

Assignment 6

Meta

Author: Parmandeep Chaddha Date: Feb 16, 2022

Objective

Be able to multiply two matrices, and show that matrix multiplication is not commutative.

multiplyMatrix

multiplyMatrix Multiplies two matrices. The second dimension of matA must equal the first dimension of matB.

Args: matA: A matrix (or array) with m by n elements. matB: Another matrix (or array) that should be n by p .

Returns: matC: A matrix with m by p elements.

- `include("/Users/pchaddha/OneDrive - University of Waterloo/Waterloo - 4B/psych_420_intro_to_computational_neuroscience/compNeuroIntro420/juliapsych420/assignment6/matrix_multiplication.jl")`

- `using LinearAlgebra` ✓

testCase1 (generic function with 1 method)

- `function testCase1()`
- `matA = [1; 2; 3] # 3x1`
- `matB = [1 2 3] # 1x3`
- `# Our expected result is a 1x1 matrix.`
- `matC = multiplyMatrix(matB, matA)`
- `return matC`
- `end`

`matC =` ▶ [14]

- `matC = testCase1()`

testCase2 (generic function with 1 method)

- `function testCase2()`
- `matA = [1; 2; 3] # 3x1`
- `matB = [1 2 3] # 1x3`

- `matB = [1 2 3] # 1x3`
- *# Our expected result is a 3x3 matrix.*
- `matC = multiplyMatrix(matA, matB)`
- `return matC`
- `end`

```
matCNotCommutative = 3x3 Matrix{Int64}:
      1  2  3
      2  4  6
      3  6  9
```

- *# Even though MatA and MatB are the same in testCase2, the result should be completely different because of the non-commutativity of matrix multiplication.*
- `matCNotCommutative = testCase2()`

testCase3 (generic function with 1 method)

- `function testCase3()`
- `matA = [1 2; 3 4] # 2x2`
- `matB = [2 1; 4 3] # 2x2`
- *# Our expected result is a 3x3 matrix.*
- `matC = multiplyMatrix(matA, matB)`
- `return matC`
- `end`

```
matC3 = 2x2 Matrix{Int64}:
      10  7
      22 15
```

- `matC3 = testCase3()`