

## 2장 신경망의 수학적 구성 요소

아래 요소들을 직관적으로 살펴본다.

- 텐서(tensor)
- 텐서 연산
- 미분
- 경사 하강

## 2.1 신경망 소개

MNIST 데이터셋 샘플 예제



그림 출처: [Deep Learning with Python\(Manning MEAP\)](#)

## 2.2 신경망에 사용되는 데이터 표현

## 데이터 텐서 사례

앞으로 다룰 데이터는 크게 아래 네 종류의 텐서들이다.

- 벡터 데이터: (샘플 수, 특성 수) 모양의 2D 텐서
- 시계열 또는 시퀀스 데이터: (샘플 수, 타임 스텝 수, 특성 수) 모양의 3D 텐서
- 이미지 데이터: (샘플 수, 가로, 세로, 채널 수) 또는 (샘플 수, 채널 수, 가로, 세로) 모양의 4D 텐서
- 동영상 데이터: (샘플 수, 프레임 수, 가로, 세로, 채널 수) 모양의 5D 텐서

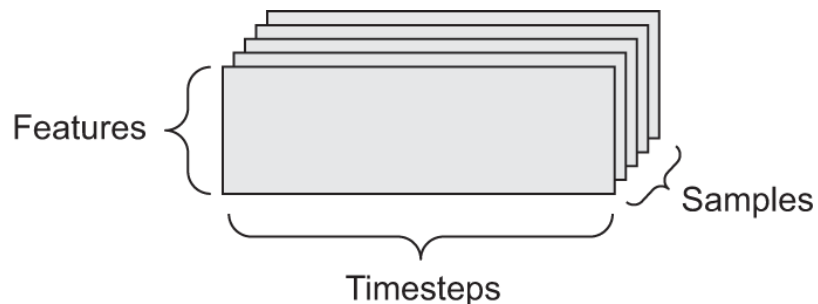
## 벡터 데이터

- 각 샘플은 여러 개의 특성값으로 이루어진 벡터
- 데이터 배치는 따라서 2D 텐서, 즉 벡터들의 어레이로 표현됨.
- 0번 축은 샘플 축, 1번 축은 **특성 축**임.
- 예제 1
  - 사람의 나이, 우편 번호, 소득 세 개의 특성으로 구성된 이니구 통계 데이터
  - 10만 명의 통계 데이터를 포함한 데이터셋은 **(100000, 3)** 모양의 2D 텐서임.
- 예제 2
  - 2만개의 단어 각각이 하나의 문서에서 사용된 빈도수로 구성된 데이터는 길이(차원) 2만인 벡터로 표현 가능.
  - 500개의 문서를 대상으로 할 경우 **(500, 20000)** 모양의 2D 텐서로 표현됨.

# 시계열 또는 시퀀스 데이터

- 증시 데이터, 트위터 데이터 등 시간의 흐름 또는 순서가 중요한 데이터를 다룰 때 사용
- 예제 1(시간의 흐름)
  - 매 분마다 측정된 (현재 증시 가격, 지난 1분 동안의 최고가, 지난 1분 봉안의 최저가) 데이터
  - 하루에 390번 250일 동안 측정해서 얻은 데이터셋은  $(250, 390, 3)$  모양의 3D 텐서.
- 예제 2(순서)
  - 하나의 트윗: 최대 280개의 문자 사용 가능. 문자는 총 128개 중에 하나.
  - 하나의 트윗을  $(280, 128)$  모양의 2D 텐서로 표현 가능. 문자 하나를 0과 1로 구성된 길이가 128인 벡터로 표현. 사용된 문자의 위치에서는 1, 나머지는 0.
  - 백만 개의 트윗은  $(1000000, 280, 128)$  모양의 3D 텐서.

## 3D 데이터 텐서





# 이미지 데이터

- 이미지는 보통 3D 텐서로 표현: (가로, 세로, 컬러 채널 수)
  - 컬러 이미지의 컬러 채널 수: 3
  - 흑백 이미지의 컬러 채널 수: 1
- 256x256 크기의 흑백 이미지 128개로 구성된 배치: (128, 256, 256, 1) 모양의 4D 텐서
- 256x256 크기의 컬러 이미지 128개로 구성된 배치: (128, 256, 256, 3) 모양의 4D 텐서

## 4D 데이터 텐서

## 동영상 데이터

- 동영상은 프레임(frame)으로 구성된 순차 데이터. 하나의 프레임은 컬러 이미지. 따랏 3D 텐서로 표현.
- 따라서 하나의 동영상은 (프레임 수, 가로, 세로, 컬러 채널 수) 모양의 4D 텐서로 표현 가능.
- 여러 개의 동영상으로 이루어진 데이터셋: (동영상 수, 프레임 수, 가로, 세로, 컬러 채널 수) 모양의 5D 텐서.
- 예제: 144x256 크기의 프레임으로 구성된 60초 길이의 동영상. 초당 4개의 프레임 구성.
  - 총 240 프레임.
  - 총 10개의 동영상으로 구성된 데이터셋: (10, 240, 144, 256, 3) 모양의 5D 텐서

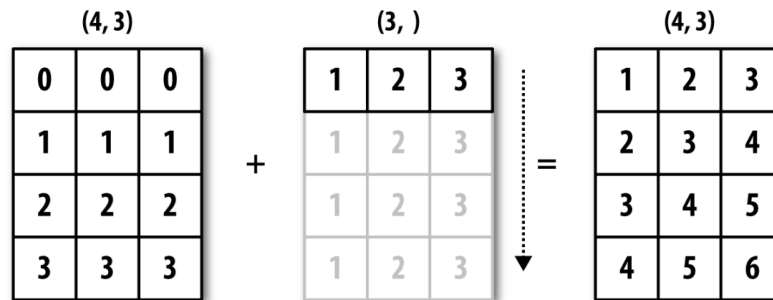
## 2.3 텐서 연산

## 브로드캐스팅(Broadcasting)

두 텐서의 모양이 아래와 같을 때 항목별 연산을 위한 브로드캐스팅이 가능하다.

$(a, b, \dots, n, n + 1, \dots, m)$  와  $(n, n + 1, \dots, m)$

- 둘째 텐서에 대해 브로드캐스팅 발생
- $a$  부터  $n-1$  까지의 축에 대해 축(axis)이 자동으로 추가됨.



(4, 3)                      (4, 1)                      (4, 3)

|   |   |   |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 1 | 1 |
| 2 | 2 | 2 |
| 3 | 3 | 3 |

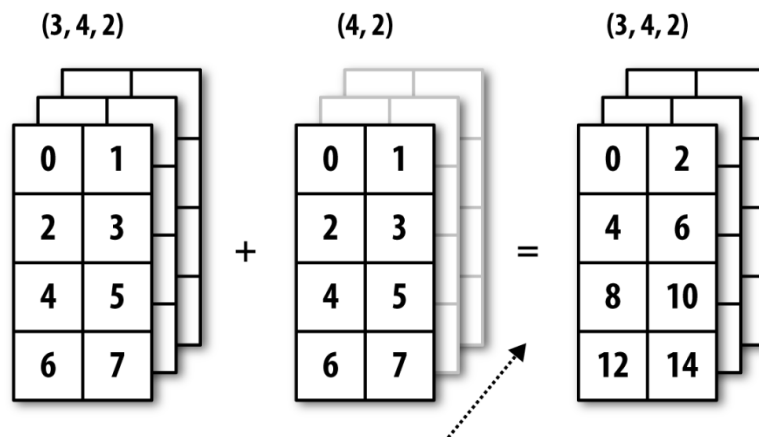
+

|   |   |   |
|---|---|---|
| 1 | 1 | 1 |
| 2 | 2 | 2 |
| 3 | 3 | 3 |
| 4 | 4 | 4 |

=

|   |   |   |
|---|---|---|
| 1 | 1 | 1 |
| 3 | 3 | 3 |
| 5 | 5 | 5 |
| 7 | 7 | 7 |

.....→



텐서 연산의 기하학적 의미



이동: 벡터 합

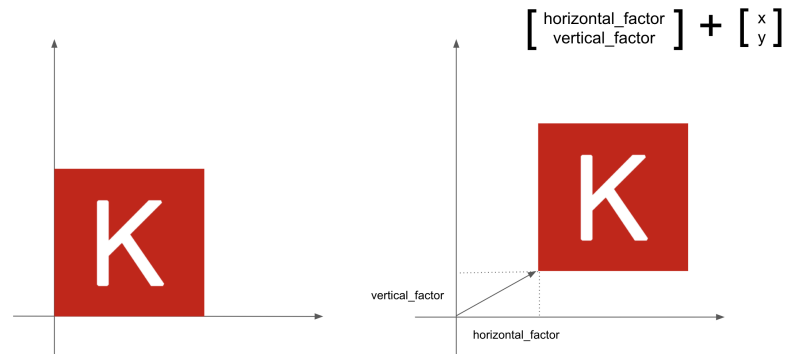


그림 출처: [Deep Learning with Python\(Manning MEAP\)](#)

회전: 점곱

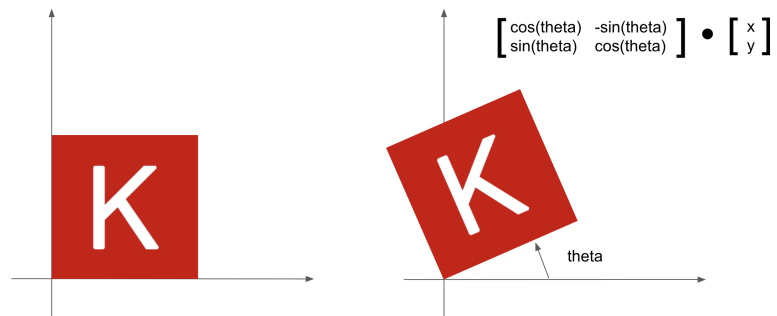


그림 출처: [Deep Learning with Python\(Manning MEAP\)](#)

스케일링: 점곱

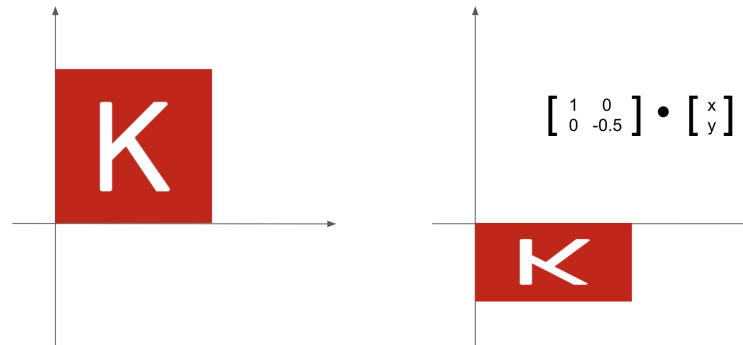


그림 출처: [Deep Learning with Python\(Manning MEAP\)](#)

## 아핀 변환

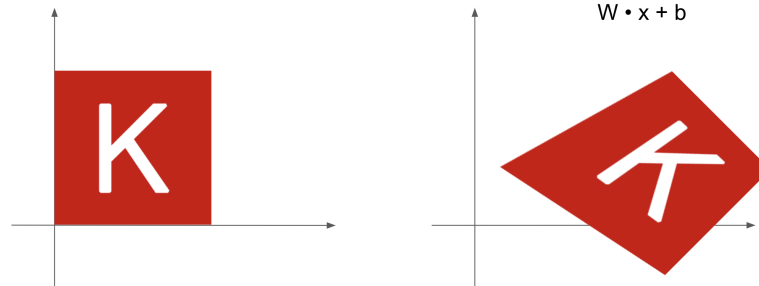


그림 출처: [Deep Learning with Python\(Manning MEAP\)](#)

## 아핀 변환과 relu 활성화 함수

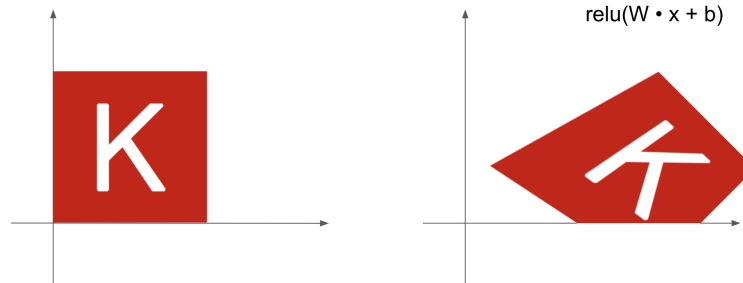


그림 출처: [Deep Learning with Python\(Manning MEAP\)](#)

## 딥러닝의 기하학적 의미

- 신경망은 기본적으로 앞서 언급된 텐서 연산의 조합에 불과함.
- 즉, 고차원 공간에서의 매우 복잡한 기하학적 변환을 단순한 텐서 연산의 조합으로 해결함.
- 예제: 3차원 매니폴드
  - 빨간 종이와 파란 종이 두 장을 겹쳐 뭉친 입력값
  - 연속된 종이 펼치기 과정을 통해 명료하게 구분할 수 있는 두 장의 종으로 펼치기
  - 펼치기 과정에 사용된 과정: 손가락으로 조금씩 펼치기
  - 딥러닝 모델의 많은 층(layer)에서 이루어지는 일과 크게 다르지 않음.

### 복잡한 다양체 펼치기



그림 출처: [Deep Learning with Python\(Manning MEAP\)](#)