Tópico adicional: Vetorização

Esta aula constitui um tópico adicional da disciplina. Trata-se de um conteúdo opcional. Sua atividade não valerá nota e não precisa ser enviada.

Importância da Vetorização

Ao implementar um algoritmo de Aprendizado de Máquina, a vetorização provê dois benefícios principais:

1. Seu código torna-se mais compacto.
2. Seu algoritmo é capaz de rodar de forma mais rápida e eficiente.

Um código escrito de forma vetorizada é também capaz de extrair máximo proveito de:

- bibliotecas numéricas modernas e otimizadas para cálculo matemático, como a biblioteca de Álgebra Linear **NumPy**, por exemplo.
- elementos de hardware voltados ao processamento intenso de dados, como GPUs (*Graphical Processing Units*), por exemplo.

Exemplo de Vetorização

Sejam os seguintes parâmetros e características de um modelo:

$$\vec{w} = \begin{bmatrix} w_1 & w_2 & w_3 \end{bmatrix} \rightarrow (n = 3)$$

b (escalar)

$$\vec{x} = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}$$

Em Álgebra Linear, a indexação (contagem de elementos) começa em 1.

Em código escrito em Python (NumPy), a indexação começa em 0.

Exemplo:

```
w = np.array([1.0, 2.5, -3.3])
b = 4
x = np.array([10, 20, 30])
```

Para acessar o primeiro elemento de w, usamos w[0].
Para acessar o segundo elemento de w, usamos w[1].
Para acessarmos o terceiro elemento de w, usamos w[2].

Idem para os elementos do vetor x.

Exemplo de Vetorização

Suponha que você quer implementar uma previsão feita pelo modelo

$$f_{\vec{w},b}(\vec{x}) = w_1 x_1 + w_2 x_2 + w_3 x_3 + b$$

Código sem vetorização:

```
f = w[0]*x[0] + w[1]*x[1] + w[2]*x[2] + b
```

Pergunta:

Seria fácil programar a linha de código acima se tivéssemos $n = 1000$?

Exemplo de Vetorização

Código ainda sem vetorização, mas usando loop for:

Sabemos que

$$f_{\vec{w},b}(\vec{x}) = \left(\sum_{j=1}^{n} w_j x_j \right) + b$$

Portanto poderíamos usar o seguinte código:

```
f = 0
for j in range(0,n):
        f = f + w[j] * x[j]
f = f+b
```

Observação 1: Em Python, `j in range(0,n)` significa que j será $0, 1, 2, \cdots, n-1$.

Observação 2: Em Python, o comando `range(0,n)` faz a mesma coisa que `range(n)`

Pergunta:

Apesar de ser melhor que a nossa primeira implementação, essa segunda seria ainda a forma mais otimizada para a realização dos cálculos?

Código COM vetorização:

Sabendo que

$$f_{\vec{w},b}(\vec{x}) = \vec{w} \cdot \vec{x} + b$$

podemos implementar tal operação matemática usando uma **única linha de código**:

```
f = np.dot(w,x)+b
```

Exemplo de Vetorização

Observação:

Ao utilizarmos comandos da biblioteca NumPy, como f = np.dot(w,x)+b , por exemplo, estamos otimizando o nosso código, especialmente para os casos em que n é grande.

Isso acontece pois, por trás, a biblioteca **NumPy** utiliza **paralelismo de hardware** para a realização das operações matemáticas, mesmo que você esteja usando uma CPU comum ao invés de uma GPU.

Um segundo exemplo de Vetorização

Suponha que você deseja implementar o Método do Gradiente para um modelo com $b = 0$, $\overrightarrow{w} = (w_1, w_2, \cdots, w_{16})$, derivadas $\overrightarrow{d} = (d_1, d_2, \cdots, d_{16})$ e $\alpha = 0.1$, tal que

$$w_j = w_j - 0.1 d_j$$

Código sem vetorização:

```
for j in range(0,16):
        w[j] = w[j] - 0.1*d[j]
```

Código COM vetorização:

```
w = w - 0.1*d
```

De olho no código!

De olho no código!

Vamos agora aprender conceitos mais específicos acerca de Python, NumPy e Vetorização

Acesse o Python Notebook usando o QR code ou o link abaixo:

`https://colab.research.google.com/github/xaximpvp2/master/blob/main/codigo_aula6_topico_adicional.ipynb`

Atividade de aula

Parte 1

Rode todo o "codigo - Python, NumPy e Vetorização.ipynb" sem fazer qualquer tipo de alteração. Certifique-se de que você o compreendeu.

Parte 2

1. Qual foi a diferença de tempo observada entre rodar comandos usando "loop for" versus vetorização?