

机器学习

1 Matrix-Matrix Multiplication

We multiply two matrices by breaking it into several vector multiplications and concatenating the result.

$$\begin{bmatrix} a & b \\ c & d \\ e & f \end{bmatrix} * \begin{bmatrix} w & x \\ y & z \end{bmatrix} = \begin{bmatrix} a*w + b*y & a*x + b*z \\ c*w + d*y & c*x + d*z \\ e*w + f*y & e*x + f*z \end{bmatrix}$$

An m x n matrix multiplied by an n x o matrix results in an m x o matrix. In the above example, a 3 x 2 matrix times a 2 x 2 matrix resulted in a 3 x 2 matrix.

To multiply two matrices, the number of columns of the first matrix must equal the number of rows of the second matrix.

For example:

```
# Initialize a 3 by 2 matrix
A = [1, 2; 3, 4;5, 6]

# Initialize a 2 by 1 matrix
B = [1; 2]

# We expect a resulting matrix of (3 by 2)*(2 by 1) = (3 by 1)
mult_AB = A*B

# Make sure you understand why we got that result
```

运行结果:

A =

```
1    2
3    4
5    6
```

B =

1
2

mult_AB =

5
11
17

2 Matrix Multiplication Properties

- Matrices are not commutative: $A*B \neq B*A$
- Matrices are associative: $(A*B)*C = A*(B*C)$

The identity matrix, when multiplied by any matrix of the same dimensions, results in the original matrix. It's just like multiplying numbers by 1. The identity matrix simply has 1's on the diagonal (upper left to lower right diagonal) and 0's elsewhere.

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

When multiplying the identity matrix after some matrix ($A*I$), the square identity matrix's dimension should match the other matrix's columns. When multiplying the identity matrix before some other matrix ($I*A$), the square identity matrix's dimension should match the other matrix's rows.

```
# Initialize random matrices A and B
```

```
A = [1,2;4,5]
```

```
B = [1,1;0,2]
```

```
# Initialize a 2 by 2 identity matrix
```

```
I = eye(2)
```

```
# The above notation is the same as I = [1,0;0,1]
```

```
# What happens when we multiply I*A ?
```

```
IA = I*A
```

```
# How about A*I ?
```

```
AI = A*I
```

```

# Compute A*B
AB = A*B

# Is it equal to B*A?
BA = B*A

# Note that IA = AI but AB != BA

```

运行结果:

A=

```

1 2
4 5

```

B=

```

1 1
0 2

```

I=

Diagonal Matrix

```

1 0
0 1

```

IA=

```

1 2
4 5

```

AI=

```

1 2
4 5

```

AB=

```

1 5
4 14

```

BA=

```

5 7
8 10

```

3 Inverse and Transpose

The inverse of a matrix A is denoted A^{-1} . Multiplying by the inverse results in the identity matrix.

A non square matrix does not have an inverse matrix. We can compute inverses of matrices in octave with the `pinv(A)` function and in Matlab with the `inv(A)` function. Matrices that don't have an inverse are singular or degenerate.

The transposition of a matrix is like rotating the matrix 90° in clockwise direction and then reversing it. We can compute transposition of matrices in matlab with the `transpose(A)` function or A' :

$$A = \begin{bmatrix} a & b \\ c & d \\ e & f \end{bmatrix}$$

$$A^T = \begin{bmatrix} a & c & e \\ b & d & f \end{bmatrix}$$

In other words:

$$A_{ij} = A_{ji}^T$$

```
# Initialize matrix A
A = [1,2,0;0,5,6;7,0,9]

# Transpose A
A_trans = A'

# Take the inverse of A
A_inv = inv(A)

# What is A^(-1)*A?
A_invA = inv(A)*A
```

运行结果:

```
A=
     1     2     0
     0     5     6
     7     0     9
```

A_trans=

1	0	7
2	5	0
0	6	9

A_inv=

0.348837	-0.139535	0.093023
0.325581	0.069767	-0.046512
-0.271318	0.108527	0.038760

A_invA=

1.00000	-0.00000	0.00000
0.00000	1.00000	-0.00000
-0.00000	0.00000	1.00000