

Regressão linear - Exercícios

Universidade Federal de Santa Catarina

Departamento de Automação e Sistemas - 2020/01

Prof. Eric Aislan Antonelo

1 Exercícios

- (a) Implemente em Python ou em sua linguagem preferida o algoritmo do descenso do gradiente para um modelo linear de uma variável:

$$h_{\theta}(x) = \theta_0 + \theta_1 x$$

Implemente as funções apresentadas abaixo. Escolha um valor inicial para θ . Defina a taxa de aprendizagem. Use o procedimento **print_modelo** para verificar o comportamento da reta definida por θ ao longo do processo de treinamento. Ela tende a se aproximar da reta original definida em **f_true**?

```
1 import numpy as np
2
3 def f_true(x):
4     return 2 + 0.8 * x
5
6 # conjunto de dados {(x,y)}
7 xs = np.linspace(-3, 3, 100)
8 ys = np.array( [f_true(x) + np.random.randn()*0.5 for x in xs] )
9
10
11 ''' hipotese
12 '''
13 def h(x, theta):
14     pass
15
16
17 ''' funcao de custo
18 '''
19 def J(theta, xs, ys):
20     pass
21
22
23 ''' derivada parcial com respeito a theta[i]
24 '''
25 def gradient(i, theta, xs, ys):
26     pass
27
28
29 ''' plota no mesmo grafico: - o modelo/hipotese (reta)
30     - a reta original (true function)
31     - e os dados com ruido (xs, ys)
32 '''
33 def print_modelo(theta, xs, ys):
34     pass
```

- (b) Plote a função de custo ao longo das iterações (épocas) para diferentes valores da taxa de aprendizagem (0.9, 0.1, 0.0001) por 5000 épocas de treinamento.

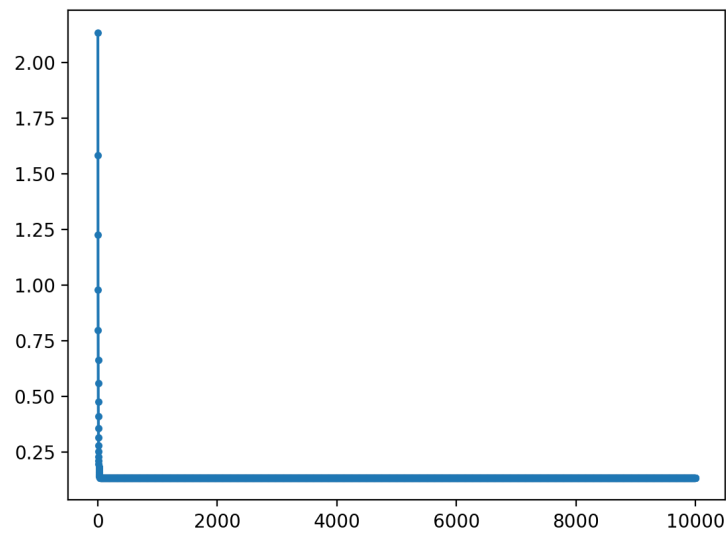


Figure 1: Função de custo para Taxa de aprendizagem = 0.1 por 10 mil épocas.

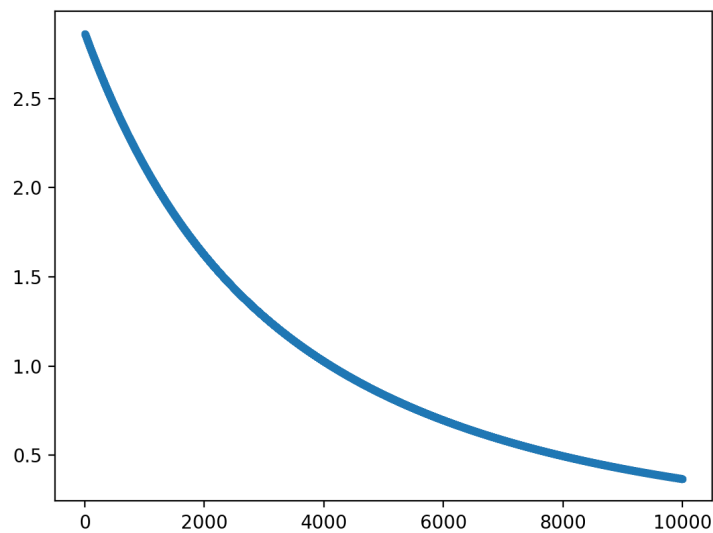


Figure 2: Função de custo para Taxa de aprendizagem = 0.0001 por 10 mil épocas.

- (c) (opcional) Plote a função de custo em função dos parâmetros θ_0 e θ_1 .
- (d) (opcional) Modifique a função original para ser algum tipo de curva, e tente usar regressão polinomial para aproximá-la.
- (e) (opcional) Adicione uma variável a mais e adapte o código para regressão multi-variável.