

Métodos de Optimización No Lineal Sin Restricciones: Un resumen

10 de abril de 2025

El Problema: Optimización

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

Optimización

Diferentes Alternativas

- Optimización Discreta: Programación entera y combinatoria
- Optimización Discreta: Ecuación diofántica
- Optimización Lineal: Programación Lineal
- Optimización No Lineal sin Restricciones Convexa: garantía de extremos globales.
- Optimización No Convexa: Primer orden: GD, SGD, ADAM,
- Optimización No Convexa: Segundo orden: Newton
- Optimización No Convexa: Cero orden: Bayesiana, Powel, Sim Optimistic Optimization

Optimización

Historia

- 1847: Cauchy gradiente descendente
- 1950s: Programación lineal, SGD
- 1980s: Teoría Convexa, Optimización General
- 2005-2015: Optimización a gran escala sobre problemas convexos.
- 2015-ahora: Deep Learning empuja una explosión en su entendimiento.

¿Por qué Cálculo es importante?



Dada una lata de coca, un tema práctico que puede buscarse es encontrar cuales tienen que ser las dimensiones de una lata cilíndrica que contenga un litro, para minimizar su superficie y así la cantidad de aluminio que hay que utilizar.

¿Por qué Cálculo es importante?



$$S(r, h) = 2 \pi r^2 + 2 \pi r h \quad (1)$$

$$g(r, h) = \pi r^2 h = 1000 \quad (2)$$

¿Por qué Cálculo es importante?

$$g(r, h) = \pi r^2 h = 1000 \text{ cm}^3$$

$$h = \frac{1000 \text{ cm}^3}{\pi r^2}$$

Reemplazando en 1

$$S(r) = 2 \pi r^2 + 2 \pi r \left(\frac{1000 \text{ cm}^3}{\pi r^2} \right)$$

$$S(r) = 2 \pi r^2 + \frac{2000 \text{ cm}^3}{r}$$

¿Por qué Cálculo es importante?

El punto extremo ocurre cuando la derivada de $S(r)$ respecto a r es cero. Esto es

$$S'(r) = 4 \pi r - \frac{2000 \text{ cm}^3}{r^2} = 0$$

$$4 \pi r = \frac{2000 \text{ cm}^3}{r^2}$$

$$4 \pi r^3 = 2000 \text{ cm}^3$$

$$r^3 = \frac{500 \text{ cm}^3}{\pi}$$

$$r = \sqrt[3]{\frac{500 \text{ cm}^3}{\pi}}$$

¿Por qué Cálculo es importante?

Como el punto que se desea encontrar es el mínimo, esto ocurre cuando la derivada pasa de negativa a positiva, y eso es justamente en la solución positiva de $r = 5,41 \text{ cm}$, lo que provoca un valor de $h = 58 \text{ cm}$ para la altura

El Problema: Optimización

Función objetivo

$$\min_{x \in \mathbb{R}^n} f(x), \quad f : \mathbb{R}^n \rightarrow \mathbb{R}, \quad \vec{x} = [x_1, x_2, \dots, x_n] \quad (3)$$

$$g_i(\vec{x}) \leq 0 \quad (4)$$

$$h_j(\vec{x}) = 0 \quad (5)$$

Donde

- 1 \vec{x} variables de diseño, $f(\vec{x})$ es la función objetivo/costo
- 2 g_i y h_j son restricciones
- 3 \vec{x}^* resuelve el problema para $f(\vec{x}^*)$

Programación Lineal: Simplex

$$\text{Maximize } x_1 - x_2$$

subject to

$$x_1 + x_2 \leq 1, \quad -x_1 + 2x_2 \leq 2,$$

$$x_1 \geq -1, \quad -x_1 + 3x_2 \geq -3.$$

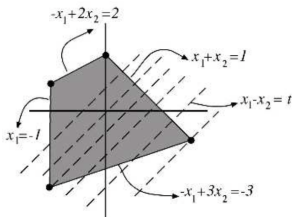


Figure 2.1. The feasible region and level curves in an LPP.

El Problema: Optimización NO Lineal sin restricciones

Función objetivo

$$\min_{x \in \mathbb{R}^n} f(x), \quad f : \mathbb{R}^n \rightarrow \mathbb{R}, \quad (6)$$

Donde f es una función no lineal, por ejemplo:

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

¿Por qué nos interesa?

Nos interesa porque detrás de todo algoritmo de aprendizaje automático hay un mecanismo de optimización matemática que resuelve el **aprendizaje automático**.

Optimización en Redes Neuronales

$$\min_{w \in \mathbb{R}^n} E(w), \quad E : \mathbb{R}^n \rightarrow \mathbb{R}, \quad (7)$$

- $E(w_1, \dots, w_n) = \frac{1}{2} \sum_{\mu=1}^p (\zeta^\mu - g(\sum_{i=1}^n w_i \xi_i^\mu))^2$ es la función de error.
- $w = (w_1, \dots, w_n)$ 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 $x^t A x > 0$, $x \neq 0$.
- 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 $x^t A x \geq 0$, $x \neq 0$, 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 .

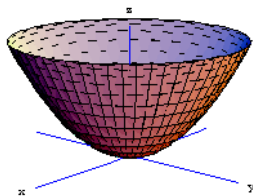
Funciones Convexas y Cóncavas

Las funciones **convexas** son aquellas que son diferenciables y donde el punto medio entre dos puntos siempre es menor o igual a la media aritmética entre los dos extremos. Las **cóncavas** se definen de la misma forma tomando el valor inverso de la función.

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

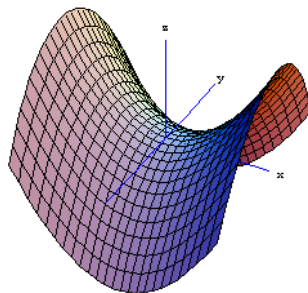
Las funciones convexas permiten atacar el problema de la optimización con un modelo más simplificado intermedio que ofrece condiciones de optimalidad.

Función cuadrática



Paraboloide elíptico

Hessiano: definida positiva



Paraboloide hiperbólico

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 $f(x_{k+1}) < f(x_k)$

Con el algoritmo general, el problema se reduce a:

- Encontrar la dirección de movimiento d_k .
- Encontrar el paso adecuado α_k .

Procedimiento General de Optimización

Dirección de Decrecimiento

- La dirección en la cuál realizar la búsqueda del próximo x_{k+1} .

Condiciones sobre la dirección de movimiento

Por supuesto, la idea es que sea descendente, o sea $d_k^t \nabla f(x_k) < 0$.

- Recordar que 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.

Procedimiento General de Optimización

Tasa de aprendizaje

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

Búsqueda Unidimensional: para encontrar α_k

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

$$\alpha_k = \arg \min_{\alpha > 0} g(\alpha) = f(x_k + \alpha d_k) \quad (8)$$

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, como una pelota que cae por un cuenco y que cambia de dirección abruptamente.

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 zagado del método del gradiente. El hiperparámetro β , $0 < \beta < 1$, equivale a la *inercia* del sistema, y a mayor valor, es más difícil cambiar la dirección (se suaviza el zig zag).

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 Broyden Fletcher Goldfarb Shanno (BFGS) algorithm)**. Este algoritmo, siguiendo con la idea del huevo y la gallina, implementa una búsqueda iterativa para estimar el Hessiano B_{k+1} B_k .

Métodos de Direcciones Conjugadas

Definición

Sea el conjunto de direcciones d_1, \dots, 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, \dots, 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 $f : \mathbb{R}^n \rightarrow \mathbb{R}$
 $f(x) = x^t H x + b^t x$

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, \dots, d_k .

Problema: Hay que conocer el gradiente y el hessiano.

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, \dots, d_n^k .

- Comienza con un conjunto de direcciones l.i d_1^0, \dots, 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 $d_1^{k-1}, \dots, d_{k-1}^{k-1}$ (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*

Funciones de Error

Problema

Supongamos que tenemos un problema de aprendizaje supervisado, donde queremos aprender una función $f(\xi) = \zeta$, donde tenemos p observaciones $\xi^\mu = \{\xi_1, \dots, \xi_n\}$, $\mu = 1, \dots, p$ de dimensión n del conjunto de entrenamiento (ξ) junto con sus p salidas ζ^μ de dimensión 1.

Queremos encontrar una función $g(\cdot)$ que aproxime $f(\cdot)$ lo mejor posible. Para eso podemos plantear

$$E(\cdot) = \frac{1}{2} \sum_{\mu=1}^p (\zeta^\mu - g(\xi_i^\mu))^2 \quad (9)$$

Funciones de Error

Problema

Si hacemos que $g(\cdot)$ dependa de parámetros libres, que vamos a llamar $\mathbf{w} = (w_1, \dots, w_n)$, lo que podemos hacer es plantear esto como un problema de optimización e intentar buscar los w que minimizan (9) y así encontrar w^* con alguna formulación para g , como por ejemplo combinación lineal $g(\xi_i^\mu) = \sum_{i=1}^n w_i \xi_i^\mu$.

$$E(\mathbf{w}) = \frac{1}{2} \sum_{\mu=1}^p (\zeta^\mu - \sum_{i=1}^n w_i \xi_i^\mu) \quad (10)$$

Así el problema de aproximar f con g se reduce a **optimizar** g en base a los parámetros libres w de forma que se minimice $E(\mathbf{w})$.

Métodos Estocásticos

Solo sirven para minimizar funciones de error.

Recapitulando el problema de aproximación

Se tienen p observaciones $\xi^\mu = \{\xi_1, \dots, \xi_n\}$, $\mu = 1, \dots, p$ de dimensión n del conjunto de entrenamiento (X) junto con sus p salidas ζ^μ de dimensión 1 y $\mathbf{w} = (w_1, \dots, w_n)$ el vector de pesos sinápticos (parámetros libres). Entonces, queremos hallar \mathbf{w} que minimice $E(\mathbf{w})$:

$$E(\mathbf{w}) = \frac{1}{2} \sum_{\mu=1}^p (\zeta^\mu - \sum_{i=1}^n w_i \xi_i^\mu)^2 \quad (11)$$

Métodos Estocásticos

Entonces, podemos pensar

$$E(\mathbf{w}) = \frac{1}{p} \sum_{\mu=1}^p E^{\mu}(\zeta^{\mu}, \xi^{\mu}, \mathbf{w}) \quad (12)$$

Es decir, a la función a optimizar la podemos visualizar con el cálculo de un estimador del comportamiento poblacional de los patrones.

Métodos Estocásticos

Con esto el cálculo del gradiente descendiente y la formula de actualización sería:

Solución del Gradiente Descendiente Estocástico

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

con lo que se calcularían los parámetros libres w en relación a la formulación estocástica aproximada del verdadero gradiente.

Gradiente Descendiente Estocástico

Estimador del Gradiente

$\frac{1}{p} \sum_{\mu=1}^p \nabla_w E(\xi^\mu, \mathbf{w}^t)$ es un estimador de la esperanza del gradiente, dado un conjunto de entrenamiento.

Una solución computacionalmente adecuada, es obtener $\nabla_w E(\xi^\nu, \mathbf{w}^t)$ para algún subconjunto ν arbitrario de los patrones (en vez de elegirlos a todos). Así surgen los *minibatch*.

Minibatch

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

Gradiente Descendiente Estocástico

El nombre original de este método fue ADALINE (Adaptive Linear Element)

Gradient Descend (GD) vs Stochastic Gradient Descend (SGD)

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.

Los autores del método, demuestran en su libro [1] (1998) que el método converge, pero no siempre va descendiendo.

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 \mathbf{w} se realiza coordenada a coordenada.

Sea $g_t = \nabla E(\mathbf{w}_{t-1})$ entonces $(g_t)_i$ es la i -ésima coordenada $g_{1:t} = \{g_1, \dots, g_t\}$ 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

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

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

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 $\beta_1 = 0,9$, $\beta_2 = 0,999$ y $\epsilon = 10^{-8}$

Referencias

Un libro recomendado [3] También [4],y [5] Un libro clásico popular en ingeniería es [2]

- [1] L. Bottou. *Online Algorithms and Stochastic Approximations. Online Learning and Neural Networks.* Cambridge University Press., 1998.
- [2] Stephen P Boyd and Lieven Vandenberghe. *Convex optimization.* Cambridge university press, 2004.
- [3] Richard P. Brent. *Algorithms for minimization without derivatives.* Englewood Cliffs, N.J., Prentice-Hall, 1973.
- [4] Pablo Pedregal. *Introduction to Optimization.* Springer, 2003.
- [5] Jan A Snyman, Daniel N Wilke, et al. *Practical mathematical optimization.* Springer, 2005.