

## Projeto final MC970

Rian Radeck - 187793

Igor Brito - 171929

Cirilo Moraes - 168838

2 de Julho de 2023

## Contents

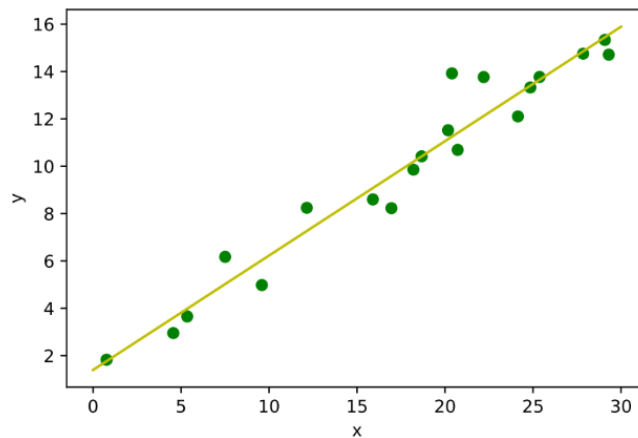
<b>1</b>	<b>Introdução</b>	<b>3</b>
<b>2</b>	<b>Regressão linear</b>	<b>3</b>
<b>3</b>	<b>Código</b>	<b>4</b>
3.1	Elementos da struct Matrix . . . . .	4
3.2	Utilização no treinamento . . . . .	4

## 1 Introdução

O objetivo principal do projeto é fazer com que seja entendido quão grande é o benefício de paralelizar um modelo de aprendizado de máquina. Para isso nós escolhemos paralelizar modelos de regressão linear que são amplamente utilizados para prever algo a partir de dados iniciais.

## 2 Regressão linear

O modelo de regressão linear funciona basicamente prevendo uma função linear sobre dados previamente fornecidos. A imagem a seguir é uma reta que minimiza o erro quadrático médio sobre os pontos.



O erro quadrático médio é definido da seguinte maneira:

$$f = \frac{1}{N} \sum_{i=1}^N (y_i - h(x_i))^2 \quad (1)$$

Onde  $N$  é a quantidade de pontos e  $h$  é a função linear. Agora nos resta responder como encontrar a função mínima. É simples, devemos seguir a direção do gradiente sobre a função do erro quadrático mínimo. Seja nossa função definida da seguinte maneira:

$$y = w_0x_0 + w_1x_1 + \dots + w_dx_d \quad (2)$$

onde  $d$  é a dimensionalidade do nosso dado (2 na imagem apresentada) e  $x_0 = 1$ .

Vamos olhar então quanto cada parâmetro  $w_i$  da função afeta nossa função de erro. Para isso devemos ver qual a direção do gradiente na dimensão  $i$ .

Portanto devemos analisar

$$\nabla f(x_1, x_2, \dots, x_d) = \begin{bmatrix} \frac{\delta f}{\delta x_1} \\ \frac{\delta f}{\delta x_2} \\ \vdots \\ \frac{\delta f}{\delta x_d} \end{bmatrix} \quad (3)$$

e com base nisso atualizar nossos pesos para ajustar nossa função linear aos dados.

### 3 Código

A linguagem escolhida foi o C++ e nós decidimos fazer o projeto do zero implementando inclusive nossa própria biblioteca de matrizes. Nós fizemos essa escolha pela flexibilidade e rapidez da linguagem.

#### 3.1 Elementos da struct Matrix

Vamos apresentar o que foi necessário para fazer os cálculos.

- Limites da matriz (row, col, size).
- Vetor que guarda os itens da matriz (\*matrix).
- Construtores padrão e de cópia.
- Funções auxiliares de indexação.
- Operador de multiplicação serial (\*).
- Operador de multiplicação paralela (%).

#### 3.2 Utilização no treinamento