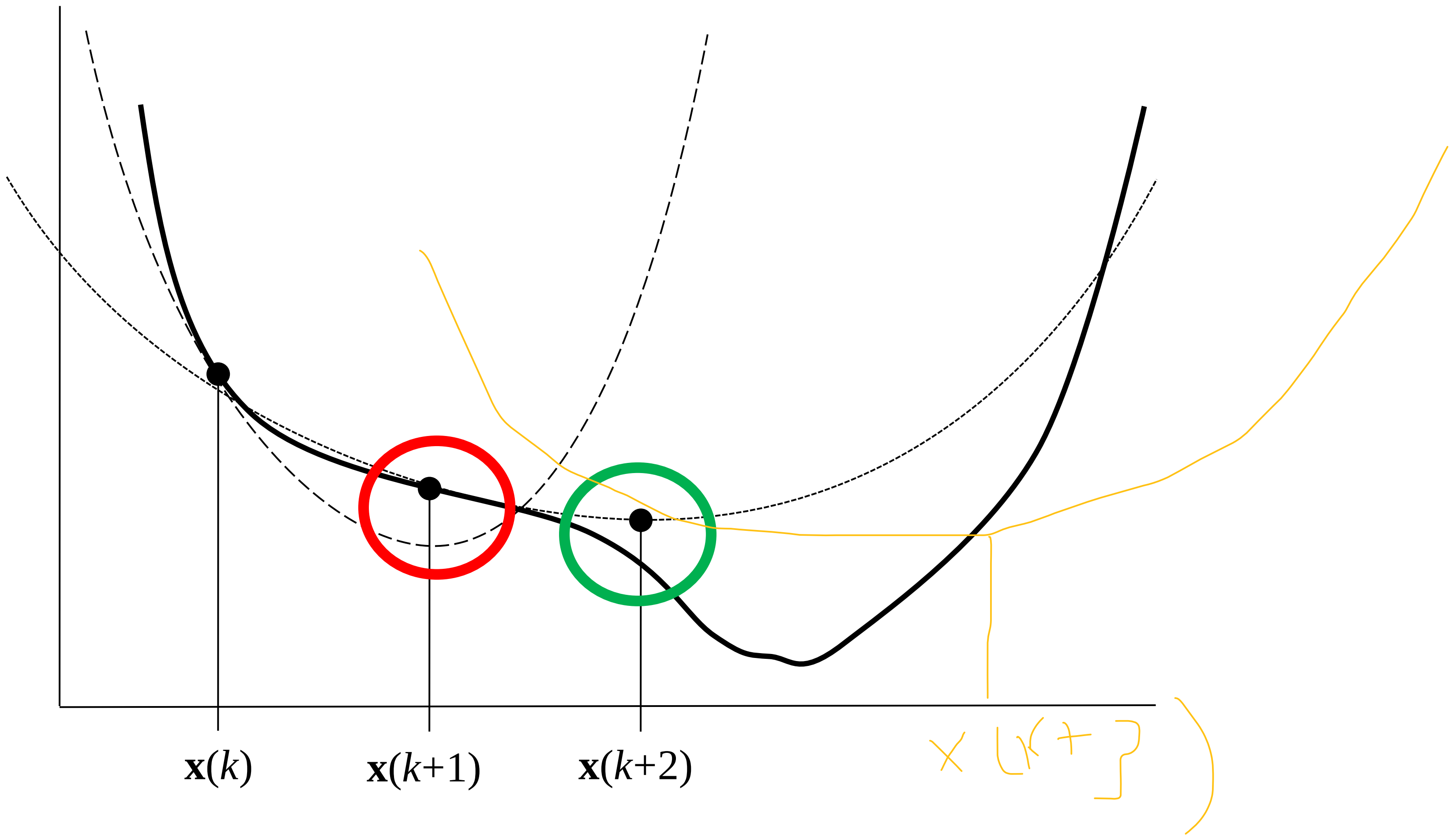
Se puede generalizar introduciendo un tamaño de paso:

$$\boldsymbol{x}(k+1) = \mathbf{x}(k) - \alpha(k) \times \mathbf{g}(k)/\mathbf{H}(k)$$

$\alpha(k)$ debe asegurar $f(\mathbf{x}(k+1)) \leq f(\mathbf{x}(k))$.

Método de Newton

Se aproxima a una parábola y el siguiente punto es el óptimo de la parábola.



Ejemplo sencillo

- Minimizar: f(X) = $(x_1 - 2)^4 + (x_1 - 2x_2)^2$

$$\frac{\partial f}{\partial x_1} = 4(x_1 - 2)^3 + 2(x_1 - 2x_2)$$

$$\frac{\partial f}{\partial x_2} = -4(x_1 - 2x_2)$$

$$\frac{\partial^2}{\partial x_1^2} f(\mathbf{x}) = 12(x_1 - 2)^2 + 2 \qquad \frac{\partial^2}{\partial x_1\, x_2} f(\mathbf{x}) = -4$$

$$\frac{\partial^2}{\partial x_2\, x_1} f(\mathbf{x}) = -4 \qquad \frac{\partial^2}{\partial x_2^2} f(\mathbf{x}) = 8$$

Comencemos en algún punto inicial, elegido aleatoriamente $\mathbf{x}(1) = \begin{bmatrix} 0.00 \\ 3.00 \end{bmatrix}$, $f\big(x(1)\big) = 52.00$

1. $\nabla f\big(\mathbf{x}(1)\big) = \begin{bmatrix} -44.0 \\ 24.0 \end{bmatrix}$

2. $\mathbf{H}\big(\mathbf{x}(1)\big) = \begin{bmatrix} 50.0 & -4.0 \\ -4.0 & 8.0 \end{bmatrix}$, suponiedo α=1

3. $\mathbf{x}(2) = \begin{bmatrix} 0.00 \\ 3.00 \end{bmatrix} - \begin{bmatrix} -44.0 \\ 24.0 \end{bmatrix} \div \begin{bmatrix} 50.0 & -4.0 \\ -4.0 & 8.0 \end{bmatrix} = \begin{bmatrix} 0.67 \\ 0.33 \end{bmatrix}$

4. $f\big(x(1)\big) = 3.16$

Disminuyó!!!! Seguir iterando hasta que gradiente cercano a 0

En conclusión

- Aplicar métodos de segundo orden en funciones muy complejas y altamente no lineales y de alta dimensionalidad es difícil por el cálculo de hessiano, pero son mucho más eficientes en numero de iteraciones que los de primer orden