Inteligência Artificial

2024/2

Profa. Dra. Juliana Félix

jufelix16@gmail.com

PUC GOIÁS

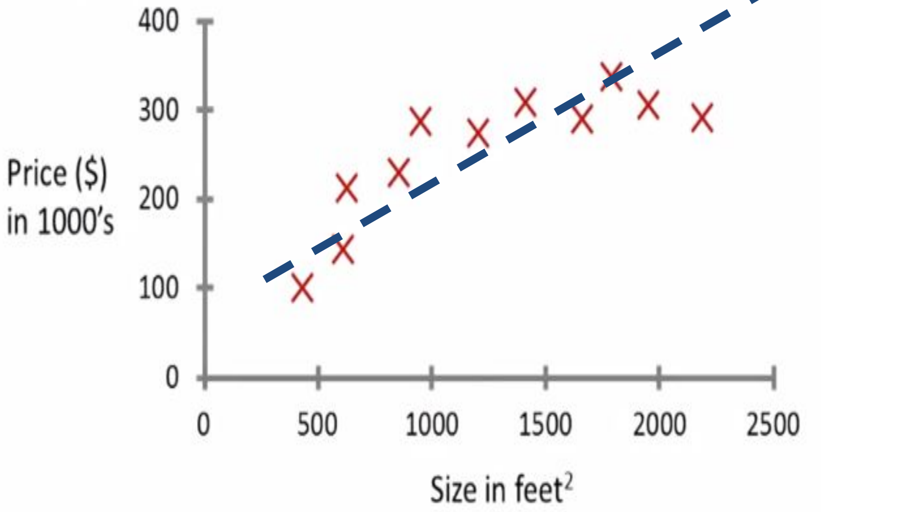
Regressão Linear

Regressão Linear

Modelo

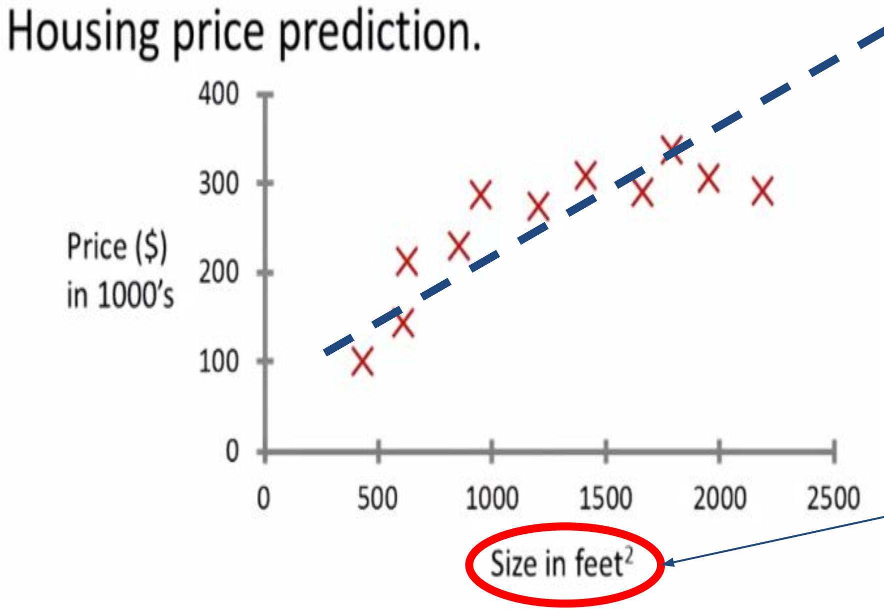
$$\boxed{y = \theta_0 + \theta_1 x}$$

Housing price prediction.



Regressão Linear

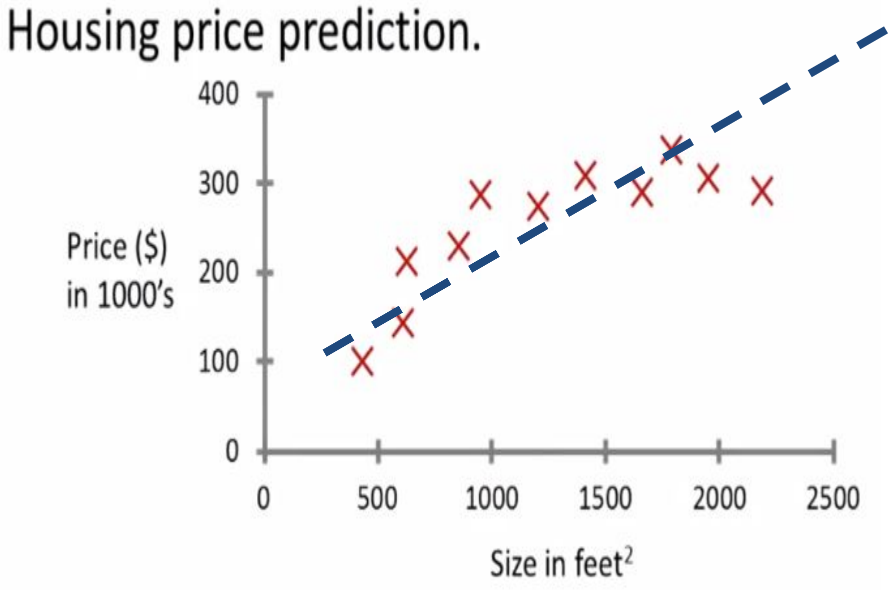
$$y = \theta_0 + \theta_1 x$$



Características
(*features*)

Regressão Linear

$$y = \theta_0 + \theta_1 x$$



Parâmetros

Regressão Linear

Saída (output)

$$y = \theta_0 + \theta_1 x$$

Exemplo

Conjunto de dados (*Dataset*)

Um ***Dataset*** (conjunto de dados) combina amostras com

- Valores ou **variáveis de entrada** (features, características, recursos) e
- Valores de **saída** (outcome, labels) utilizados no aprendizado supervisionado.

Dataset

Exemplo:

Preço de imóveis na 'Terra tão tão distante'	
Tamanho do imóvel (em m^2) - X	**Preço (R$) - Y**
2104	399.900
1600	329.900
2400	369.000
...	...

Total amostras = 47

Feature (característica)

Outcome (saída)

Notação

Podemos pensar no problema anterior como um problema que tem:

- Um total de $\mathbf{m}$ amostras/*samples* ($\mathbf{m}$ = 47)
- Cada amostra tem 1 única *feature*/característica (tamanho do imóvel)
    - Costumamos representar uma variável de entrada por $\mathbf{x}$
- Para cada amostra, temos uma única saída (preço do imóvel).
    - Costumamos representar uma variável de saída por $\mathbf{y}$
- Cada amostra pode ser representada por um par, ou tupla $\mathbf{(x,y)}$
    - Uma tupla $\mathbf{(x^i, y^i)}$ representa a i-ésima amostra do problema, com $\mathbf{1 \leq i \leq m}$

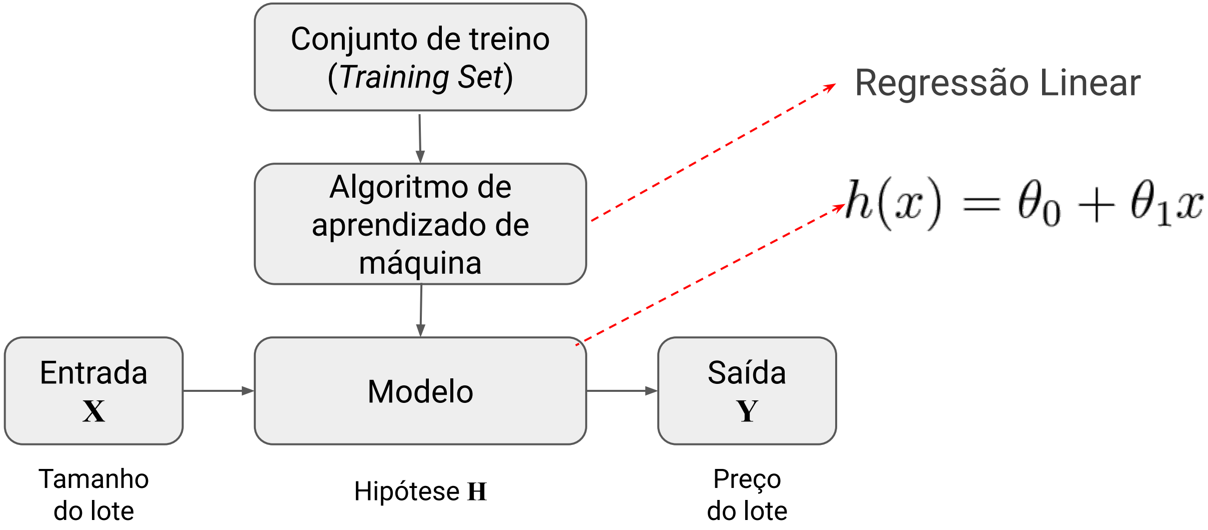
Notação

- No conjunto abaixo, o par (x^1, y^1) refere-se aos dados da primeira linha $(2104, 399900)$.

Preço de lotes na 'Terra tão tão distante'	
Tamanho do lote (em m^2) - X	**Preço do lote (R$) - Y**
2104	399.900
1600	329.900
2400	369.000
...	...

Processo básico de Machine Learning

A base de qualquer processo de machine learning consiste em mapear um dado de entrada $\mathbf{X}$ em um dado de saída $\mathbf{Y}$.



Desvio

Quando fazemos a predição de um valor, o **desvio** é a diferença entre o **valor esperado** (conhecido) e o **valor predito** pelo modelo construído.

$$\text{desvio}^{i} = \mathbf{Y^{i} - h(x^{i})}$$

$$\text{desvio}^{i} = \mathbf{Y^{i} - \hat{Y}^{i}}$$

MSE

O ***Mean Square Error*** (MSE - Erro Médio Quadrático) é a **média** do **quadrado** dos **erros** obtidos pelo modelo.

$$MSE = \frac{1}{m}\sum_{i=1}^{m}\left(Y^i - h(x^i)\right)^2$$

MSE

O ***Mean Square Error*** (MSE - Erro Médio Quadrático) é a **média** do **quadrado** dos **erros** obtidos pelo modelo.

$$MSE = \frac{1}{2m}\sum_{i=1}^{m}\left(Y^i - h(x^i)\right)^2$$

Na prática, a média é dividida pela metade (1/2) como uma conveniência para o cálculo do ***gradiente descendente***, método utilizado na regressão linear, que cancelará o termo 1/2.

Em outras palavras, dividir por m ou 2m não traz diferenças significativas para o cálculo dos valores analisados.

Leia mais aqui: https://math.stackexchange.com/questions/884887/why-divide-by-2m

MSE

$$MSE = \frac{1}{2m} \sum_{i=1}^{m} \left(Y^i - h(x^i)\right)^2$$

Considerando os seguintes valores preditos, podemos calcular o *MSE* do modelo.

Preço de lotes na 'Terra tão tão distante'				
x	y	h(x)	desvio	desvio^2
2104	399.900	399.800		
1600	329.900	339.900		
2400	369.000	367.000		
			Soma	
			MSE	

MSE

$$MSE = \frac{1}{2m}\sum_{i=1}^{m}\left(Y^i - h(x^i)\right)^2$$

Considerando os seguintes valores preditos, podemos calcular o *MSE* do modelo.

Preço de lotes na 'Terra tão tão distante'				
x	y	h(x)	desvio	desvio^2
2104	399.900	399.800	100	10.000
1600	329.900	339.900	-10.000	10.000.000
2400	369.000	367.000	2.000	4.000.000
			Soma	14.010.000
			MSE	**2.335.000**

Exercício

Considerando os valores x e y fornecidos, tente encontrar, manualmente, uma reta que melhor se ajuste aos dados abaixo:

```
x = np.array([480, 510, 520, 850, 960, 1200, 1400, 1650, 1700, 1920, 2350])
y = np.array([98, 110, 200, 210, 280, 265, 300, 287, 325, 300, 290])
```

Faça isso utilizando:

a) Apenas θ_0
b) Apenas θ_1
c) Atribuindo valores para θ_0 e θ_1

Para cada reta, calcule o respectivo MSE. Plote as retas e o MSE encontrado em todos os casos.

Função Custo

Função Custo

Queremos encontrar $\boldsymbol{\theta}_0$ e $\boldsymbol{\theta}_1$ para os quais os valores preditos serão os mais próximos possíveis dos valores $\mathbf{y}$ reais.

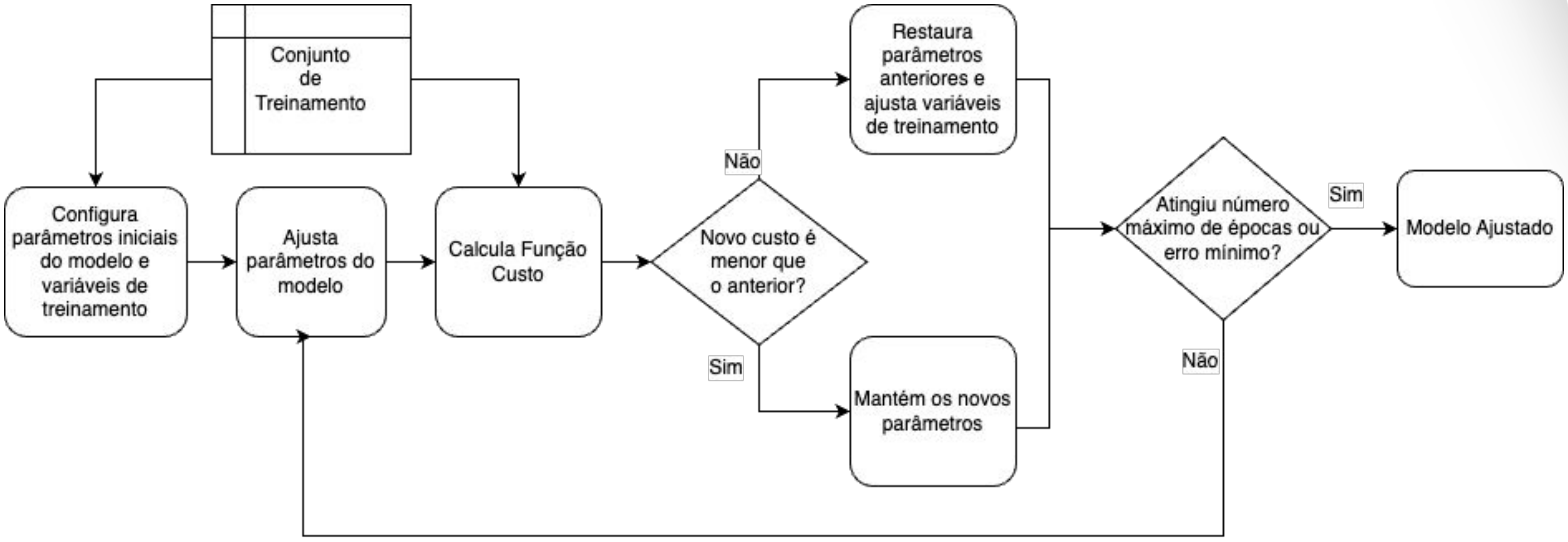
MSE

Objetivo

$$\min_{\theta_0, \theta_1} J(\theta_0, \theta_1) = \left[\frac{1}{2m} \sum_{i=1}^{m} \left(Y^i - h(x^i)\right)^2 \right]$$

$$h(x) = \theta_0 + \theta_1 x$$

Treinamento do Modelo



Conceitos importantes

- **Parâmetros** - As variáveis do modelo a serem ajustadas
- **Dados de treinamento** - Grupo de dados utilizado para ajustar os parâmetros do modelo.
- **Variáveis de treinamento** - variáveis utilizadas para controlar o algoritmo de treinamento.
  - Step/Passo - tamanho do ajuste que será utilizado para ajustar um parâmetro do modelo.
  - Época - O intervalo durante o qual o modelo é ajustado baseado no dado de treinamento.
  - Épocas - O número de épocas pelo qual o algoritmo de treinamento funcionará.
- **Parâmetros anteriores e atuais** - valores que representam o modelo antes e depois do ajuste.
- **Custo anterior e atual** - valores que representam o erro do modelo antes e depois do ajuste.
- **Parâmetros anteriores e atuais** - valores que representam o modelo antes e depois do ajuste.
- **Fator de ajuste** - valor que ajuda a determinar o comportamento do algoritmo de treinamento.

PUC
GOIÁS

Critérios de parada

- Número de épocas
  - O programa para ao atingir o número de épocas definido
- Erro mínimo atingido
  - O programa para ao atingir um erro menor ou igual ao erro mínimo definido
- Sem alterações após **n** épocas
  - O programa para ao perceber que o erro não sofreu alterações após uma quantidade **n** de épocas estabelecidas

Função Custo

Para um problema com apenas 1 característica em cada amostra, o problema consiste em encontrar os valores de $\boldsymbol{\theta}_0$ e $\boldsymbol{\theta}_1$ que minimizam o erro entre os valores preditos e os esperados.

parâmetros da reta

Objetivo

$$\min_{\theta_0, \theta_1} J(\theta_0, \theta_1) = \left[\frac{1}{2m} \sum_{i=1}^{m} \left(Y^i - h(x^i) \right)^2 \right]$$

$$h(x) = \theta_0 + \theta_1 x$$

Características da Função Custo

- A função custo é o resultado da soma do quadrado dos desvios (*Mean Square Error*).
- Os desvios são obtidos pela diferença entre Y^i e $h(x^i)$.

$$\min_{\theta_0, \theta_1} J(\theta_0, \theta_1) = \left[\frac{1}{2m} \sum_{i=1}^{m} \left(Y^i - h(x^i) \right)^2 \right]$$

Características da Função Custo

- Se simplificarmos o problema e considerarmos apenas θ_0 (ou seja, se quisermos encontrar apenas uma reta paralela ao eixo x, sem inclinação), o problema se resume a:

$$J(\theta_0) = \left[\frac{1}{2m} \sum_{i=1}^{m} \left(Y^i - h(x^i) \right)^2 \right]$$

$$(Y - h(x))^2$$

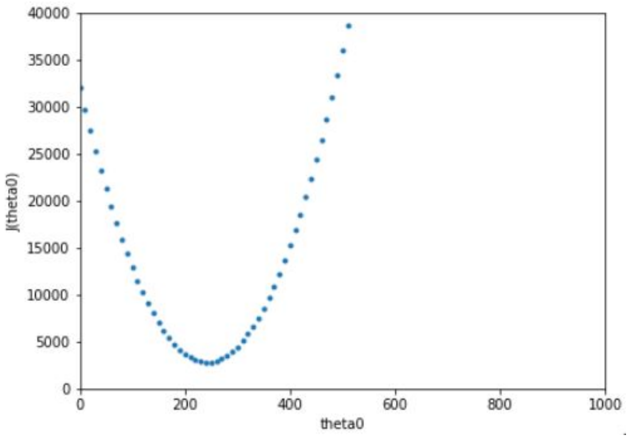
$$Y^2 - 2 \cdot Y \cdot h(x) + h(x)^2$$

Equação do 2º grau, uma parábola!

Características da Função Custo

- Sabemos que $h(x) = \theta_1 x + \theta_0$
- Simplificando $h(x) = \theta_0$, teremos uma parábola:

```
Cost:    29741.954545454544
Theta 0: 10
Epoch:   1
```



$$J(\theta_0) = Y^2 - 2 \cdot Y \cdot h(x) + h(x)^2$$

Exercício

Considerando os valores x e y fornecidos, crie um algoritmo para encontrar, automaticamente, uma reta que melhor se ajuste aos dados abaixo:

```
x = np.array([480, 510, 520, 850, 960, 1200, 1400, 1650, 1700, 1920, 2350])
y = np.array([98, 110, 200, 210, 280, 265, 300, 287, 325, 300, 290])
```

Minimiza o MSE

Faça isso utilizando:

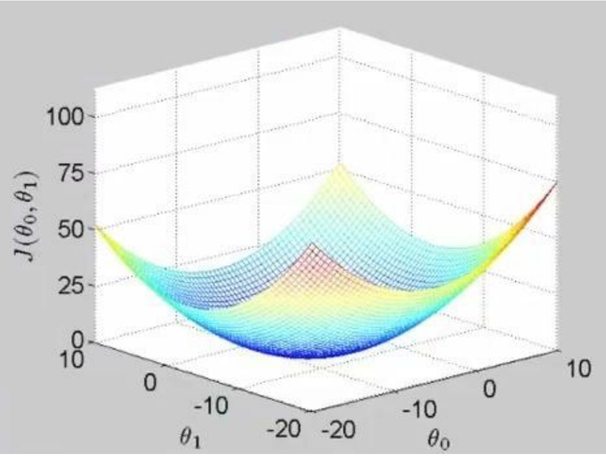
- Apenas θ_0
- Apenas θ_1

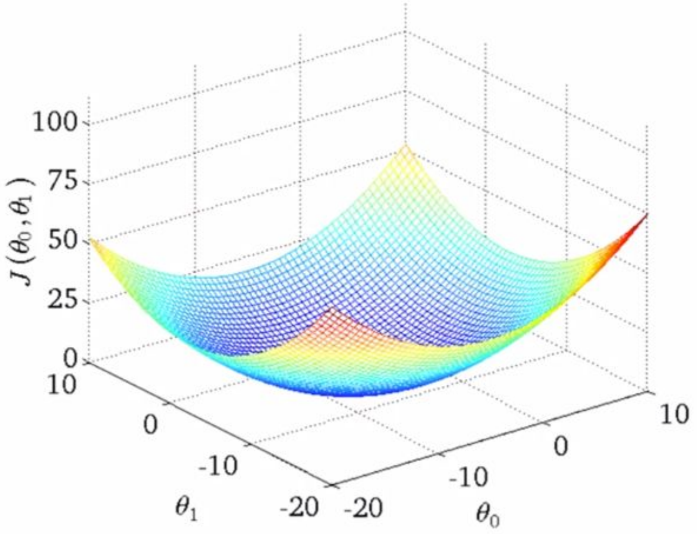
Para cada reta, calcule o respectivo MSE. Plote as retas e o MSE encontrado em todos os casos.

Gradiente Descendente

Características da Função Custo

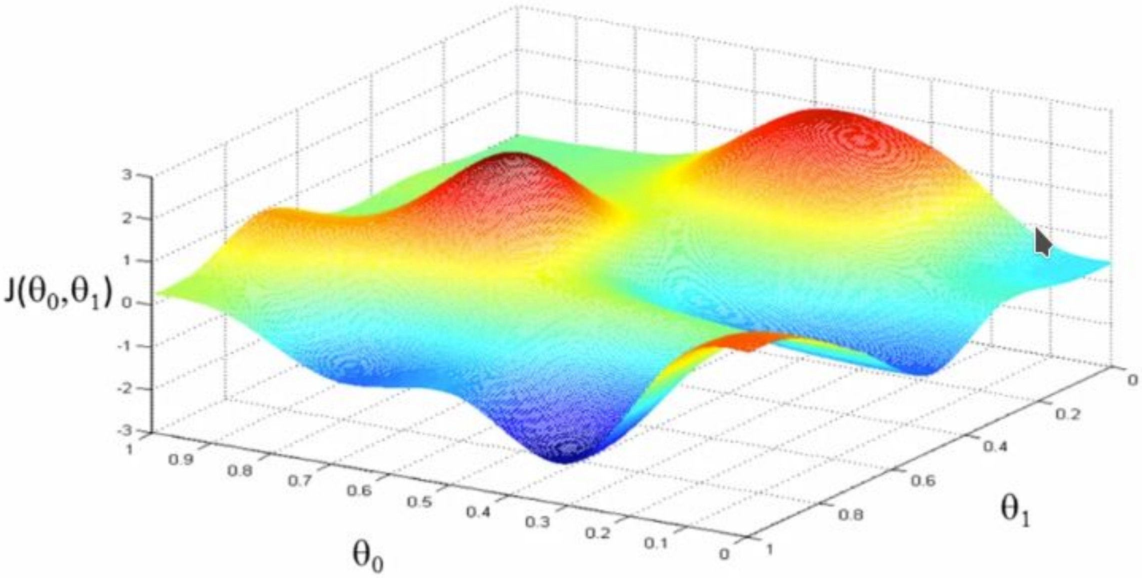
- Sabemos que $h(x) = \theta_1 x + \theta_0$
- Ao considerar θ_0 e θ_1, o gráfico da função custo nos trará um espaço 3D.





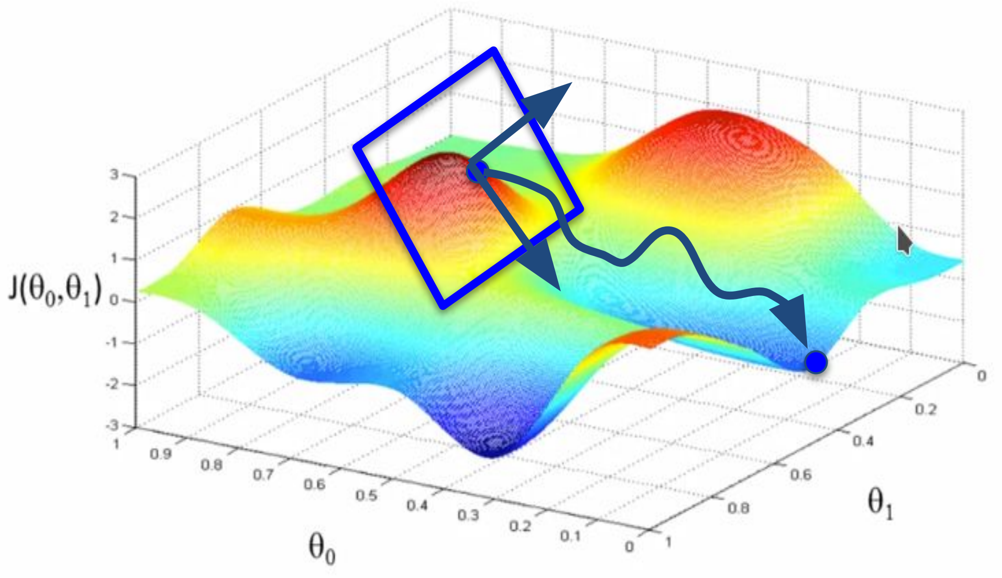
Características da Função Custo

Em certos modelos não lineares, mesmo tendo apenas θ_0 e θ_1, podemos nos deparar com superfícies cada vez mais complexas...



Características da Função Custo

Precisamos de uma maneira para encontrar o mínimo da função correspondente



$$\partial\left[J(\theta_0,\theta_1)\right] = \begin{bmatrix} \frac{\partial}{\partial\theta_0} J(\theta_0,\theta_1) \\ \\ \frac{\partial}{\partial\theta_1} J(\theta_0,\theta_1) \end{bmatrix}$$

Gradiente Descendente

- Algoritmo utilizado para encontrar os valores que minimizam a função custo.
- O algoritmo funciona iterativamente, ajustando os parâmetros em pequenos incrementos para minimizar a função de custo.
- O primeiro passo é calcular o gradiente da função de custo em relação a cada parâmetro.

$$new\theta_0 = \theta_0 - \alpha \cdot \boxed{\frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)}$$

$$new\theta_1 = \theta_1 - \alpha \cdot \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$

PUC GOIÁS

Gradiente Descendente

$$J(\theta_0, \theta_1) = \left[\frac{1}{2m} \sum_{i=1}^{m} \left(Y^i - h(x^i)\right)^2 \right]$$

Calculando-se as derivadas parciais, temos...

$$h(x) = \theta_0 + \theta_1 x$$

$$\frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} [(Y^i - \theta_0 - \theta_1 x^i)]$$

$$\frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} [(Y^i - \theta_0 - \theta_1 x^i) x^i]$$

Gradiente Descendente

$$new\theta_0 = \theta_0 - \alpha \cdot \boxed{\frac{\partial}{\partial\theta_0} J(\theta_0, \theta_1)}$$

$$new\theta_o = \theta_0 - \alpha \cdot \left[\frac{1}{m} \sum_{i=1}^{m} [(Y^i - h(x^i))] \right]$$

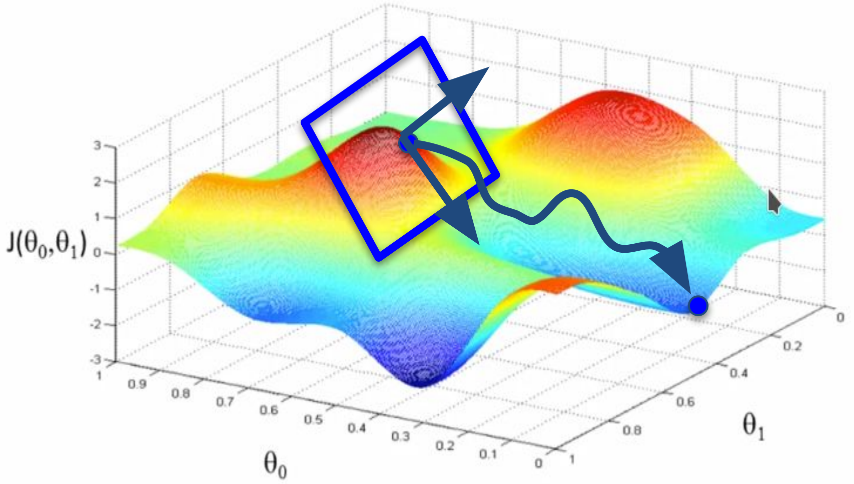
Substituindo...

$$new\theta_1 = \theta_1 - \alpha \cdot \boxed{\frac{\partial}{\partial\theta_1} J(\theta_0, \theta_1)}$$

$$new\theta_1 = \theta_1 - \alpha \cdot \left[\frac{1}{m} \sum_{i=1}^{m} [(Y^i - h(x^i) \cdot x^i)] \right]$$

Gradiente Descendente

- Em seguida, os parâmetros são atualizados movendo-se na direção oposta ao gradiente.
- Esse processo é repetido várias vezes até que uma condição de parada seja atingida.



$$new\theta_o = \theta_0 - \alpha \cdot \left[\frac{1}{m}\sum_{i=1}^{m}[(Y^i - h(x^i))]\right]$$

$$new\theta_1 = \theta_1 - \alpha \cdot \left[\frac{1}{m}\sum_{i=1}^{m}[(Y^i - h(x^i) \cdot x^i)]\right]$$

- Alfa (α) é a **taxa de aprendizagem**

Aula de hoje...

- Falamos sobre o funcionamento da regressão linear
  - Entendemos o que é um **modelo** de RL
  - Como funciona o treinamento de um modelo de RL
  - Como o cálculo do erro é realizado
  - Como minimizar o erro (função custo)
  - Apresentamos o algoritmo do gradiente descendente para minimização da função custo.

Atividade 2

- A atividade pode ser feita em duplas.
- Sua solução deve ser entregue em um notebook python (.ipynb).
- Reinicie o ambiente de execução e rode todas as células antes de gerar o arquivo a ser entregue (garantindo a exibição de todos os gráficos, por ex.)
- A entrega deve ser realizada até 23h59 de segunda-feira, 16/09/2024.
- O ambiente de entrega ainda será definido (estou aguardando acesso aos sistemas da universidade) e a turma será avisada.

PUC GOIÁS

Leitura recomendada

Calculadora gráfica: Desmos | Calculadora Gráfica

Regressão linear: Explicação sobre o modelo de regressão linear