

1-1. 스칼라, 벡터, 행렬, 텐서

1. 스칼라(Scalar)

스칼라는 단일 값을 의미합니다. 이는 차원이 없는 숫자라고 생각할 수 있습니다.

예를 들어, 3, -2.5, π 등은 모두 스칼라입니다. 스칼라는 하나의 크기만 갖고 있으며 방향은 없습니다. 일반적으로 실수나 복소수 등의 숫자를 나타낼 때 사용됩니다.

예시 :

- $a = 5$
- $b = -3.14$

스칼라는 딥러닝에서 학습률, 손실 값 등과 같은 하나의 값을 표현할 때 자주 사용됩니다.

```
# 스칼라
a = 5
b = -3.14
print(f"a: {a}, b: {b}")
```

2. 벡터(Vector)

벡터는 여러 개의 숫자를 모아서 하나의 리스트로 나타낸 것입니다. 벡터는 크기와 방향을 가지며, 1차원 배열이라고 할 수 있습니다. 벡터는 n-차원의 공간에서 한 점을 나타낼 수 있습니다.

예시 :

- $v = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$ (3차원 벡터)
- $w = \begin{pmatrix} 4 \\ 5 \end{pmatrix}$ (2차원 벡터)

벡터는 딥러닝에서 주로 입력 데이터나 가중치 등을 표현할 때 사용됩니다. 예를 들어, 한 이미지의 RGB 값을 하나의 벡터로 표현할 수 있습니다.

```
import numpy as np
```

```
# 벡터
v = np.array([1, 2, 3]) # 1차원 배열로 벡터 표현
print("v:", v)
```

3. 행렬 (Matrix)

행렬은 벡터가 모여서 이루어진 2차원 배열입니다. 행과 열을 가지며, 다양한 형태의 데이터를 다룰 수 있게 해줍니다. 행렬은 딥러닝에서 가중치나 편향을 저장하는데 많이 사용됩니다.

예시 :

$$\bullet M = \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix} (3 \times 2 \text{행렬})$$

행렬 연산은 딥러닝의 신경망에서 매우 중요한 역할을 합니다. 예를 들어, 각 층(layer)에서의 가중치 업데이트나 활성화 함수 적용 등이 행렬 연산을 통해 이루어집니다.

```
import numpy as np
# 행렬
M = np.array([[1, 2], [3, 4], [5, 6]]) # 3x2 행렬
print("M:\n", M)
```

4. 텐서(Tensor)

텐서는 스칼라, 벡터, 행렬을 일반화한 개념입니다. 즉, 텐서는 다차원 배열을 의미합니다. 0차원 텐서는 스칼라, 1차원 텐서는 벡터, 2차원 텐서는 행렬입니다. 그 이상 차원의 배열도 모두 텐서로 표현됩니다.

예시 :

$$\bullet \text{ 3차원 텐서 : } T \in R^{2 \times 3 \times 4}$$

딥러닝에서 텐서는 신경망의 입력, 출력, 가중치 등 여러 종류의 데이터를 다루기 위해 사용됩니다. 특히, 이미지는 3차원 텐서로 표현될 수 있는데, 높이, 너비, 그리고 색상 채널(RGB)을 각각 차원으로 갖습니다.

```
import numpy as np
# 텐서 (3차원 배열)
```

```
T = np.array([[[1, 2], [3, 4]], [[5, 6], [7, 8]]]) # 2x2x2
print("T:\n", T)
```

요약

- **스칼라**: 하나의 숫자 (0차원)
- **벡터**: 1차원 배열 (1차원)
- **행렬**: 2차원 배열 (2차원)
- **텐서**: 다차원 배열 (3차원 이상)