

| 일반 소스 공유 | >

행렬 계산 함수



OtakoidTony 챗봇 고수 1:1 채팅

2019.12.08. 21:58 조회 81

댓글 5 URL 복사

```

1  var Matrix = {};
2
3  Matrix.dot = function(a, b) {
4      var aNumRows = a.length,
5          aNumCols = a[0].length,
6          bNumRows = b.length,
7          bNumCols = b[0].length,
8          m = new Array(aNumRows);
9      for (var r = 0; r < aNumRows; ++r) {
10         m[r] = new Array(bNumCols);
11         for (var c = 0; c < bNumCols; ++c) {
12             m[r][c] = 0;
13             for (var i = 0; i < aNumCols; ++i) {
14                 m[r][c] += a[r][i] * b[i][c];
15             }
16         }
17     }
18     return m;
19 }
20 Matrix.T = function(A) {
21     return A[0].map((col, i) => A.map(row => row[i]));
22 }
23 Matrix.shape = function(A) {
24     return [A.length, A[0].length];
25 }
26 Matrix.zeros = function(a, b) {
27     var i = 0;
28     var j = 0;
29     var A = [];
30     var B = [];
31     while (j < b) {
32         B.push(0);
33         j = j + 1;
34     }
35     while (i < a) {
36         A.push(B);
37         i = i + 1;
38     }
39     return A;
40 }
41 Matrix.numbers = function(a, b, n) {
42     var i = 0;
43     var j = 0;
44     var A = [];
45     var B = [];
46     while (j < b) {
47         B.push(n);
48         j = j + 1;
49     }
50     while (i < a) {
51         A.push(B);
52         i = i + 1;
53     }
54     return A;
55 }
56 Matrix.time = function(A, t) {
57     var out = [];

```

```

57     var out = [];
58     for (var i in A) {
59         out[i] = [];
60         for (var j in A[i]) {
61             out[i][j] = A[i][j] * t;
62         }
63     }
64     return out;
65 }
66 Matrix.multiply_scalar = function(A, B) {
67     var out = 0;
68     for (var i in A) {
69         for (var j in A[i]) {
70             out = out + A[i][j] * B[i][j];
71         }
72     }
73     return out;
74 }
75 Matrix.plus = function(A, B) {
76     var i = 0;
77     var j = 0;
78     var Output = [];
79     for (i in A) {
80         Output[i] = [];
81         for (j in A[i]) {
82             Output[i][j] = A[i][j] + B[i][j];
83         }
84     }
85     return Output;
86 }
87 Matrix.minus = function(A, B) {
88     var i = 0;
89     var j = 0;
90     var Output = [];
91     for (i in A) {
92         Output[i] = [];
93         for (j in A[i]) {
94             Output[i][j] = A[i][j] - B[i][j];
95         }
96     }
97     return Output;
98 }
99 Matrix.rand = function(a, b) {
100     var q = [];
101     var i = 0;
102     var j = 0;
103     while (i < a) {
104         q[i] = [];
105         while (j < b) {
106             q[i][j] = Math.random();
107             j = j + 1;
108         }
109         j = 0;
110         i = i + 1;
111     }
112     return q;
113 }
114 Matrix.sigmoid = function(A) {
115     var out = [];
116     for (var i in A) {
117         out[i] = [];
118         for (var j in A[i]) {
119             out[i][j] = 1 / (1 + Math.exp(-A[i][j]));
120         }
121     }
122     return out;
123 }
124 Matrix.sigmoid_derivative = function(A) {
125     var out = [];

```

```

126     for (var i in A) {
127         out[i] = [];
128         for (var j in A[i]) {
129             out[i][j] = A[i][j] * (1 - A[i][j]);
130         }
131     }
132     return out;
133 }
134 Matrix.ReLU = function(A) {
135     var out = [];
136     for (var i in A) {
137         out[i] = [];
138         for (var j in A[i]) {
139             if (A[i][j] >= 0) {
140                 out[i][j] = A[i][j];
141             } else {
142                 out[i][j] = 0;
143             }
144         }
145     }
146     return out;
147 }
148 Matrix.ReLU_derivative = function(A) {
149     var out = [];
150     for (var i in A) {
151         out[i] = [];
152         for (var j in A[i]) {
153             if (A[i][j] >= 0) {
154                 out[i][j] = 1;
155             } else {
156                 out[i][j] = 0;
157             }
158         }
159     }
160     return out;
161 }
162 Matrix.multiply = function(A, B) {
163     var answer = [];
164     for (var i = 0; i < A.length; i++) {
165         answer[i] = [];
166         for (var j = 0; j < A[0].length; j++) {
167             answer[i][j] = A[i][j] * B[i][j];
168         }
169     }
170     return answer;
171 }

```



OtakoidTony님의 게시글 더보기 >

❤️ 좋아요 0 💬 댓글 5

🔗 공유 | 신고

댓글 등록순 최신순

댓글알림



SOODA

multiply 함수에서 저 경우는 두 행렬이 같은 크기일때만 작동하는거 아닌가요?
일단 C로 짠 코드지만 두 행렬의 곱 함수 남겨두고 가요

```

void multi(int A[p][q], int B[q][r]) {
    for (int y = 0; y < p; y++) {
        for (int x = 0; x < r; x++) {
            int c = 0;
            for (int i = 0; i < q; i++) {
                c += A[y][i] * B[i][x];
            }
            printf("%d ", c);

```

```
}  
printf("%n");  
}  
}
```

2019.12.08. 23:05 답글 쓰기



OtakoidTony 작성자

에.... 그거 행렬곱인 것 같은데..... 행렬곱은 Matrix.dot이여~!

2019.12.08. 23:22 답글 쓰기



SOODA

OtakoidTony 와 그렇네요 제가 켈 위에는 당연히 변수 선언하고 그런것들 있을거 같아 쓱 넘기고 아래 보다가 multiply만 눈에 보여서 댓글 달았는데 ㅎㅎ

2019.12.08. 23:28 답글 쓰기



OtakoidTony 작성자

SOODA 아ㅋㅋㅋㅋㅋ

2019.12.08. 23:29 답글 쓰기



새름

중급소스로 이동하였습니다 |

2019.12.09. 00:08 답글 쓰기

Hibot

댓글을 남겨보세요



등록

글쓰기

답글

목록

▲ TOP

'| 일반 소스 공유 |' 게시판 글

이 게시판 새글 구독하기 ☐

| | | |
|-------------------------------|-------------|-------------|
| 파이값 계산하기 [6] | SOODA | 2019.12.09. |
| 영화 검색 Jsoup 🍷 [7] | doami2005 | 2019.12.08. |
| 행렬 계산 함수 [5] | OtakoidTony | 2019.12.08. |
| 심층 신경망 여러 은닉층을 이용한 머신러닝 [6] | OtakoidTony | 2019.12.08. |
| 행렬 연산 | scap | 2019.12.08. |

이 카페 인기글



네이버에서 지역별 날씨 정보 크롤링하기 / 스압 알아서 주의

Dark Tornado
Steve28
♡2 💬4

요즘 나타나는 새로운 홍보 수법 중 하나가

Dark Tornado
♡0 💬7

마녀사냥 금지

AlphaDo
♡3 💬8

kakao.py에 대해서

undefined
♡0 💬13

간편 자동응답과 한글코딩을 메인으로 제공
하는 카톡봇앱 개발 예정

성빈
♡1 💬16

자동학습 봇 특정 말만 무시하기?

헤히
♡0 💬4

예전에 어떤분이 카톡봇을 이용하여 카페에
게시글을 올렸었는데

리페
♡0 💬1

안녕하세요

tomohong
♡1 💬3