

Aplicações de Matrizes na Inteligência Artificial

Grupo 1

Angelo Gabriel, Gabriel Siqueira, Kauany Blomer, Julio Cesar

April 3, 2025

1 Introdução

As matrizes desempenham um papel crucial na inteligência artificial (IA), sendo amplamente utilizadas em operações como redes neurais, visão computacional e otimização. Este relatório explora as operações básicas com matrizes e suas aplicações na IA.

2 Operações Básicas com Matrizes

2.1 Adição de Matrizes

A adição de matrizes consiste em somar os elementos correspondentes de duas matrizes de mesma ordem:

$$A + B = \begin{bmatrix} a_{11} + b_{11} & a_{12} + b_{12} \\ a_{21} + b_{21} & a_{22} + b_{22} \end{bmatrix}$$

Aplicação na IA: Utilizada na combinação de pesos em redes neurais.

Exemplo Numérico:

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} + \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 6 & 8 \\ 10 & 12 \end{bmatrix}$$

Código em Python:

```
1 import numpy as np
2 A = np.array([[1, 2], [3, 4]])
3 B = np.array([[5, 6], [7, 8]])
4 C = A + B
5 print(C)
```

2.2 Subtração de Matrizes

A subtração de matrizes segue o mesmo princípio da adição:

$$A - B = \begin{bmatrix} a_{11} - b_{11} & a_{12} - b_{12} \\ a_{21} - b_{21} & a_{22} - b_{22} \end{bmatrix}$$

Aplicação na IA: Usada para calcular erros em aprendizado de máquina.

Exemplo Numérico:

$$\begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} - \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 4 & 4 \\ 4 & 4 \end{bmatrix}$$

Código em Python:

```
1 D = A - B
2 print(D)
```

3 Aplicando Criptografia Básica

Conforme o material recomendado [?], utilizamos matrizes para implementar um método simples de criptografia. O código abaixo demonstra como transformar uma mensagem em uma matriz e criptografá-la usando permutação de índices.

Código em Python:

```
1 # importando bibliotecas
2 # numpy para fazer operacoes com as matrizes
3 import numpy as np
4 # pra deixar um pouco mais interessante diferente do artigo
   fornecido os indices sao aleatoris nesse codigo
5 import random
6
7 # Defini o dos caracteres suportados
8 caracteres = ['A','B','C','D','E','F','G','H','I','J','K','L',
   ',','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z',''
   '.',' ','_',' ',' ' ]
9
10 # Criando uma permuta o aleat ria dos ndices
11 indices = random.sample(range(len(caracteres)), len(
   caracteres))
12 # fucao responsavel por capturar o index correspondente a
   cada caracter
13 def getIndex(Caracter):
14     for character, index in zip(caracteres, indices):
15         if Caracter == character:
```

```

16         return index
17     return None
18
19 # funcao responsavel por capturar o caracter responsavel por
20 # cada index
21 def getCaracter(Index):
22     for caracter, index in zip(caracteres, indices):
23         if Index == index:
24             return caracter
25     return None
26
27 # a mensagem e convertida em uma matriz correspondente aos
28 # indices criados acima
29 def criar_matriz(palavra):
30     palavra = palavra.upper()
31     tam = len(palavra)
32     tamanho_matriz = int(np.ceil(np.sqrt(tam)))
33     matriz = np.full((tamanho_matriz, tamanho_matriz),
34                      fill_value=' ')
35
36     idx = 0
37     for i in range(tamanho_matriz):
38         for j in range(tamanho_matriz):
39             if idx < tam:
40                 matriz[i][j] = palavra[idx]
41                 idx += 1
42     return matriz
43
44 def criptografar(palavra):
45     matriz = criar_matriz(palavra)
46     matriz_indices = np.vectorize(getIndex)(matriz)
47     return matriz_indices.flatten().tolist()
48
49 def descriptografar(codificada):
50     tamanho_matriz = int(np.sqrt(len(codificada)))
51     matriz = np.array(codificada).reshape((tamanho_matriz,
52                                             tamanho_matriz))
53     matriz_caracteres = np.vectorize(getCaracter)(matriz)
54     palavra = ''.join(matriz_caracteres.flatten())
55     return palavra.strip().capitalize()
56
57 # Teste do c digo
58 mensagem = 'Mensagem a ser criptografada'
59 mensagem_codificada = criptografar(mensagem)
60 mensagem_decodificada = descriptografar(mensagem_codificada)
61
62 print(f' ndices de permuta o: {indices}')
63 print(f'Mensagem original: {mensagem}')
64 print(f'Mensagem criptografada: {mensagem_codificada}')

```

```
61 print(f'Mensagem descriptografada: {mensagem_decodificada}')
```