

Introdução à Otimização Matemática e Algumas Aplicações

Dr Majela Pentón Machado

Instituto de Matemática e Estatística, Universidade Federal da Bahia

11º WorkShop de Soluções Matemáticas para Problemas Industriais

21 março de 2026

Realização:



CeMEAI
CEPID - Centro de Ciências
Matemáticas Aplicadas e Indústria



UFBA
Universidade
Federal da Bahia

Apoio:



COMISSÃO FEDERAL DE FOMENTO INDUSTRIAL DE CARAJÁS



40 ANOS



International
Statistical
Institute

Problemas de Otimização

Problema de localização

Regressão linear

Redes neurais

Problemas de Otimização

O que é otimização?

Ideia central

Otimizar significa escolher, entre várias possibilidades, aquela que **melhor atende a um certo critério**.

O que é otimização?

Ideia central

Otimizar significa escolher, entre várias possibilidades, aquela que **melhor atende a um certo critério**.

Exemplo:

- ▶ Quero realizar uma viagem de Salvador ao Rio de Janeiro no menor tempo possível.

O que é otimização?

Ideia central

Otimizar significa escolher, entre várias possibilidades, aquela que **melhor atende a um certo critério**.

Exemplo:

- ▶ Quero realizar uma viagem de Salvador ao Rio de Janeiro no menor tempo possível.
- ▶ Ir de ônibus;
- ▶ Ir de avião;

O que é otimização?

Ideia central

Otimizar significa escolher, entre várias possibilidades, aquela que **melhor atende a um certo critério**.

Exemplo:

- ▶ Quero realizar uma viagem de Salvador ao Rio de Janeiro no menor tempo possível.
- ▶ Ir de ônibus;
- ▶ Ir de avião;
- ▶ Não posso gastar mais de 1000 reais na viagem
- ▶ Devo estar no Rio no dia 22/03/2026.

O que é otimização?

Outros exemplos

- ▶ Qual é a melhor rota para fazer entregas?
- ▶ Como ajustar uma reta aos dados observados?
- ▶ Como treinar uma rede neural para reduzir o erro?
- ▶ Como distribuir produtos minimizando custos?

O que é otimização?

Outros exemplos

- ▶ Qual é a melhor rota para fazer entregas?
- ▶ Como ajustar uma reta aos dados observados?
- ▶ Como treinar uma rede neural para reduzir o erro?
- ▶ Como distribuir produtos minimizando custos?

Ou seja, estamos otimizando toda vez que resolvemos um problema que envolva encontrar o melhor, o menor, o mais rápido, o mais barato...

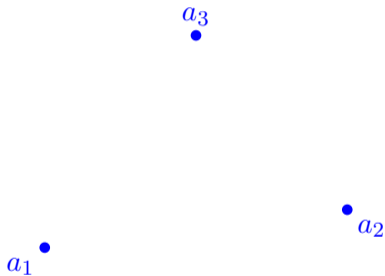
Problemas de otimização

Elementos básicos de um problema de otimização

- ▶ **Variáveis de decisão:** o que pode ser escolhido
- ▶ **Função objetivo:** o que queremos minimizar ou maximizar
- ▶ **Restrições:** condições que a solução deve satisfazer

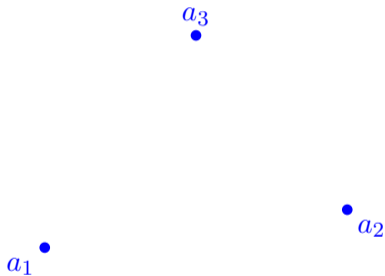
Exemplo: onde construir um ponto de encontro?

Suponha que três cidades estejam localizadas nos pontos a_1 , a_2 e a_3 do plano.



Exemplo: onde construir um ponto de encontro?

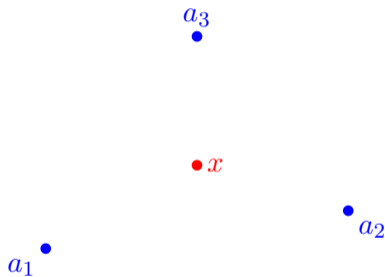
Suponha que três cidades estejam localizadas nos pontos a_1 , a_2 e a_3 do plano.



Pergunta: Onde construir um ponto de encontro para essas três cidades?

Exemplo: onde construir um ponto de encontro?

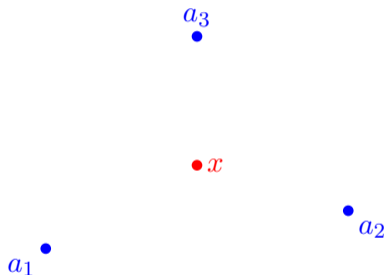
Suponha que três cidades estejam localizadas nos pontos a_1 , a_2 e a_3 do plano.



Pergunta: Onde construir um ponto de encontro para essas três cidades?

Exemplo: onde construir um ponto de encontro?

Suponha que três cidades estejam localizadas nos pontos a_1 , a_2 e a_3 do plano.



Pergunta: Onde construir um ponto de encontro para essas três cidades?

Objetivo: Queremos escolher um ponto no plano que minimize a distância total até essas cidades.

Por que otimização?

- ▶ x : onde vamos construir
- ▶ a_i : as cidades

Por que otimização?

- ▶ x : onde vamos construir
- ▶ a_i : as cidades

Queremos escolher um ponto x que minimize a distância total até as três cidades:

$$\min_x (\|x - a_1\| + \|x - a_2\| + \|x - a_3\|).$$

Por que otimização?

- ▶ x : onde vamos construir
- ▶ a_i : as cidades

Queremos escolher um ponto x que minimize a distância total até as três cidades:

$$\min_x (\|x - a_1\| + \|x - a_2\| + \|x - a_3\|).$$

Estamos tentando escolher a **melhor posição** x .

- ▶ x : variável de decisão
- ▶ soma das distâncias: função objetivo

Exemplo: produção e maximização de lucro

Uma empresa produz dois produtos: x_1 e x_2 .

Cada unidade gera lucro:

- ▶ produto 1: R\$ 3
- ▶ produto 2: R\$ 2

Exemplo: produção e maximização de lucro

Uma empresa produz dois produtos: x_1 e x_2 .

Cada unidade gera lucro:

- ▶ produto 1: R\$ 3
- ▶ produto 2: R\$ 2

A produção é limitada por recursos:

$$\begin{cases} x_1 + x_2 \leq 4 & \text{(matéria-prima)} \\ 2x_1 + x_2 \leq 5 & \text{(tempo de máquina)} \\ x_1, x_2 \geq 0 \end{cases}$$

Exemplo: produção e maximização de lucro

Uma empresa produz dois produtos: x_1 e x_2 .

Cada unidade gera lucro:

- ▶ produto 1: R\$ 3
- ▶ produto 2: R\$ 2

A produção é limitada por recursos:

$$\begin{cases} x_1 + x_2 \leq 4 & \text{(matéria-prima)} \\ 2x_1 + x_2 \leq 5 & \text{(tempo de máquina)} \\ x_1, x_2 \geq 0 \end{cases}$$

Objetivo: maximizar o lucro total.

Por que otimização?

Elementos do problema

- ▶ **Variáveis:** quantidade produzida x_1, x_2
- ▶ **Função objetivo:** lucro total

Por que otimização?

Elementos do problema

- ▶ **Variáveis:** quantidade produzida x_1, x_2
- ▶ **Função objetivo:** lucro total

$$\max_{x_1, x_2} 3x_1 + 2x_2$$

Por que otimização?

Elementos do problema

- ▶ **Variáveis:** quantidade produzida x_1, x_2
- ▶ **Função objetivo:** lucro total

$$\max_{x_1, x_2} 3x_1 + 2x_2$$

Nem toda escolha é possível

- ▶ **Há restrições:** limites de recursos
- ▶ Queremos a melhor escolha **dentro da região viável**

Formulação

Em linguagem matemática, muitos desses problemas podem ser escritos como

$$\begin{aligned} \min f(x) \\ \text{s.a. } x \in \Omega \end{aligned}$$

onde $f : \mathbb{R}^n \rightarrow \mathbb{R}$ e $\Omega \subset \mathbb{R}^n$.

Formulação

Em linguagem matemática, muitos desses problemas podem ser escritos como

$$\begin{aligned} \min f(x) \\ \text{s.a. } x \in \Omega \end{aligned}$$

onde $f : \mathbb{R}^n \rightarrow \mathbb{R}$ e $\Omega \subset \mathbb{R}^n$.

- ▶ x variável de decisão
- ▶ f função objetivo que mede o custo, erro ou distância associados a essa decisão
- ▶ Ω conjunto viável do problema

Formulação

Em linguagem matemática, muitos desses problemas podem ser escritos como

$$\begin{aligned} \min f(x) \\ \text{s.a. } x \in \Omega \end{aligned}$$

onde $f : \mathbb{R}^n \rightarrow \mathbb{R}$ e $\Omega \subset \mathbb{R}^n$.

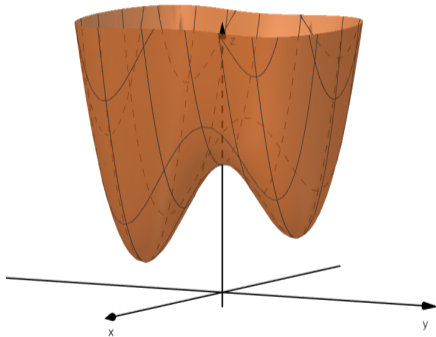
- ▶ x variável de decisão
- ▶ f função objetivo que mede o custo, erro ou distância associados a essa decisão
- ▶ Ω conjunto viável do problema
- ▶ Problema irrestrito: $\Omega = \mathbb{R}^n$
- ▶ Problema com restrições: $\Omega \subset \mathbb{R}^n$

Tipos de solução

- ▶ **Mínimo global:** melhor solução possível

$$f(x^*) \leq f(x), \quad \forall x$$

- ▶ $f(x^*)$ é chamado de valor ótimo do problema.



Tipos de solução

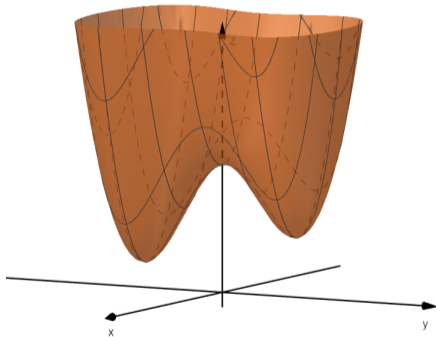
- ▶ **Mínimo global:** melhor solução possível

$$f(x^*) \leq f(x), \quad \forall x$$

- ▶ $f(x^*)$ é chamado de valor ótimo do problema.

- ▶ **Mínimo local:** melhor solução em uma região próxima

$$f(x^*) \leq f(x), \quad \text{para } x \text{ próximo de } x^*$$



Nos cursos de Cálculo

Otimização

- ▶ Valores máximos e mínimos
- ▶ Teorema do valor extremo
- ▶ Teorema de Fermat (condição de otimalidade)
- ▶ Método do intervalo fechado
- ▶ Teste da primeira derivada
- ▶ Teste da segunda derivada
- ▶ Multiplicadores de Lagrange

Teste da primeira derivada

Se queremos minimizar uma função diferenciável no \mathbb{R}^n :

$$\begin{aligned} \min f(x) \\ \text{s.a. } x \in \mathbb{R}^n \end{aligned}$$

Teste da primeira derivada

Se queremos minimizar uma função diferenciável no \mathbb{R}^n :

$$\begin{aligned} \min f(x) \\ \text{s.a. } x \in \mathbb{R}^n \end{aligned}$$

Condição necessária de primeira ordem

Se x^* é um mínimo local de f , então

$$\nabla f(x^*) = 0.$$

Teste da primeira derivada

Se queremos minimizar uma função diferenciável no \mathbb{R}^n :

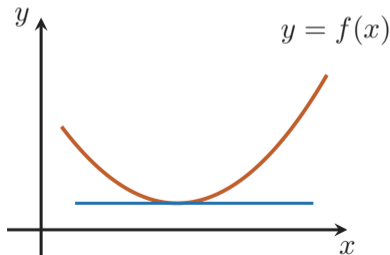
$$\begin{aligned} \min f(x) \\ \text{s.a. } x \in \mathbb{R}^n \end{aligned}$$

Condição necessária de primeira ordem

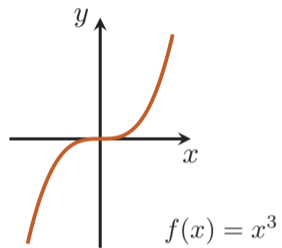
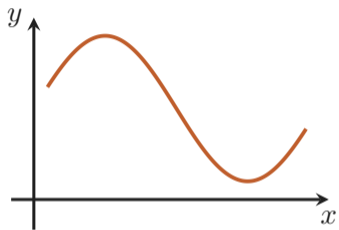
Se x^* é um mínimo local de f , então

$$\nabla f(x^*) = 0.$$

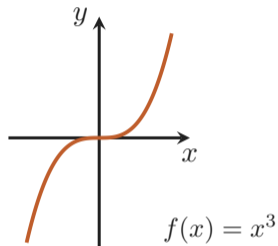
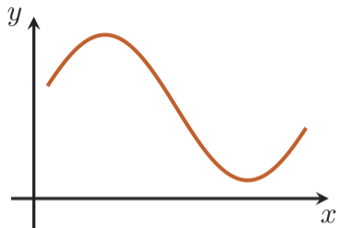
- ▶ O gradiente indica a direção de crescimento
- ▶ No mínimo, não há direção de descida



Não é suficiente



Não é suficiente



Ponto estacionário

Todo x tal que $\nabla f(x) = 0$ é chamado de **ponto estacionário** do problema.

Como distinguir um mínimo?

Se $\nabla f(x^*) = 0$, ainda precisamos verificar o tipo de ponto.

- ▶ Pode ser mínimo
- ▶ Pode ser máximo
- ▶ Pode ser ponto de sela

Como distinguir um mínimo?

Se $\nabla f(x^*) = 0$, ainda precisamos verificar o tipo de ponto.

- ▶ Pode ser mínimo
- ▶ Pode ser máximo
- ▶ Pode ser ponto de sela

A curvatura da função ajuda a distinguir esses casos.

Teste da segunda derivada

Se f é uma função duas vezes diferenciável.

Condição necessária de segunda ordem

Se x^* é um **mínimo local** de f , então

- ▶ x^* é um **ponto estacionário** ($\nabla f(x^*) = 0$)
- ▶ a matriz Hessiana de f no ponto x^* é **semidefinida positiva** ($f''(x^*) \geq 0$)

Teste da segunda derivada

Se f é uma função duas vezes diferenciável.

Condição necessária de segunda ordem

Se x^* é um **mínimo local** de f , então

- ▶ x^* é um **ponto estacionário** ($\nabla f(x^*) = 0$)
- ▶ a matriz Hessiana de f no ponto x^* é **semidefinida positiva** ($f''(x^*) \geq 0$)

Condição suficiente de segunda ordem

Se x^* é um **ponto estacionário** ($\nabla f(x^*) = 0$) tal que a Hessiana de f em x^* é **definida positiva** ($f''(x^*) > 0$), então

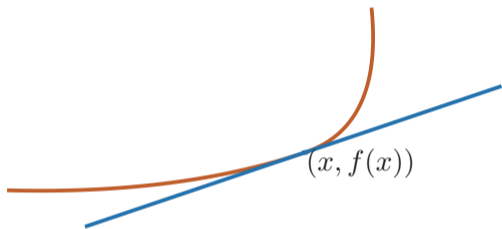
- ▶ x^* é um **mínimo local** de f .

Funções convexas

Uma função $f : \mathbb{R}^n \rightarrow \mathbb{R}$ diferenciável é dita convexa no \mathbb{R}^n se

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle$$

para todo $x, y \in \mathbb{R}^n$.

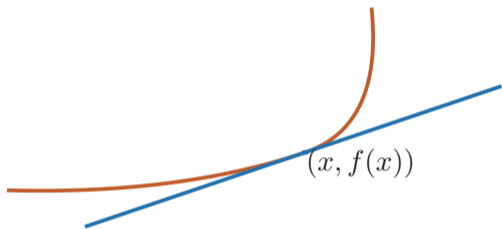


Funções convexas

Uma função $f : \mathbb{R}^n \rightarrow \mathbb{R}$ diferenciável é dita convexa no \mathbb{R}^n se

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle$$

para todo $x, y \in \mathbb{R}^n$.



Condição suficiente de otimalidade

Se f é uma função convexa, diferenciável, e $x^* \in \mathbb{R}^n$ é um ponto estacionário, então

$$f(x) \geq f(x^*) + \langle \nabla f(x^*), y - x^* \rangle = f(x^*)$$

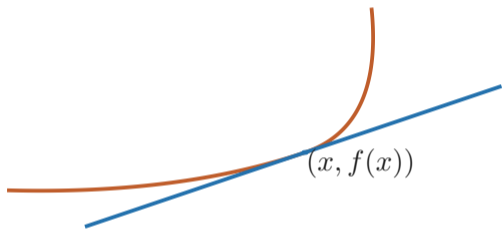
para todo $x \in \mathbb{R}^n$.

Funções convexas

Uma função $f : \mathbb{R}^n \rightarrow \mathbb{R}$ diferenciável é dita convexa no \mathbb{R}^n se

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle$$

para todo $x, y \in \mathbb{R}^n$.



Condição suficiente de otimalidade

Se f é uma função convexa, diferenciável, e $x^* \in \mathbb{R}^n$ é um ponto estacionário, então

$$f(x) \geq f(x^*) + \langle \nabla f(x^*), y - x^* \rangle = f(x^*)$$

para todo $x \in \mathbb{R}^n$.

► x^* é **mínimo global**

Problema de localização

Problema de localização

Pergunta:

Onde devemos construir uma instalação (hospital, escola, depósito, antena) para atender melhor uma população?

Problema de localização

Pergunta:

Onde devemos construir uma instalação (hospital, escola, depósito, antena) para atender melhor uma população?

Dados:

- ▶ Pontos $a_1, \dots, a_m \in \mathbb{R}^2$ (locais de demanda)
- ▶ Pesos $w_1, \dots, w_m > 0$ (importância de cada ponto)

Problema de localização

Pergunta:

Onde devemos construir uma instalação (hospital, escola, depósito, antena) para atender melhor uma população?

Dados:

- ▶ Pontos $a_1, \dots, a_m \in \mathbb{R}^2$ (locais de demanda)
- ▶ Pesos $w_1, \dots, w_m > 0$ (importância de cada ponto)

Objetivo: encontrar a melhor localização $x \in \mathbb{R}^2$.

Modelagem

Escolher um ponto $x \in \mathbb{R}^2$ que minimize a soma ponderada das distâncias até pontos dados a_1, \dots, a_m :

$$\min_{x \in \mathbb{R}^2} \sum_{i=1}^m w_i \|x - a_i\|$$

- ▶ a_i representam os locais de demanda e w_i os pesos.

Modelagem

Escolher um ponto $x \in \mathbb{R}^2$ que minimize a soma ponderada das distâncias até pontos dados a_1, \dots, a_m :

$$\min_{x \in \mathbb{R}^2} \sum_{i=1}^m w_i \|x - a_i\|$$

- ▶ a_i representam os locais de demanda e w_i os pesos.

Interpretação

- ▶ Cada termo mede um “custo de atendimento”
- ▶ A função objetivo soma esses custos

Modelagem

Escolher um ponto $x \in \mathbb{R}^2$ que minimize a soma ponderada das distâncias até pontos dados a_1, \dots, a_m :

$$\min_{x \in \mathbb{R}^2} \sum_{i=1}^m w_i \|x - a_i\|$$

- ▶ a_i representam os locais de demanda e w_i os pesos.

Interpretação

- ▶ Cada termo mede um “custo de atendimento”
- ▶ A função objetivo soma esses custos

Limitação: A função não é diferenciável quando $x = a_i$.

Um modelo mais fácil de resolver

Considerando as distâncias ao quadrado:

$$\min_{x \in \mathbb{R}^2} \sum_{i=1}^m w_i \|x - a_i\|^2$$

Um modelo mais fácil de resolver

Considerando as distâncias ao quadrado:

$$\min_{x \in \mathbb{R}^2} \sum_{i=1}^m w_i \|x - a_i\|^2$$

- ▶ A função agora é **diferenciável**

Um modelo mais fácil de resolver

Considerando as distâncias ao quadrado:

$$\min_{x \in \mathbb{R}^2} \sum_{i=1}^m w_i \|x - a_i\|^2$$

- ▶ A função agora é **diferenciável**

Consequências:

- ▶ Podemos usar condições de primeira ordem: $\nabla f(x) = 0$
- ▶ Isso permite encontrar a solução de forma simples

Um modelo mais fácil de resolver

Considerando as distâncias ao quadrado:

$$\min_{x \in \mathbb{R}^2} \sum_{i=1}^m w_i \|x - a_i\|^2$$

- ▶ A função agora é **diferenciável**

Consequências:

- ▶ Podemos usar condições de primeira ordem: $\nabla f(x) = 0$
- ▶ Isso permite encontrar a solução de forma simples

Observação: A escolha da modelagem influencia diretamente a dificuldade do problema.

Solução

Gradiente:

$$\nabla f(x) = 2 \sum_{i=1}^m w_i (x - a_i)$$

Solução

Gradiente:

$$\nabla f(x) = 2 \sum_{i=1}^m w_i (x - a_i)$$

Solução:

$$x^* = \frac{\sum_{i=1}^m w_i a_i}{\sum_{i=1}^m w_i}$$

Regressão linear

Regressão

Regressão é uma técnica de aprendizado de máquina (machine learning).

Regressão

Regressão é uma técnica de aprendizado de máquina (machine learning).

Conjunto de dados observados

$$D_m = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}$$

- ▶ cada $x^{(i)} \in \mathbb{R}^d$ representa um vetor de características
- ▶ $y^{(i)} \in \mathbb{R}$ a resposta observada.

Regressão

Regressão é uma técnica de aprendizado de máquina (machine learning).

Conjunto de dados observados

$$D_m = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}$$

- ▶ cada $x^{(i)} \in \mathbb{R}^d$ representa um vetor de características
- ▶ $y^{(i)} \in \mathbb{R}$ a resposta observada.

Objetivo

Dado um novo dado $x^{(m+1)}$ prever o valor de $y^{(m+1)}$

Exemplo

	Motor	Cilindros	Emissão de CO2
1	2.0	4	196
2	1.5	4	136
3	3.5	6	244
4	3.7	6	255
5	2.4	4	?

Exemplo

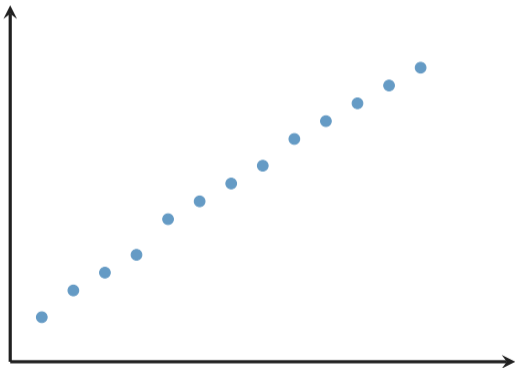
	Motor	Cilindros	Emissão de CO2
1	2.0	4	196
2	1.5	4	136
3	3.5	6	244
4	3.7	6	255
5	2.4	4	?

Pergunta

Como usar os dados das primeiras linhas para prever a emissão de CO₂ do carro da linha 5?

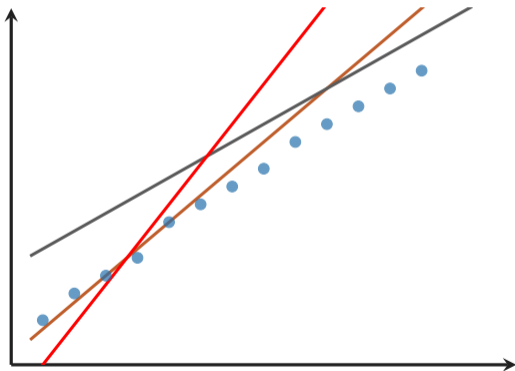
Regressão linear

É usada quando os dados podem ser bem aproximados por uma relação linear:



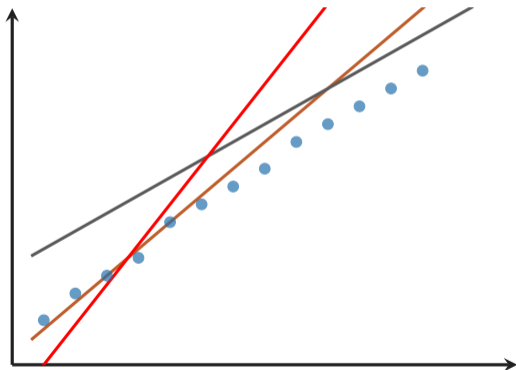
Regressão linear

É usada quando os dados podem ser bem aproximados por uma relação linear:



Regressão linear

É usada quando os dados podem ser bem aproximados por uma relação linear:

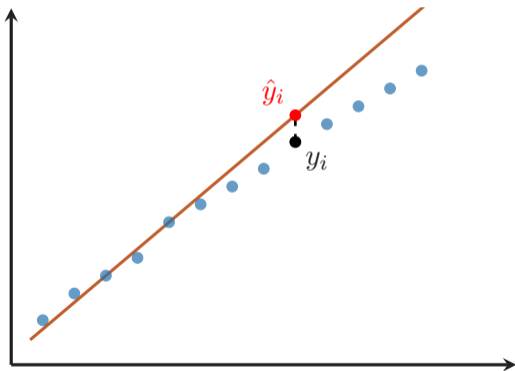


Modelo linear

$$\hat{y} = \theta^T x + b$$

Regressão linear

É usada quando os dados podem ser bem aproximados por uma relação linear:



Modelo linear

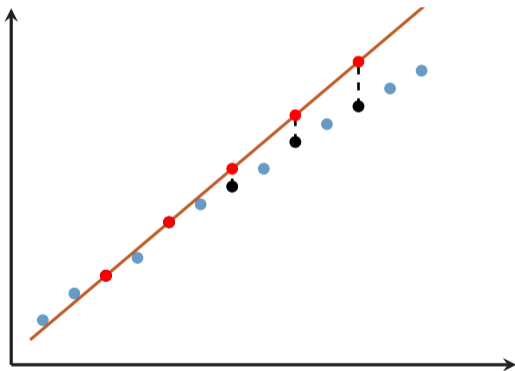
$$\hat{y} = \theta^T x + b$$

$\hat{y}^{(i)}$: predição para $x^{(i)}$

$y^{(i)}$: valor observado

Regressão linear

É usada quando os dados podem ser bem aproximados por uma relação linear:



Modelo linear

$$\hat{y} = \theta^T x + b$$

$\hat{y}^{(i)}$: predição para $x^{(i)}$

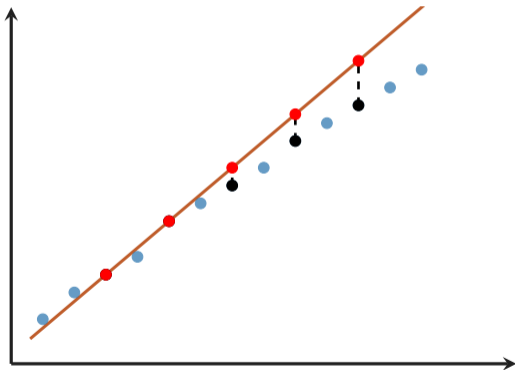
$y^{(i)}$: valor observado

Erro médio quadrático:

$$J(\theta, b) = \frac{1}{m} \sum_{i=1}^m (\hat{y}^{(i)} - y^{(i)})^2$$

Regressão linear

É usada quando os dados podem ser bem aproximados por uma relação linear:



Modelo linear

$$\hat{y} = \theta^T x + b$$

$\hat{y}^{(i)}$: predição para $x^{(i)}$

$y^{(i)}$: valor observado

Erro médio quadrático:

$$J(\theta, b) = \frac{1}{m} \sum_{i=1}^m (\hat{y}^{(i)} - y^{(i)})^2$$

Objetivo: minimizar $J(\theta, b)$.

Regressão linear regularizada

Para evitar ajustes excessivos aos dados e favorecer soluções com parâmetros de menor norma, adicionamos um termo de regularização em θ :

$$J(\theta, b) = \frac{1}{m} \sum_{i=1}^m (\theta^T x^{(i)} + b - y^{(i)})^2 + \lambda \|\theta\|^2$$

Regressão linear regularizada

Para evitar ajustes excessivos aos dados e favorecer soluções com parâmetros de menor norma, adicionamos um termo de regularização em θ :

$$J(\theta, b) = \frac{1}{m} \sum_{i=1}^m (\theta^T x^{(i)} + b - y^{(i)})^2 + \lambda \|\theta\|^2$$

- ▶ Minimizar $J(\theta, b)$;

Regressão linear regularizada

Para evitar ajustes excessivos aos dados e favorecer soluções com parâmetros de menor norma, adicionamos um termo de regularização em θ :

$$J(\theta, b) = \frac{1}{m} \sum_{i=1}^m (\theta^T x^{(i)} + b - y^{(i)})^2 + \lambda \|\theta\|^2$$

- ▶ Minimizar $J(\theta, b)$;
- ▶ Encontrar o gradiente $\nabla J(\theta, b)$;
- ▶ Resolver a equação $\nabla J(\theta, b) = 0$.

Regressão linear regularizada

Para evitar ajustes excessivos aos dados e favorecer soluções com parâmetros de menor norma, adicionamos um termo de regularização em θ :

$$J(\theta, b) = \frac{1}{m} \sum_{i=1}^m (\theta^T x^{(i)} + b - y^{(i)})^2 + \lambda \|\theta\|^2$$

- ▶ Minimizar $J(\theta, b)$;
- ▶ Encontrar o gradiente $\nabla J(\theta, b)$;
- ▶ Resolver a equação $\nabla J(\theta, b) = 0$.

Observação:

Como $J(\theta, b)$ é convexa, qualquer solução do sistema corresponde a um mínimo global.

Regressão linear regularizada

Função objetivo:

$$J(\theta, b) = \frac{1}{m} \|X\theta + b\mathbf{1} - Y\|^2 + \lambda \|\theta\|^2$$

$$\text{com } X = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \cdots & x_d^{(1)} \\ \vdots & \vdots & \cdots & \vdots \\ x_1^{(m)} & x_2^{(m)} & \cdots & x_d^{(m)} \end{bmatrix}, \quad Y = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(m)} \end{bmatrix} \quad \text{e} \quad \mathbf{1} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$

Regressão linear regularizada

Função objetivo:

$$J(\theta, b) = \frac{1}{m} \|X\theta + b\mathbf{1} - Y\|^2 + \lambda \|\theta\|^2$$

$$\text{com } X = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \cdots & x_d^{(1)} \\ \vdots & \vdots & \cdots & \vdots \\ x_1^{(m)} & x_2^{(m)} & \cdots & x_d^{(m)} \end{bmatrix}, \quad Y = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(m)} \end{bmatrix} \quad \text{e} \quad \mathbf{1} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$

Gradiente:

$$\nabla_{\theta} J = \frac{2}{m} X^T (X\theta + b\mathbf{1} - Y) + 2\lambda\theta, \quad \frac{\partial J}{\partial b} = \frac{2}{m} \mathbf{1}^T (X\theta + b\mathbf{1} - Y)$$

Regressão linear regularizada

Condição de otimalidade:

$$\nabla J(\theta, b) = 0$$

Regressão linear regularizada

Condição de otimalidade:

$$\nabla J(\theta, b) = 0$$

Sistema linear equivalente:

$$\begin{bmatrix} X^T X + m\lambda I & X^T \mathbf{1} \\ \mathbf{1}^T X & m \end{bmatrix} \begin{bmatrix} \theta \\ b \end{bmatrix} = \begin{bmatrix} X^T y \\ \mathbf{1}^T y \end{bmatrix}$$

Regressão linear regularizada

Condição de otimalidade:

$$\nabla J(\theta, b) = 0$$

Sistema linear equivalente:

$$\begin{bmatrix} X^T X + m\lambda I & X^T \mathbf{1} \\ \mathbf{1}^T X & m \end{bmatrix} \begin{bmatrix} \theta \\ b \end{bmatrix} = \begin{bmatrix} X^T y \\ \mathbf{1}^T y \end{bmatrix}$$

Observação:

A matriz do sistema tem ordem $(d + 1) \times (d + 1)$.

Limitações

Se d é muito grande, resolver o sistema diretamente não é prático.

Limitações

Se d é muito grande, resolver o sistema diretamente não é prático.

Precisamos resolver um sistema linear envolvendo a matriz

$$X^T X + m\lambda I \in \mathbb{R}^{d \times d}$$

que requer $O(d^3)$ operações aritméticas.

Limitações

Se d é muito grande, resolver o sistema diretamente não é prático.

Precisamos resolver um sistema linear envolvendo a matriz

$$X^T X + m\lambda I \in \mathbb{R}^{d \times d}$$

que requer $O(d^3)$ operações aritméticas.

Se $d \approx 10^5$, isso significa aproximadamente 10^{15} operações.

Limitações

Se d é muito grande, resolver o sistema diretamente não é prático.

Precisamos resolver um sistema linear envolvendo a matriz

$$X^T X + m\lambda I \in \mathbb{R}^{d \times d}$$

que requer $O(d^3)$ operações aritméticas.

Se $d \approx 10^5$, isso significa aproximadamente 10^{15} operações.

- ▶ Um computador realiza cerca de 10^8 operações por segundo

Limitações

Se d é muito grande, resolver o sistema diretamente não é prático.

Precisamos resolver um sistema linear envolvendo a matriz

$$X^T X + m\lambda I \in \mathbb{R}^{d \times d}$$

que requer $O(d^3)$ operações aritméticas.

Se $d \approx 10^5$, isso significa aproximadamente 10^{15} operações.

- ▶ Um computador realiza cerca de 10^8 operações por segundo
- ▶ Tempo total: 10^7 segundos

Limitações

Se d é muito grande, resolver o sistema diretamente não é prático.

Precisamos resolver um sistema linear envolvendo a matriz

$$X^T X + m\lambda I \in \mathbb{R}^{d \times d}$$

que requer $O(d^3)$ operações aritméticas.

Se $d \approx 10^5$, isso significa aproximadamente 10^{15} operações.

- ▶ Um computador realiza cerca de 10^8 operações por segundo
- ▶ Tempo total: 10^7 segundos
- ▶ Isso corresponde a aproximadamente 0,32 anos

Limitações

Se d é muito grande, resolver o sistema diretamente não é prático.

Precisamos resolver um sistema linear envolvendo a matriz

$$X^T X + m\lambda I \in \mathbb{R}^{d \times d}$$

que requer $O(d^3)$ operações aritméticas.

Se $d \approx 10^5$, isso significa aproximadamente 10^{15} operações.

- ▶ Um computador realiza cerca de 10^8 operações por segundo
- ▶ Tempo total: 10^7 segundos
- ▶ Isso corresponde a aproximadamente 0,32 anos

Conclusão: Precisamos de métodos mais eficientes.

Método do gradiente

Em vez de resolver o sistema linear diretamente, atualizamos (θ, b) de forma iterativa.

Método do gradiente

Em vez de resolver o sistema linear diretamente, atualizamos (θ, b) de forma iterativa.

Objetivo: Gerar uma sequência $\{(\theta_k, b_k)\}_{k \in \mathbb{N}}$ tal que $J(\theta_{k+1}, b_{k+1}) \leq J(\theta_k, b_k)$

Método do gradiente

Em vez de resolver o sistema linear diretamente, atualizamos (θ, b) de forma iterativa.

Objetivo: Gerar uma sequência $\{(\theta_k, b_k)\}_{k \in \mathbb{N}}$ tal que $J(\theta_{k+1}, b_{k+1}) \leq J(\theta_k, b_k)$

Iteração do método do gradiente:

- ▶ Escolher um ponto inicial (θ_0, b_0)

Método do gradiente

Em vez de resolver o sistema linear diretamente, atualizamos (θ, b) de forma iterativa.

Objetivo: Gerar uma sequência $\{(\theta_k, b_k)\}_{k \in \mathbb{N}}$ tal que $J(\theta_{k+1}, b_{k+1}) \leq J(\theta_k, b_k)$

Iteração do método do gradiente:

- ▶ Escolher um ponto inicial (θ_0, b_0)
- ▶ Atualizar: para $k = 0, 1, \dots$

$$\theta_{k+1} = \theta_k - \alpha_k \nabla_{\theta} J(\theta_k, b_k)$$

$$b_{k+1} = b_k - \alpha_k \frac{\partial J}{\partial b}(\theta_k, b_k)$$

Método do gradiente

Em vez de resolver o sistema linear diretamente, atualizamos (θ, b) de forma iterativa.

Objetivo: Gerar uma sequência $\{(\theta_k, b_k)\}_{k \in \mathbb{N}}$ tal que $J(\theta_{k+1}, b_{k+1}) \leq J(\theta_k, b_k)$

Iteração do método do gradiente:

- ▶ Escolher um ponto inicial (θ_0, b_0)
- ▶ Atualizar: para $k = 0, 1, \dots$

$$\theta_{k+1} = \theta_k - \alpha_k \nabla_{\theta} J(\theta_k, b_k)$$

$$b_{k+1} = b_k - \alpha_k \frac{\partial J}{\partial b}(\theta_k, b_k)$$

- ▶ $\alpha_k > 0$ é o tamanho do passo, que controla o quanto nos movemos a cada iteração.

Por que usar o método do gradiente?

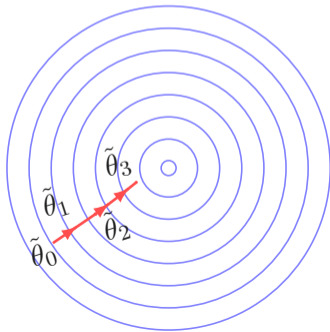
- ▶ Evita inverter matrizes (custo alto)
- ▶ Cada iteração é simples e barata
- ▶ Funciona bem em problemas grandes

Por que usar o método do gradiente?

- ▶ Evita inverter matrizes (custo alto)
- ▶ Cada iteração é simples e barata
- ▶ Funciona bem em problemas grandes

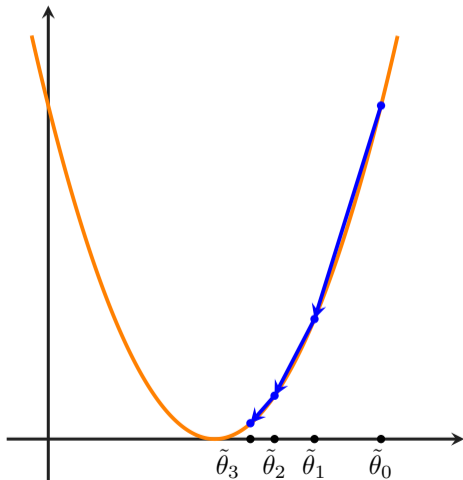
Interpretação:

- ▶ O gradiente aponta para onde a função cresce mais rápido
- ▶ Então andamos na direção oposta para minimizar



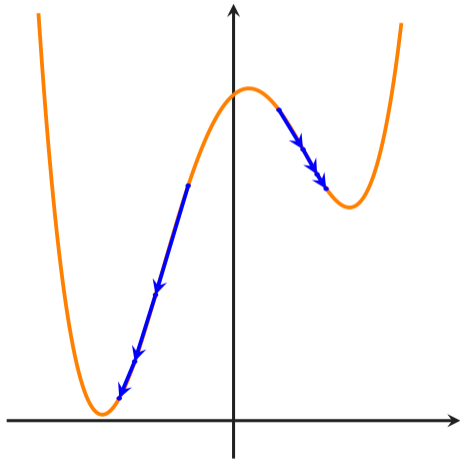
Convergência

Se J é uma função convexa, diferenciável e seu gradiente é L -Lipschitz contínuo (com $L > 0$), então a sequência gerada pelo método do gradiente com passo $\alpha_k = 1/L$, converge a um mínimo global.

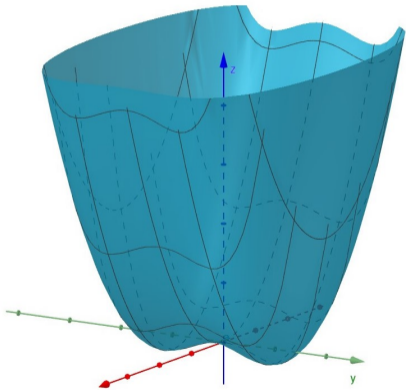


No caso geral

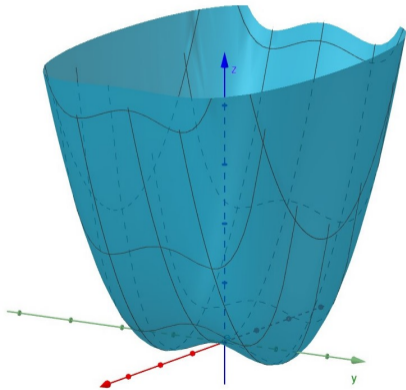
- ▶ Se J não é convexa, o método do gradiente pode não convergir para um mínimo global da função.



Exemplo: $f(x, y) = \frac{1}{2}x^2 + \frac{1}{4}y^4 - \frac{1}{2}y^2$



Exemplo: $f(x, y) = \frac{1}{2}x^2 + \frac{1}{4}y^4 - \frac{1}{2}y^2$

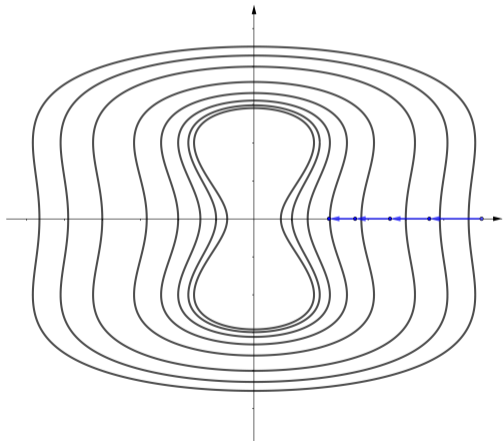


$$\theta = (x, y), \quad \nabla f(\theta) = (x, y^3 - y)$$

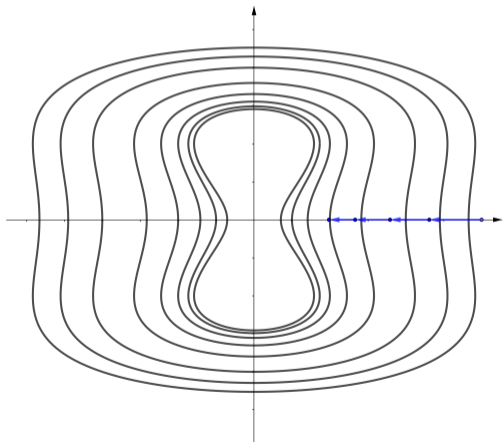
- ▶ $(0, 1)$ e $(0, -1)$ mínimos locais
- ▶ $(0, 0)$ ponto sela

Exemplo: $f(x, y) = \frac{1}{2}x^2 + \frac{1}{4}y^4 - \frac{1}{2}y^2$

Ponto inicial: $\theta_0 = (1, 0)$



Exemplo: $f(x, y) = \frac{1}{2}x^2 + \frac{1}{4}y^4 - \frac{1}{2}y^2$



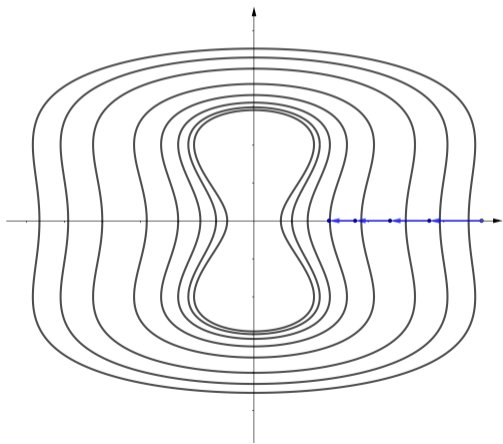
Ponto inicial: $\theta_0 = (1, 0)$

$$\nabla f(\theta_0) = (1, 0)$$

$$\theta_1 = (1, 0) - \alpha_k(1, 0)$$

$$(\theta_1)_2 = 0$$

Exemplo: $f(x, y) = \frac{1}{2}x^2 + \frac{1}{4}y^4 - \frac{1}{2}y^2$



Ponto inicial: $\theta_0 = (1, 0)$

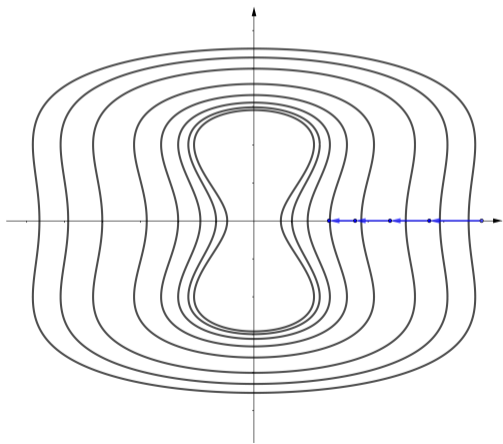
$$\nabla f(\theta_0) = (1, 0)$$

$$\theta_1 = (1, 0) - \alpha_k(1, 0)$$

$$(\theta_1)_2 = 0$$

Então $(\theta_k)_2 = 0$ para todo $k = 1, 2, \dots$

Exemplo: $f(x, y) = \frac{1}{2}x^2 + \frac{1}{4}y^4 - \frac{1}{2}y^2$



Ponto inicial: $\theta_0 = (1, 0)$

$$\nabla f(\theta_0) = (1, 0)$$

$$\theta_1 = (1, 0) - \alpha_k(1, 0)$$

$$(\theta_1)_2 = 0$$

Então $(\theta_k)_2 = 0$ para todo $k = 1, 2, \dots$
O método converge para o ponto $(0, 0)$
que é um ponto sela.

Redes neurais

Limitações da regressão linear

Na regressão linear, modelamos:

$$\hat{y} = X\theta + b$$

Limitações da regressão linear

Na regressão linear, modelamos:

$$\hat{y} = X\theta + b$$

Problema:

- ▶ O modelo é linear
- ▶ Não captura relações mais complexas nos dados

Exemplo: Dados que não podem ser separados por uma reta.

Modelo de rede neural

Uma alternativa para construir modelos mais expressivos são as **redes neurais**.

Modelo de rede neural

Uma alternativa para construir modelos mais expressivos são as **redes neurais**.

Uma rede neural aplica aplica **várias camadas sucessivas** aos dados de entrada.

Modelo de rede neural

Uma alternativa para construir modelos mais expressivos são as **redes neurais**.

Uma rede neural aplica aplica **várias camadas sucessivas** aos dados de entrada.

Fluxo da informação:

$$x \longrightarrow \text{camada 1} \longrightarrow \text{camada 2} \longrightarrow \dots \longrightarrow \hat{y}$$

Modelo de rede neural

Uma alternativa para construir modelos mais expressivos são as **redes neurais**.

Uma rede neural aplica aplica **várias camadas sucessivas** aos dados de entrada.

Fluxo da informação:

$$x \longrightarrow \text{camada 1} \longrightarrow \text{camada 2} \longrightarrow \dots \longrightarrow \hat{y}$$

Cada camada tem a forma:

$$a = \sigma(Wx + b)$$

onde σ é uma função não linear (ativação).

Estrutura de uma camada

- ▶ Entrada:

$$a^{(l-1)}$$

- ▶ Passo 1 (linear):

$$z^{(l)} = W^{(l)}a^{(l-1)} + b^{(l)}$$

- ▶ Passo 2 (não linear):

$$a^{(l)} = \sigma(z^{(l)})$$

Estrutura de uma camada

- ▶ Entrada:

$$a^{(l-1)}$$

- ▶ Passo 1 (linear):

$$z^{(l)} = W^{(l)}a^{(l-1)} + b^{(l)}$$

- ▶ Passo 2 (não linear):

$$a^{(l)} = \sigma(z^{(l)})$$

Ideia:

- ▶ Transformação linear + não linear
- ▶ A saída de uma camada é a entrada da próxima

$$x \rightarrow z^{(1)} \rightarrow a^{(1)} \rightarrow z^{(2)} \rightarrow a^{(2)} \rightarrow \dots \rightarrow \hat{y}$$

Estrutura de uma camada

- ▶ Entrada:

$$a^{(l-1)}$$

- ▶ Passo 1 (linear):

$$z^{(l)} = W^{(l)}a^{(l-1)} + b^{(l)}$$

- ▶ Passo 2 (não linear):

$$a^{(l)} = \sigma(z^{(l)})$$

Ideia:

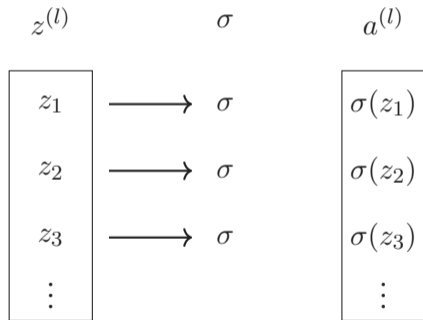
- ▶ Transformação linear + não linear
- ▶ A saída de uma camada é a entrada da próxima

$$x \rightarrow z^{(1)} \rightarrow a^{(1)} \rightarrow z^{(2)} \rightarrow a^{(2)} \rightarrow \dots \rightarrow \hat{y}$$

Observação: Se não tivesse σ , seria regressão linear empilhada.

A função ativação

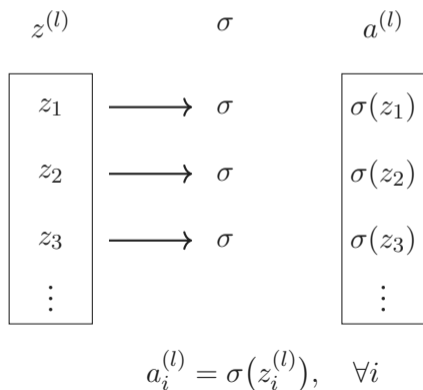
A função $\sigma : \mathbb{R} \rightarrow \mathbb{R}$ é aplicada separadamente a cada componente.



$$a_i^{(l)} = \sigma(z_i^{(l)}), \quad \forall i$$

A função ativação

A função $\sigma : \mathbb{R} \rightarrow \mathbb{R}$ é aplicada separadamente a cada componente.



Exemplos de σ :

ReLU:

$$\sigma(z) = \max(0, z)$$

Sigmoid:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Tanh:

$$\sigma(z) = \tanh(z)$$

Problema de otimização em redes neurais

Dado $\{(x_i, y_i)\}$, queremos minimizar:

$$J(W, b) = \frac{1}{n} \sum_{i=1}^n \ell(\hat{y}_i, y_i)$$

onde:

- ▶ ℓ : função de erro
- ▶ \hat{y}_i = saída da rede para x_i

Problema de otimização em redes neurais

Dado $\{(x_i, y_i)\}$, queremos minimizar:

$$J(W, b) = \frac{1}{n} \sum_{i=1}^n \ell(\hat{y}_i, y_i)$$

onde:

- ▶ ℓ : função de erro
- ▶ \hat{y}_i = saída da rede para x_i

Parâmetros:

- ▶ W : matrizes de pesos
- ▶ b : vetores de bias

Função erro total

A função J mede o erro total da rede entre a previsão \hat{y} e o valor real y :

$$J(W, b) = \frac{1}{n} \sum_{i=1}^n \ell(\hat{y}_i, y_i)$$

Função erro total

A função J mede o erro total da rede entre a previsão \hat{y} e o valor real y :

$$J(W, b) = \frac{1}{n} \sum_{i=1}^n \ell(\hat{y}_i, y_i)$$

Exemplos

- ▶ Regressão:

$$\ell(\hat{y}, y) = (\hat{y} - y)^2$$

- ▶ Classificação binária:

$$\ell(\hat{y}, y) = -[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})]$$

Treinando uma rede

Queremos minimizar a função de erro total:

$$\min_{W,b} J(W, b)$$

Treinando uma rede

Queremos minimizar a função de erro total:

$$\min_{W,b} J(W, b)$$

Para minimizar $J(W, b)$, usamos o método do gradiente:

$$(W_{k+1}, b_{k+1}) = (W_k, b_k) - \alpha \nabla J(W_k, b_k)$$

Treinando uma rede

Queremos minimizar a função de erro total:

$$\min_{W,b} J(W, b)$$

Para minimizar $J(W, b)$, usamos o método do gradiente:

$$(W_{k+1}, b_{k+1}) = (W_k, b_k) - \alpha \nabla J(W_k, b_k)$$

Ideia: ajustar os parâmetros na direção de maior redução do erro.

Redes neurais não são convexas

Em geral, o problema de treinamento de redes neurais é **não convexo**

Redes neurais não são convexas

Em geral, o problema de treinamento de redes neurais é **não convexo**

Consequências:

- ▶ Podem existir vários mínimos locais
- ▶ O resultado depende da inicialização
- ▶ Não há garantia de encontrar o mínimo global

Redes neurais não são convexas

Em geral, o problema de treinamento de redes neurais é **não convexo**

Consequências:

- ▶ Podem existir vários mínimos locais
- ▶ O resultado depende da inicialização
- ▶ Não há garantia de encontrar o mínimo global

Mesmo assim: métodos de gradiente funcionam muito bem na prática.

Método do gradiente

A cada iteração:

- ▶ Calculamos o gradiente de J
- ▶ Atualizamos os parâmetros

$$(W_{k+1}, b_{k+1}) = (W_k, b_k) - \alpha \nabla J(W_k, b_k)$$

Método do gradiente

A cada iteração:

- ▶ Calculamos o gradiente de J
- ▶ Atualizamos os parâmetros

$$(W_{k+1}, b_{k+1}) = (W_k, b_k) - \alpha \nabla J(W_k, b_k)$$

Notação alternativa:

$$W^{(l)} \leftarrow W^{(l)} - \alpha \frac{\partial J}{\partial W^{(l)}}$$

$$b^{(l)} \leftarrow b^{(l)} - \alpha \frac{\partial J}{\partial b^{(l)}}$$

Método do gradiente

A cada iteração:

- ▶ Calculamos o gradiente de J
- ▶ Atualizamos os parâmetros

$$(W_{k+1}, b_{k+1}) = (W_k, b_k) - \alpha \nabla J(W_k, b_k)$$

Notação alternativa:

$$W^{(l)} \leftarrow W^{(l)} - \alpha \frac{\partial J}{\partial W^{(l)}}$$

$$b^{(l)} \leftarrow b^{(l)} - \alpha \frac{\partial J}{\partial b^{(l)}}$$

Objetivo: Repetimos esse processo até reduzir o erro.

Como calcular o gradiente?

Para aplicar o método do gradiente, precisamos de:

$$\nabla J(W, b)$$

Como calcular o gradiente?

Para aplicar o método do gradiente, precisamos de:

$$\nabla J(W, b)$$

Backpropagation:

- ▶ Calcula os gradientes de forma eficiente

Como calcular o gradiente?

Para aplicar o método do gradiente, precisamos de:

$$\nabla J(W, b)$$

Backpropagation:

- ▶ Calcula os gradientes de forma eficiente
- ▶ A rede é uma composição de funções:

$$a^{(0)} = x \rightarrow a^{(1)} \rightarrow \dots \rightarrow a^{(L)} = \hat{y}$$

Usa a **regra da cadeia**.

Ideia do backpropagation

Para calcular o gradiente de J em relação aos parâmetros, calculamos o erro na saída:

$$J \rightarrow a^{(L)} \rightarrow z^{(L)} \rightarrow \dots \rightarrow W^{(l)}$$

Ideia do backpropagation

Para calcular o gradiente de J em relação aos parâmetros, calculamos o erro na saída:

$$J \rightarrow a^{(L)} \rightarrow z^{(L)} \rightarrow \dots \rightarrow W^{(l)}$$

E vamos voltando camada por camada:

$$\frac{\partial J}{\partial W^{(l)}} = \frac{\partial J}{\partial a^{(L)}} \cdot \frac{\partial a^{(L)}}{\partial z^{(L)}} \cdots \frac{\partial z^{(l)}}{\partial W^{(l)}}$$

Ideia do backpropagation

Para calcular o gradiente de J em relação aos parâmetros, calculamos o erro na saída:

$$J \rightarrow a^{(L)} \rightarrow z^{(L)} \rightarrow \dots \rightarrow W^{(l)}$$

E vamos voltando camada por camada:

$$\frac{\partial J}{\partial W^{(l)}} = \frac{\partial J}{\partial a^{(L)}} \cdot \frac{\partial a^{(L)}}{\partial z^{(L)}} \cdots \frac{\partial z^{(l)}}{\partial W^{(l)}}$$

Gradiente em relação aos pesos

Se $W^{(l)}$ é uma matriz, então: $\frac{\partial J}{\partial W^{(l)}}$ também é uma matriz.

Cada entrada corresponde a: $\frac{\partial J}{\partial w_{ij}}$

Na prática

Para treinar uma rede são usadas variantes do método do gradiente.

Na prática

Para treinar uma rede são usadas variantes do método do gradiente.

Exemplos

- ▶ Gradiente descendente (mini-batch)
- ▶ Gradiente estocástico (SGD)
- ▶ Adam (muito usado)

Na prática

Para treinar uma rede são usadas variantes do método do gradiente.

Exemplos

- ▶ Gradiente descendente (mini-batch)
- ▶ Gradiente estocástico (SGD)
- ▶ Adam (muito usado)

Todos seguem a mesma ideia:

$$\text{novo} = \text{antigo} - \text{passo} \times \text{gradiente}$$