

# 부록: 텐서 소개

# 텐서

- 넘파이 어레이 `numpy.ndarray`
  - 대표적인 **텐서**<sub>tensor</sub> 자료형
  - 머신러닝에 사용되는 데이터셋은 일반적으로 텐서로 저장됨
- 텐서플로우
  - `Tensor` 자료형인 `tensorflow.Tensor`
  - 넘파이 어레이와 유사하며 GPU를 활용한 연산 지원
- 케라스 신경망 모델의 입력, 출력값
  - 넘파이 어레이를 기본으로 사용
  - 내부적으로는 `tf.Tensor` 로 변환해서 사용

## 텐서의 차원

- 텐서의 표현에 사용된 **축**<sub>axis</sub>의 개수
- **랭크**<sub>rank</sub>로도 불림

## 0차원(0D) 텐서 (랭크-0 텐서)

- 정수 한 개, 부동소수점 한 개 등 하나의 수를 표현하는 텐서.
- **스칼라** scalar라고도 불림.

```
np.array(12)  
np.array(1.34)
```

## 1차원(1D) 텐서 (랭크-1 텐서)

- 수로 이루어진 리스트 형식의 텐서.
- **벡터**vector로 불리며 한 개의 축을 가짐.

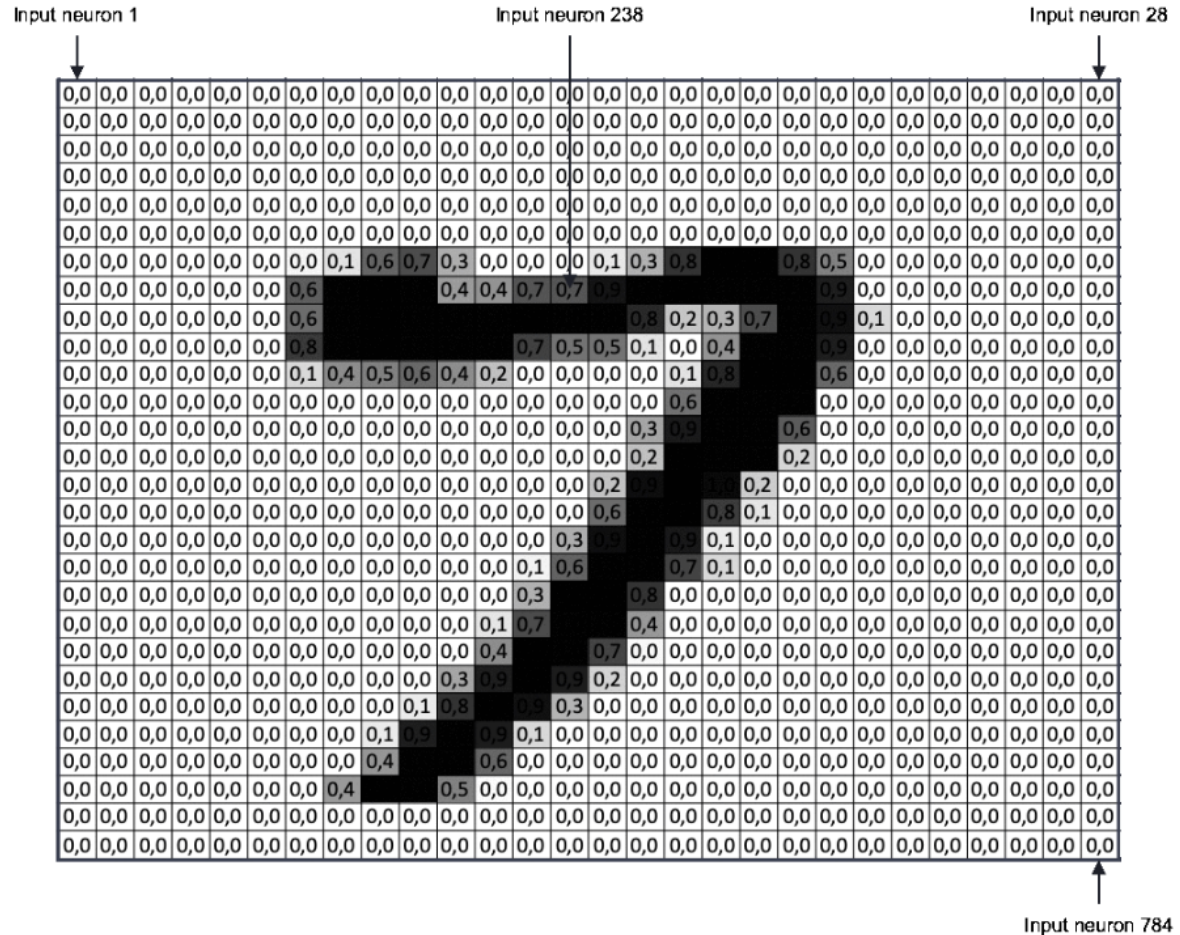
```
np.array([12, 3, 6, 14, 7])
```

## 2차원(2D) 텐서 (랭크-2 텐서)

- 행과 열 두 개의 축을 가짐.
- **행렬** `matrix`로도 불림.

```
np.array([[5, 78, 2, 34, 0],  
         [6, 79, 3, 35, 1],  
         [7, 80, 4, 36, 2]])
```

## 2D 텐서 예제: 흑백 사진 데이터



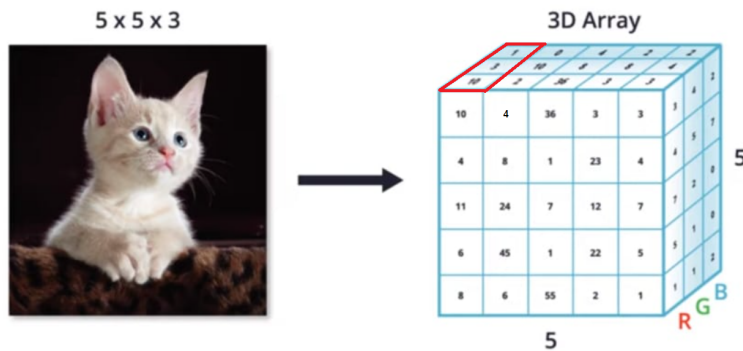
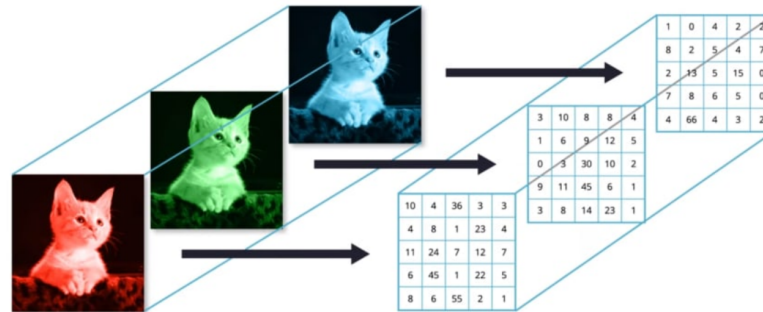
## 3차원(3D) 텐서 (랭크-3 텐서)

- 행, 열, 깊이 세 개의 축 사용.
- 동일 모양의 2D 텐서로 구성된 벡터로 이해 가능.

```
np.array([[[5, 78, 2, 34, 0],  
          [6, 79, 3, 35, 1],  
          [7, 80, 4, 36, 2]],  
         [[5, 78, 2, 34, 0],  
          [6, 79, 3, 35, 1],  
          [7, 80, 4, 36, 2]],  
         [[5, 78, 2, 34, 0],  
          [6, 79, 3, 35, 1],  
          [7, 80, 4, 36, 2]]])
```



## 3D 텐서 예제: 컬러 사진 데이터



## 4D 텐서 (랭크-4 텐서)

- 3D 텐서로 이루어진 벡터
- 예제
  - 컬러 사진 데이터로 구성된 훈련셋
  - 동영상: 연속된 컬러 사진의 데이터셋으로 간주 가능

## 주의사항: 벡터의 차원

- 벡터의 길이를 차원이라 부르기도 함
- 예제: `np.array([12, 3, 6, 14, 7])` 는 5차원 벡터

## 텐서 주요 속성

- 예제: `train_images` 가 MNIST의 훈련셋을 가리킴
- `ndim` 속성: 텐서의 차원 저장.

```
>>> train_images.ndim  
3
```

- `shape` 속성: 텐서의 모양을 튜플로 저장.

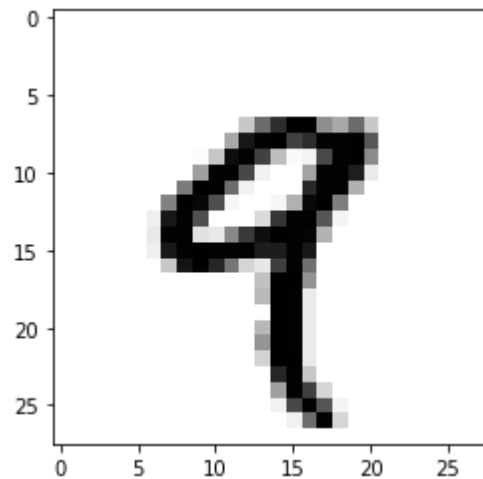
```
>>> train_images.shape  
(60000, 28, 28)
```

- `dtype` 속성: 텐서에 포함된 항목의 통일된 자료형.

```
>>> train_images.dtype  
uint8
```

# 인덱싱

```
>>> import matplotlib.pyplot as plt
>>> digit = train_images[4]
>>> plt.imshow(digit, cmap=plt.cm.binary)
>>> plt.show()
```



# 슬라이싱

- 첫째 배치

```
>>> batch = train_images[:128]
```

- 둘째 배치

```
>>> batch = train_images[128: 256]
```

- $n$  번째 배치

```
>>> batch = train_images[128 * n:128 * (n + 1)]
```

텐서 실전 예제

## 2D 텐서 예제: 캘리포니아 구역별 인구조사 데이터셋

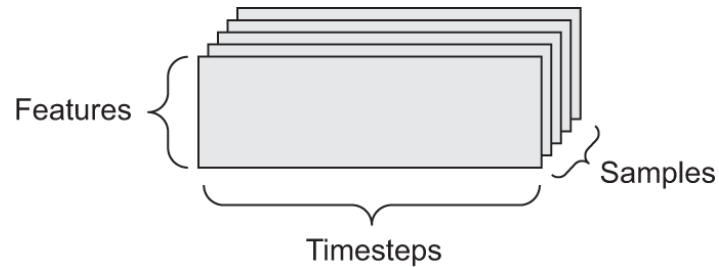
- 20,640개의 구역별 데이터 포함. 따라서 (20640, 10) 모양의 2D 텐서로 표현 가능.

	A	B	C	D	E	F	G	H	I	J
1	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value	ocean_proximity
2	-122.23	37.88	41	880	129	322	126	8.3252	452600	NEAR BAY
3	-122.22	37.86	21	7099	1106	2401	1138	8.3014	358500	NEAR BAY
4	-122.24	37.85	52	1467	190	496	177	7.2574	352100	NEAR BAY
5	-122.25	37.85	52	1274	235	558	219	5.6431	341300	NEAR BAY
6	-122.25	37.85	52	1627	280	565	259	3.8462	342200	NEAR BAY
7	-122.25	37.85	52	919	213	413	193	4.0368	269700	NEAR BAY
8	-122.25	37.84	52	2535	489	1094	514	3.6591	299200	NEAR BAY
9	-122.25	37.84	52	3104	687	1157	647	3.12	241400	NEAR BAY
10	-122.26	37.84	42	2555	665	1206	595	2.0804	226700	NEAR BAY
11	-122.25	37.84	52	3549	707	1551	714	3.6912	261100	NEAR BAY
12	-122.26	37.85	52	2202	434	910	402	3.2031	281500	NEAR BAY
13	-122.26	37.85	52	3503	752	1504	734	3.2705	241800	NEAR BAY
14	-122.26	37.85	52	2491	474	1098	468	3.075	213500	NEAR BAY
15	-122.26	37.84	52	696	191	345	174	2.6736	191300	NEAR BAY
16	-122.26	37.85	52	2643	626	1212	620	1.9167	159200	NEAR BAY
17	-122.26	37.85	50	1120	283	697	264	2.125	140000	NEAR BAY
18	-122.27	37.85	52	1966	347	793	331	2.775	152500	NEAR BAY
19	-122.27	37.85	52	1228	293	648	303	2.1202	155500	NEAR BAY
20	-122.26	37.84	50	2239	455	990	419	1.9911	158700	NEAR BAY



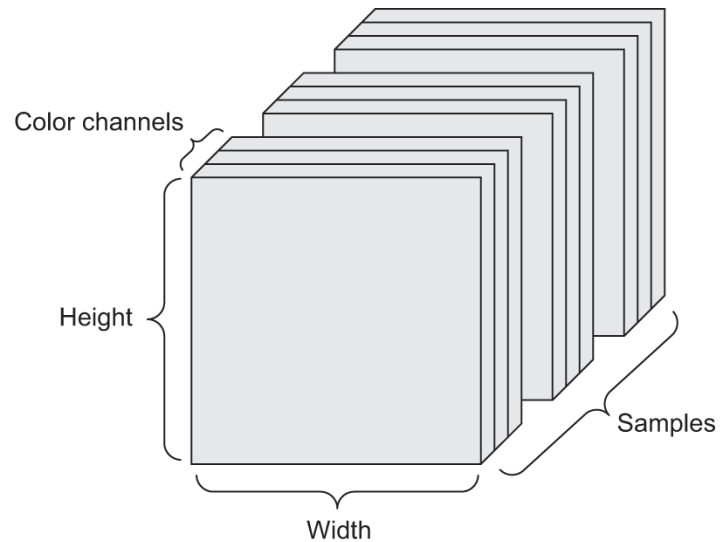
## 3D 텐서 예제

- 시계열 데이터
- 샘플: 1분마다 하루 총 390번 (현재 증시가, 지난 1분 동안 최고가, 지난 1분 동안 최저가)를 측정한 데이터. (390, 3) 모양의 2D 텐서로 표현.
- 데이터셋: 250일 동안 측정한 데이터셋은 (250, 390, 3) 모양의 3D 텐서로 표현.



## 4D 텐서 예제

- 컬러 사진으로 구성된 데이터셋: (샘플 수, 높이, 너비, 채널 수) 또는 (샘플 수, 채널 수, 높이, 너비) 모양의 4D 텐서로 표현



# 텐서 연산

```
keras.layers.Dense(512, activation="relu")  
keras.layers.Dense(10, activation="softmax")
```

- 첫째 층

- `W1`: 첫째 층에서 학습되는 가중치 행렬
- `b1`: 첫째 층에서 학습되는 편향 벡터

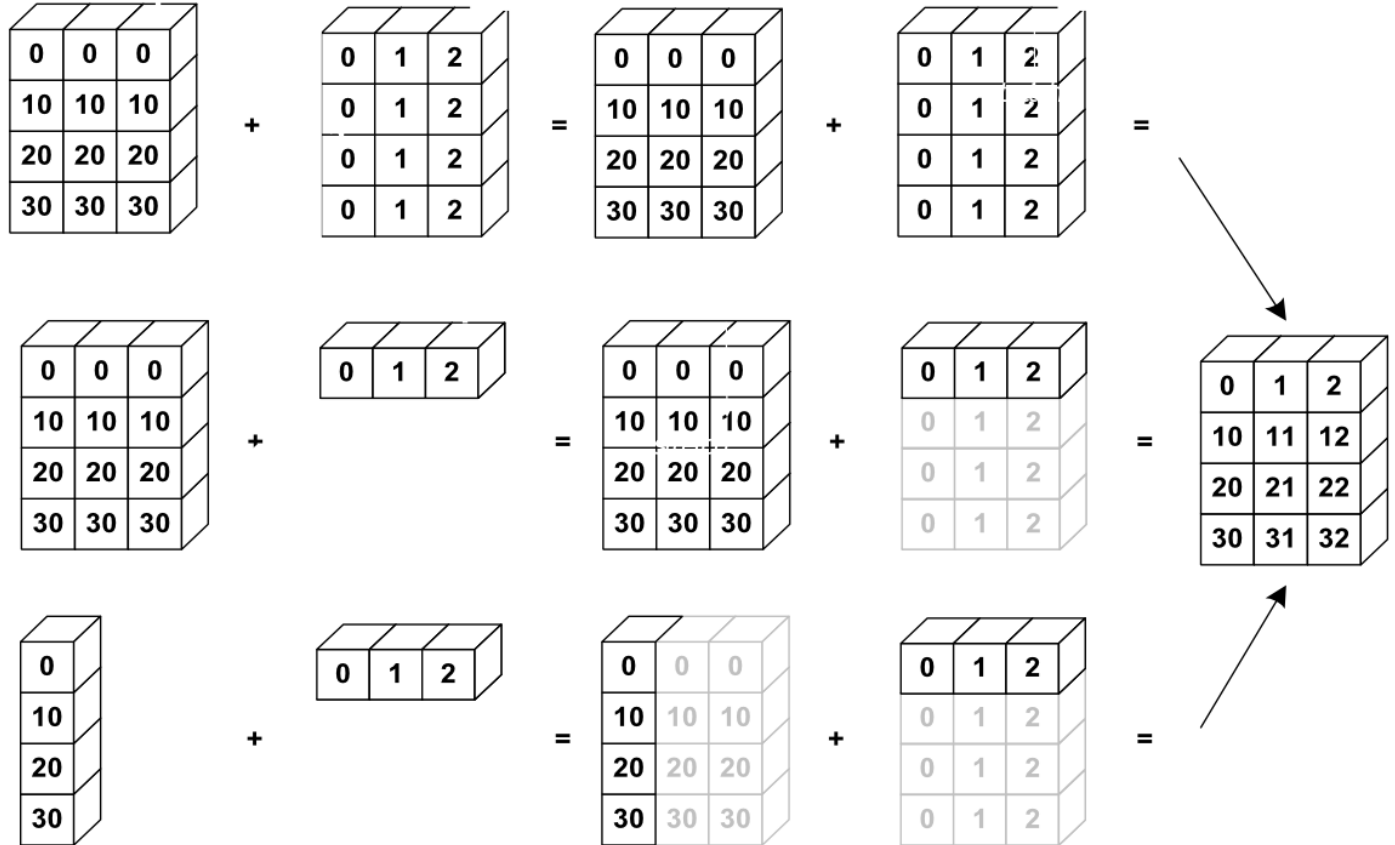
```
output = relu(np.dot(input, W1) + b1)
```

- 둘째 층

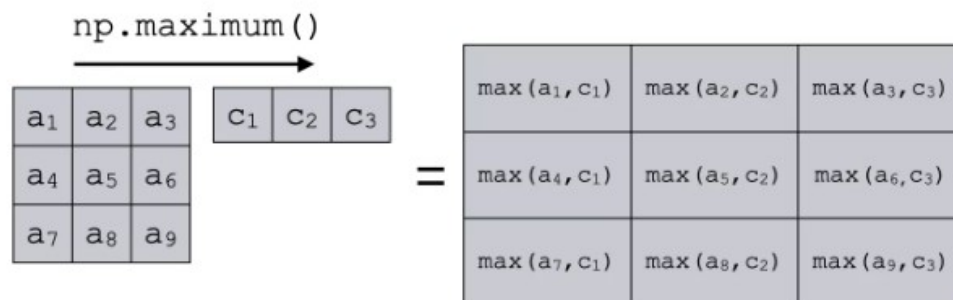
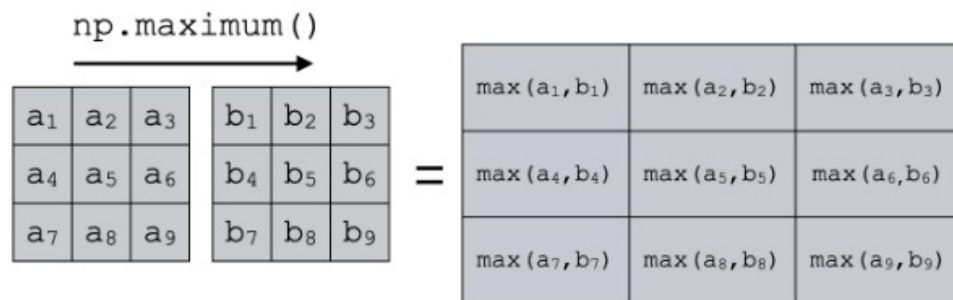
- `W2`: 둘째 층에서 학습되는 가중치 행렬
- `b2`: 둘째 층에서 학습되는 편향 벡터

```
output = softmax(np.dot(input, W2) + b2)
```

# 항목별 연산

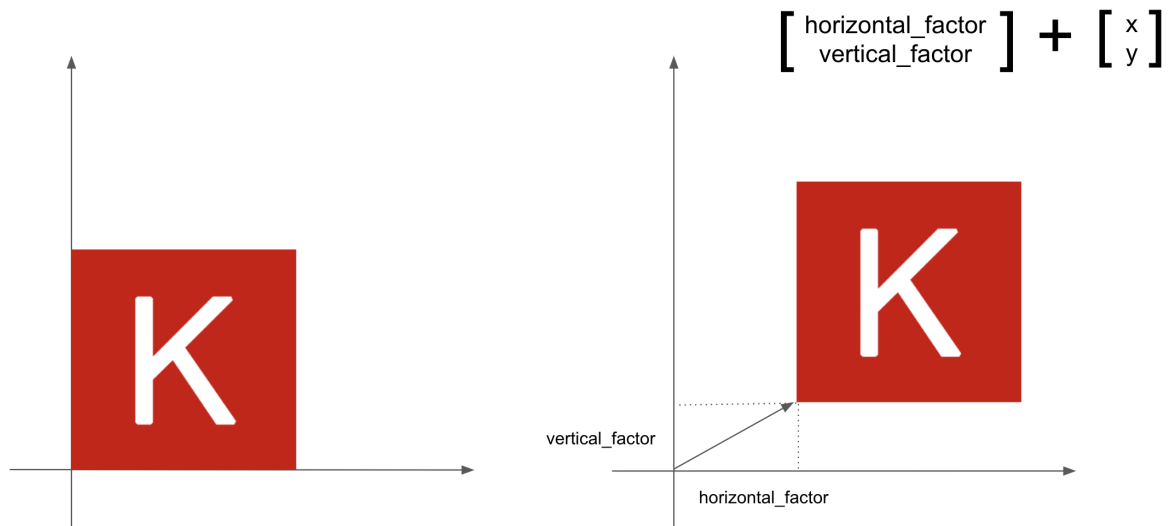


## 유니버설 함수

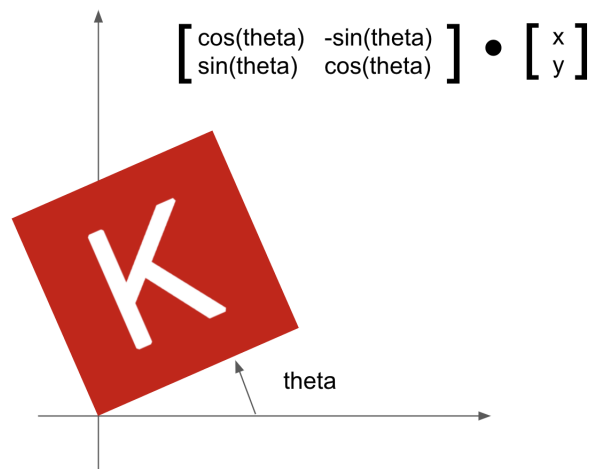
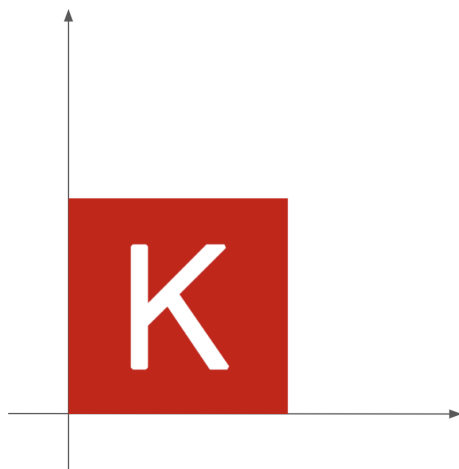


텐서 연산의 기하학적 의미

## 이동: 벡터 합

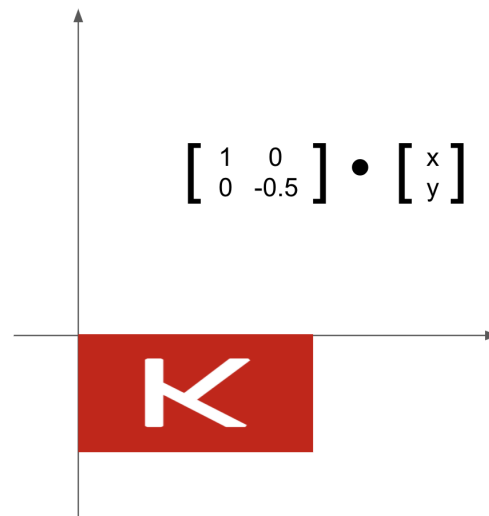
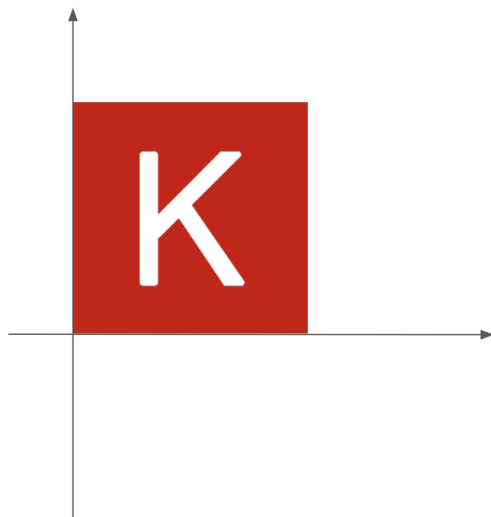


회전: 점 곱

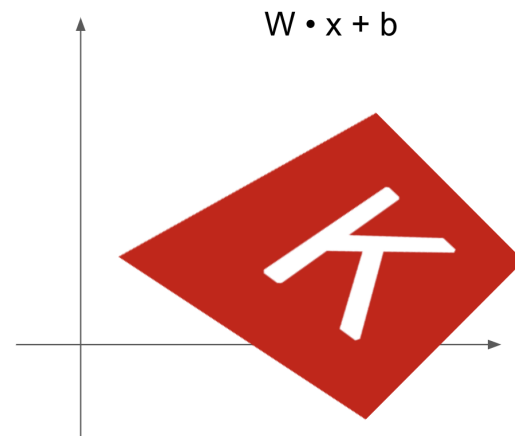
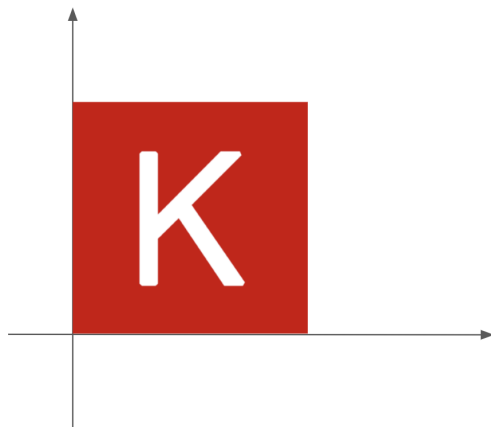




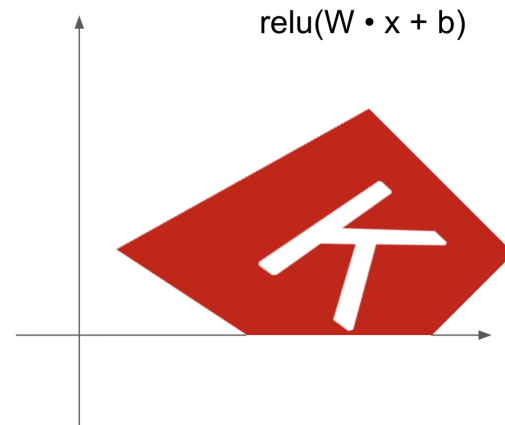
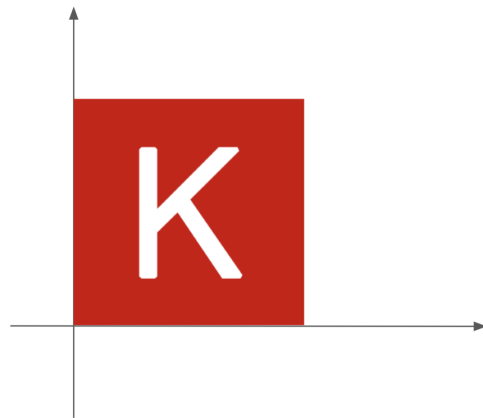
스케일링: 점 곱



## 아핀 변환



## 아핀 변환과 relu 활성화 함수



## 신경망의 텐서 연산

