Date. No.

< 11주차 >

20155137 안원영

• 전치행렬 만들기 => 열과 행을 바꾼 행렬.

```
lt = [[1,2,3],[4,5,6]]   => 2행 3열.
def lt_trans(lt):
    lt_a = [ ]; // 저장할 임의의 배열 생성.
    row, col = 2, 3;
    for i in range(3): // 열갯수
        temp = [ ]
        for j in range(2): // 행갯수
            temp.append(lt[j][i])
        lt_a.append(temp)
    return lt_a

print(lt_trans(lt));
```

↳ [[1 2 3 / 4 5 6]] ⓣ=> [1 4 / 2 5 / 3 6]

↳ 한번더 해주면 원래대로 돌아옴.

• 내적 구하기.

(2×2) • (2×1) = (2×1)이 나옴

두개가 같아야 내적함

```
lta = [[1,2],[3,4]]   => 2×2
ltb = [[5],[6]]       => 2×1   (두개가 맞아야함)

for i in lta:
    temp = [ ]
    for j in lt_trans(ltb):      # 전치해줘야함.
        temp.append(sum(lt_mul(i,j)))   # 곱해주는함수
    ltc.append(temp)
return ltc
```

=> np.dot(lta, ltb)로 확인 => 대신 lta와 ltb는 np.array( ) 해줘야함.

• 2차행렬 가운데 0으로 만드는 문제.

[1 2 3 4 / 5 6 7 8 / 9 10 11 12] => [1 2 3 4 / 5 0 0 8 / 9 10 11 12]

```
row, col = 3, 4
li = [[i+1 + j*col for i in range(col)]
        for j in range(row)]
```

i+1 → 5 = (열의갯수×1(현재열))+1

```
np1 = np.array(li)
```

① 슬라이싱 이용

```
np[1:-1, 1:-1] = 0
```

(1행 1열) 선택됨 / (5, 6, 7, 8) 선택됨

② 인덱스 이용 (ones 이용) => 1로 다 채우고 필요한부분 나머지 0으로

```
idx = np.ones(shape=(row,col), dtype=bool)
idx[0, :] = False , idx[-1, :] = False
idx[:, 0] = False , idx[:, -1] = False
np1[idx] = 0   => true인 부분만 0 채워짐.
```

③ 인덱스 이용 (zeros 이용) => 0으로 다 채우고 필요한부분만 1

```
idx = np.zeros(shape=(row,col), dtype=bool)
idx[1:-1, 1:-1] = true   => np1[idx] = 0
```

```
def lt_dot(lta, ltb):      # 내적
    if not isNumber(lta) or not isNumber(ltb):     # 숫자아닌경우
        print("error isNum");   => 숫자가 아닌경우 에러나게
        return false;
    if lt_shape(lta)[1] != lt_shape(ltb)[0]:     # 열과 행이 맞지 않을경우
        # ↳ lta의 열출력   ↳ ltb의 행출력
        print("error lt_shape()")
        return false;
    ltc = [ ];
```