Métodos de Optimización No Lineal Sin Restricciones: Un resumen

19 de abril de 2022

El Problema: Optimización NO Lineal sin restricciones

Aplicaciones de todo tipo

- Minimizar el costo de producción de una empresa.
- Maximizar las ganancias.
- Hallar las medidas del rectángulo que mejor ajusta a la patente en fotos de autos.
- Hallar los parámetros óptimos de una función de distribución para una muestra de datos.
- Redes Neuronales

El Problema: Optimización NO Lineal sin restricciones

Función objetivo

$$\min_{x \in \Re^n} f(x), \; f : \Re^n \rightarrow \Re, \quad (1)$$

Donde f es una función no lineal, por ejemplo:

1. $f(x_1, x_2) = x_1^2 + sen(x_1 * x_2) - x_1 * x_2$
2. $f(x_1, \ldots, x_n) = \sum_{i=1}^{n}(x_i - cos(x_i))^2$
3. $f(x_1, \ldots, x_n) = \sum_{i=1}^{n}(x_i - g(x_i, \xi_i))^2$

¿Por qué nos interesa?

El perceptron...

$$\min_{w \in \Re^n} E(w), \; E : \Re^n \to \Re, \tag{2}$$

- $E(w_1, \ldots, w_n) = \frac{1}{2} \sum_{\mu=1}^{p} (\zeta^\mu - g(\sum_{i=1}^{n} w_i \xi_i^\mu))^2$ es la función de error.
- $w = (w_1, \ldots, w_n)$ es el vector de pesos sinápticos que queremos hallar.
- n es la cantidad de pesos sinápticos.

Métodos de Resolución

- Métodos exactos: Calculan una fórmula cerrada para la solución.
- Métodos de aproximación de la solución:
    - Métodos basados en las derivadas primeras, o en el gradiente.
    - Métodos basados en las derivadas segundas, o en el hessiano.
    - Métodos sin derivadas.
    - Métodos Estocásticos (estiman la dirección).

Matriz definida positiva

Una matriz A es definida positiva

- Si $x^t Ax > 0, \; x \neq 0$.
- Si todos sus autovalores son positivos
- Hay otras características que definen a las matrices definidas positivas, por ejemplo los valores de la diagonal deben ser mayores que la suma de los otros elementos de la fila.
- Si $x^t Ax \geq 0, \; x \neq 0$, entonces se dice que la matriz es **Semi definida Positiva**

Definiciones

Sea la función f diferenciable con primera y segunda derivadas continuas, entonces x^*...

- Es mínimo global si $\forall x, \ f(x^*) \leq f(x)$
- Es mínimo local si $f(x^*) \leq f(x) \ \forall x, \ \|x - x^*\| < \epsilon$

Condiciones de optimalidad

Condiciones necesarias

- **Condición necesaria de primer orden**: Si x^* es un mínimo local de f entonces $\nabla f(x^*) = 0$.
- **Condición necesaria de segundo orden**: Si x^* es un mínimo local de f entonces $\nabla f(x^*) = 0$ y $H_f(x^*)$ (el hessiano) es una matriz semidefinida positiva.

Condiciones de optimalidad

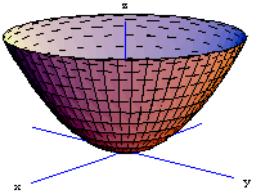
Condiciones suficientes

- Condición suficiente de primer orden:
  Si x^* es tal que $\nabla f(x^*) = 0$ y $H_f(x^*)$ es definida positiva, entonces es un mínimo local de f.

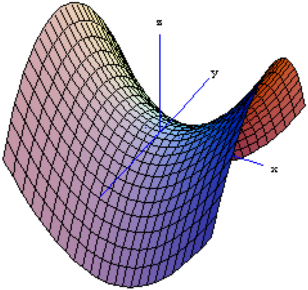
Las funciones

- Función convexa $\rightarrow$ Hessiano matriz definida positiva.
- Función cóncava $\rightarrow$ Hessiano matriz definida negativa.

Función cuadrática



Paraboloide elíptico



Paraboloide hiperbólico

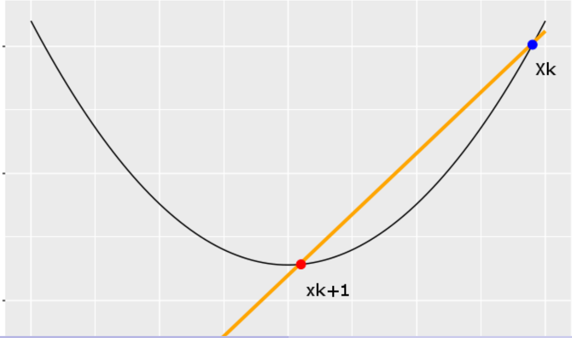
Hessiano: definida positiva

Hessiano: NO definida positiva

Procedimiento General de Optimización

Iterativamente, en el paso k:

- Punto inicial x_k, dato de entrada.
- Buscar una dirección de movimiento d_k.
- Calcular o decidir la longitud de paso α_k.
- Actualizar al nuevo punto $x_{k+1} = x_k + \alpha_k d_k$.



Procedimiento General de Optimización

La idea es

elegir el nuevo punto de manera que el valor de la función disminuya, o sea que $f(x_{k+1}) < f(x_k)$

Procedimiento General de Optimización

Condiciones sobre la dirección de movimiento

Debe ser descendiente, o sea $d_k^t \nabla f(x_k) < 0$.

- El gradiente es la dirección de máximo crecimiento de una función.
- Cualquier dirección contraria a la del gradiente, es una dirección de decrecimiento de la función.

Búsqueda Unidimensional: para encontrar α_k

Minimiza el valor de la función f sobre la recta en la que se está haciendo la búsqueda:

$$\alpha_k = \arg\min_{\alpha>0} g(\alpha) = f(x_k + \alpha d_k) \tag{3}$$

Tasa de aprendizaje

- α_k es la tasa de aprendizaje óptima.
- α_k fijo: es muy grande, el método puede no converger, si es muy pequeño converge muy lentamente.

Método del gradiente descendiente o máximo descenso

- La dirección de búsqueda para minimizar la función es

$$d_k = -\nabla f(x_k)$$

- No requiere el uso de segundas derivadas.
- Puede tener convergencia lenta.

Método del Gradiente Descendiente

Avanza en zig-zag

Método del Gradiente descendiente Con Momentum

Término regularizador

Consiste en tomar la dirección de descenso como una combinación lineal de direcciones de descenso calculadas en pasos anteriores.

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k) - \beta \alpha_{k-1} \nabla f(x_{k-1})$$

Este promedio ponderado entre direcciones suaviza el zig zagueo del método del gradiente.
$0 < \beta < 1$

Método de Newton

Aproximamos la función por el polinomio de Taylor de segundo orden,

$$\nabla f(x_{k+1}) = \nabla f(x_k + \alpha_k d_k) = \nabla f(x_k) + \alpha_k H(x_k) d_k$$

Entonces, si x_{k+1} fuera el mínimo, $\nabla f(x_{k+1}) = 0$ y por lo tanto

$$\nabla f(x_k) + \alpha_k H(x_k) d_k = 0$$

de donde resulta

$$d_k = -H^{-1}(x_k) \nabla f(x_k)$$

Métodos Quasi Newton

- La idea es disminuir el costo computacional asociado a calcular el hessiano y su inversa.
- Se basan en aproximar la matriz $H^{-1}(x_k)$
- La reemplazan por una matriz aproximada, definida positiva B_k.
- Diferentes métodos cuasi Newton difieren en la forma de aproximar esta matriz.

Pero...

Los métodos quasi Newton no pueden utilizarse para resolver problemas de Redes Neuronales porque poseen un alto costo computacional.

En su lugar, se puede utilizar el método **L-BFGS (limited memory BFGS)**

Métodos de Direcciones Conjugadas

Definición

Sea el conjunto de direcciones $d_1, \ldots, d_n$ y A una matriz simétrica definida positiva, entonces:

- Si $d_i^t A d_j = 0$, $\forall i \neq j$ entonces se dice que $d_1, \ldots, d_n$ son direcciones A-conjugadas.
- Si un conjunto de vectores es A-conjugado, con A simétrica, definida positiva entonces es también un conjunto linealmente independiente.

Métodos de Direcciones Conjugadas

Teorema: Dada una función cuadrática $f : \Re^n \to \Re$
$f(x) = x^t H x + b^t x$

si un método de minimización no lineal realiza las búsquedas unidimensionales sobre direcciones H-conjugadas, entonces el método converge en n pasos.

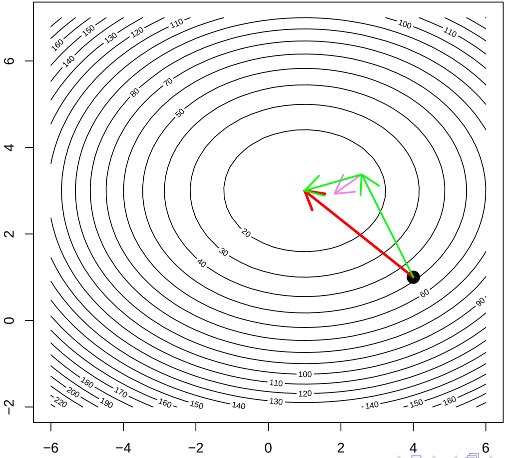
Método de gradientes conjugados (1952)

En cada paso k

Calcula una nueva dirección d_{k+1} que es $H(x_k)$-conjugada con todas las direcciones anteriores $d_1, \ldots, d_k$.

Problema: Hay que conocer el gradiente y el hessiano.

Método de gradientes conjugados



Método de direcciones conjugadas, M. Powell 1964

No necesita derivadas

Este método genera en cada paso k un conjunto de direcciones conjugadas $d_1^k, \ldots d_n^k$.

- Comienza con un conjunto de direcciones l.i $d_1^0, \ldots d_n^0$ y un punto inicial x_0.
- En el paso k, saca del conjunto la dirección d_k^k y agrega una dirección conjugada con $d_1^{k-1}, \ldots d_{k-1}^{k-1}$ (las de los pasos anteriores).
- La búsqueda lineal es igual que antes.

Método de direcciones conjugadas, M. Powell 1964

Con este método...

Si la función es convexa, entonces el método converge en n pasos.

Observaciones

- Todos estos métodos fueron desarrollados antes (o al mismo tiempo) de la aparición de Redes Neuronales.
- Incluso un método que converge en n pasos puede ser demasiado costoso para resolver un problema de redes neuronales.
- Aparece una nueva línea de investigación: desarrollo de métodos de optimización para resolver problemas de *Machine Learning*

Métodos Estocásticos

Solo sirven para minimizar funciones de error.

Problema

Sean $\xi_1^\mu, \ldots \xi_n^\mu$, $\mu = 1, \ldots, p$ las observaciones del conjunto de entrenamiento junto con su clasificación ζ^μ y $\mathbf{w} = (w_1, \ldots, w_n)$ el vector de pesos sinápticos. Entonces, queremos hallar $\mathbf{w}$ que minimice $E(\mathbf{w})$:

$$E(\mathbf{w}) = \frac{1}{2}\sum_{\mu=1}^{p}(\zeta^\mu - \sum_{i=1}^{n} w_i \xi_i^\mu) \tag{4}$$

Métodos Estocásticos

Entonces, podemos pensar

$$E(\mathbf{w}) = \frac{1}{p}\sum_{\mu=1}^{p} E^{\mu}(\xi^{\mu}, \mathbf{w}) \tag{5}$$

El gradiente descendiente haría:

Solución del Gradiente Descendiente

$$\mathbf{w}^{t+1} = \mathbf{w}^{t} - \eta_t \frac{1}{p} \sum_{\mu=1}^{p} \nabla_{\mathbf{w}} E^{\mu}(\xi^{\mu}, \mathbf{w}^{t})$$

Gradiente Descendiente Estocástico

Pero...

$\frac{1}{p}\sum_{\mu=1}^{p}\nabla_w E(\xi^\mu,\mathbf{w}^t)$ es un estimador de la esperanza del gradiente, dado un conjunto de entrenamiento, entonces...
¿Por qué no usar cualquier otro estimador?

Por ejemplo: un $\nabla_w E(\xi^\nu,\mathbf{w}^t)$, para algún ν arbitario o

Minibach

un subconjunto aleatorio $\sum_{\mu=1}^{k}\nabla_w E(\xi^\mu,\mathbf{w}^t),\ k << p$.

GD vs GD Estocástico

El nombre original de este método fue ADALINE (ADA: Adaptative)

La diferencia es que en el método GD Estocástico, solamente una parte de los datos se utiliza para calcular la dirección de descenso en cada paso.

Gradiente Descendiente Estocástico (1998)

Los autores del método, demuestran en su libro [1] que el método converge, pero no siempre va descendiendo.
No desesperar

ADAGrad, 2011 (Adaptative Gradient)

Modificación al método de gradientes estocásticos

Es un método que tiene como principal objetivo adaptar el valor de la tasa de aprendizaje en cada paso y la actualización del vector **w** se realiza coordenada a coordenada.

Sea $g_t = \nabla E(\mathbf{w}_{t-1})$ entonces $(g_t)_i$ es la i-ésima coordenada $g_{1:t} = \{g_1, \ldots, g_t\}$ son todos los gradientes anteriores hasta el paso t.

La modificación

SGD haría

$$\mathbf{w}^{t+1} = \mathbf{w}^{t} - \eta_t \nabla_{\mathbf{w}} E^{\mu}(\xi^{\mu}, \mathbf{w}^{t})$$

ADAGrad

$$\mathbf{w}_i^{t+1} = \mathbf{w}_i^{t} - \frac{\eta_t}{\sqrt{G_{ii}^t + \epsilon}} \nabla_{\mathbf{w_i}} E^{\mu}(\xi^{\mu}, \mathbf{w}^{t})$$

$G^t = \sum_{\tau=1}^{t} g_\tau * g_\tau^t$

La ventaja de este método es que cada coordenada tiene su propia actualización.

Método ADAM-2015 (Adaptive Moment Estimation)

Algoritmo

Requiere:

- α tasa de aprendizaje.
- β_1 y β_2 tasas de decaimiento.
- f la función objetivo.

Inicialización:

- w_0 parámetro inicial.
- $m_0 = 0$
- $v_0 = 0$

Método ADAM-2015 (Adaptive Moment Estimation)

Algoritmo

while w_t not converge do
$t := t + 1$
$g_t = \nabla f(w_{t-1})$
$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$
$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$
$w_t = w_{t-1} - \alpha \frac{m_t}{\sqrt{v_t + \epsilon}}$

El autor sugiere $\beta_1 = 0{,}9$, $\beta_1 = 0{,}999$ y $\epsilon = 10^{-8}$

Referencias

Un libro recomendado [2]

[1] L. Bottou. *Online Algorithms and Stochastic Approximations. Online Learning and Neural Networks.* Cambridge University Press., 1998.

[2] Richard P. Brent. *Algorithms for minimization without derivatives*. Englewood Cliffs, N.J., Prentice-Hall, 1973.