

Data Science(Machine Learning, Deep Learning)

- 소스 깃 참고

DataScience
Numpy와 호환

Tensor = 일반적으로 3차원 이상의 배열을 의미, 2차원 배열도 넓은 범위로 텐서로 정의한다.

함수와 변수
[] index

텐서의 연산

$$a = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, b = \begin{bmatrix} 2 & 2 \\ 3 & 3 \end{bmatrix}$$

$$a + b = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} + \begin{bmatrix} 2 & 2 \\ 3 & 3 \end{bmatrix} = \begin{bmatrix} 1+2 & 2+2 \\ 3+3 & 4+3 \end{bmatrix} = \begin{bmatrix} 3 & 4 \\ 6 & 7 \end{bmatrix}$$

$$a - b = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, a * b = \begin{bmatrix} 2 & 4 \\ 9 & 12 \end{bmatrix}, a / b = \begin{bmatrix} 0.5000 & 1.0000 \\ 1.0000 & 1.3333 \end{bmatrix}, a ** b = \begin{bmatrix} 1 & 4 \\ 27 & 64 \end{bmatrix}$$

inplace 연산

원본 텐서의 값을 직접 변경합니다. 이것은 텐서를 메모리 내에서 새로운 공간에 저장하는 대신에 원본 텐서의 내용을 수정

보통 연산 뒤에 접미사로 _ (밑줄)을 붙여서 사용

차원 축소 연산

차원 축소 연산(Dimension Reduction Operation)은 다차원 배열 또는 텐서의 차원을 줄이고, 데이터를 요약하거나 특정 방식으로 집계하는 연산입니다. 주로 데이터 분석, 머신 러닝 및 딥 러닝에서 사용

[합과 평균]

텐서 x의 조건

$$x = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

$$\sum(x, \dim = 0) = \begin{bmatrix} 1 & 2 \\ + & + \\ 3 & 4 \end{bmatrix} = [4 \ 6]^T$$

$$\sum(x, \dim = -1) = \begin{bmatrix} 1+2 \\ 3+4 \end{bmatrix} = \begin{bmatrix} 3 \\ 7 \end{bmatrix}$$

브로드캐스트 연산

브로드캐스트(Broadcast) 연산은 서로 다른 형상(shape)을 가진 배열 또는 텐서 사이에서 연산을 수행할 때 사용되는 기술

브로드캐스트 연산 방식은 Tensor+Scalar, Tensor+Vector, Tensor+Tensor가 있음

$$\begin{bmatrix} 1 & 2 \\ 4 & 8 \end{bmatrix} \oplus \begin{bmatrix} 3 \\ 5 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 4 & 8 \end{bmatrix} + \begin{bmatrix} 3 & 5 \\ 3 & 5 \end{bmatrix} = \begin{bmatrix} 4 & 7 \\ 7 & 13 \end{bmatrix}$$

Tensor 형태 변환

View 메모리에 순차대로 선언된 Tensor에 대해서만 작동

Squeeze와 Unsqueeze

<Tensor 자르기 붙이기>

Indexing과 Slicing

Split

Chunk

Index Select

Indice = 예약어

예약어 : 컴퓨터 프로그래밍 언어에서 이미 문법적인 용도로 사용되고 있기 때문에 식별자로 사용할 수 없는 단어들

Index = Tensor 같이 차원이 있음.

Concatenate

Stack