

밑바닥부터 시작하는 딥러닝 2

1 ~2장 발표

2024.07.01

김태균

목차

1. 신경망

- 1.1 _____ 수학과 파이썬
- 1.2 _____ 신경망의 학습과 추론

2. 자연어와 단어의 분산 표현

- 2.1 _____ 자연어 처리란
- 2.2 _____ 자연어 처리 기법
- 2.3 _____ 특잇값 분해(SVD)
- 2.4 _____ 정리

1. 신경망

1.1 수학과 파이썬

복습

벡터

: 1차원 배열



vector

행렬

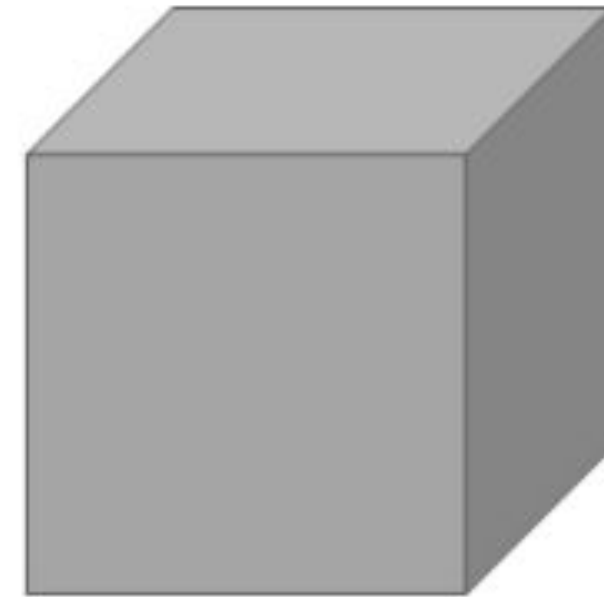
: 2차원 배열



matrix

텐서

: 벡터와 행렬을 확장하여
N차원으로 표현한 것



3d-tensor

1.1 수학과 파이썬

복습

행렬

- **형상 확인**

그림 1-6 형상 확인: 행렬의 곱에서는 대응하는 차원의 원소 수를 일치시킨다.



1.1 수학과 파이썬

복습

행렬

연산

1. 원소별 연산

```
1 import numpy as np
2
3 W = np.array([[1, 2, 3], [4, 5, 6]])
4 X = np.array([[0, 1, 2], [3, 4, 5]])
5 b = np.array([-2, -3, -4])
6
7 # 1. 원소별 연산
8 print(W + X)
9 print(W * X)
```

```
[[ 1  3  5]
 [ 7  9 11]]
[[ 0  2  6]
 [12 20 30]]
```

1.1 수학과 파이썬

복습

행렬

연산

2. 형상이 다른 배열끼리의 연산

```
1 import numpy as np
2
3 W = np.array([[1, 2, 3], [4, 5, 6]])
4 X = np.array([[0, 1, 2], [3, 4, 5]])
5 b = np.array([-2, -3, -4])
6
7 # 2. 형상이 다른 배열끼리의 연산(브로드 캐스트)
8 print(W * (-1))
9 print(W * b)
```

```
[[ -1  -2  -3]
 [-4  -5  -6]]
[[ -2  -6 -12]
 [-8 -15 -24]]
```


1.1 수학과 파이썬

복습

행렬

연산
3. 행렬의 곱

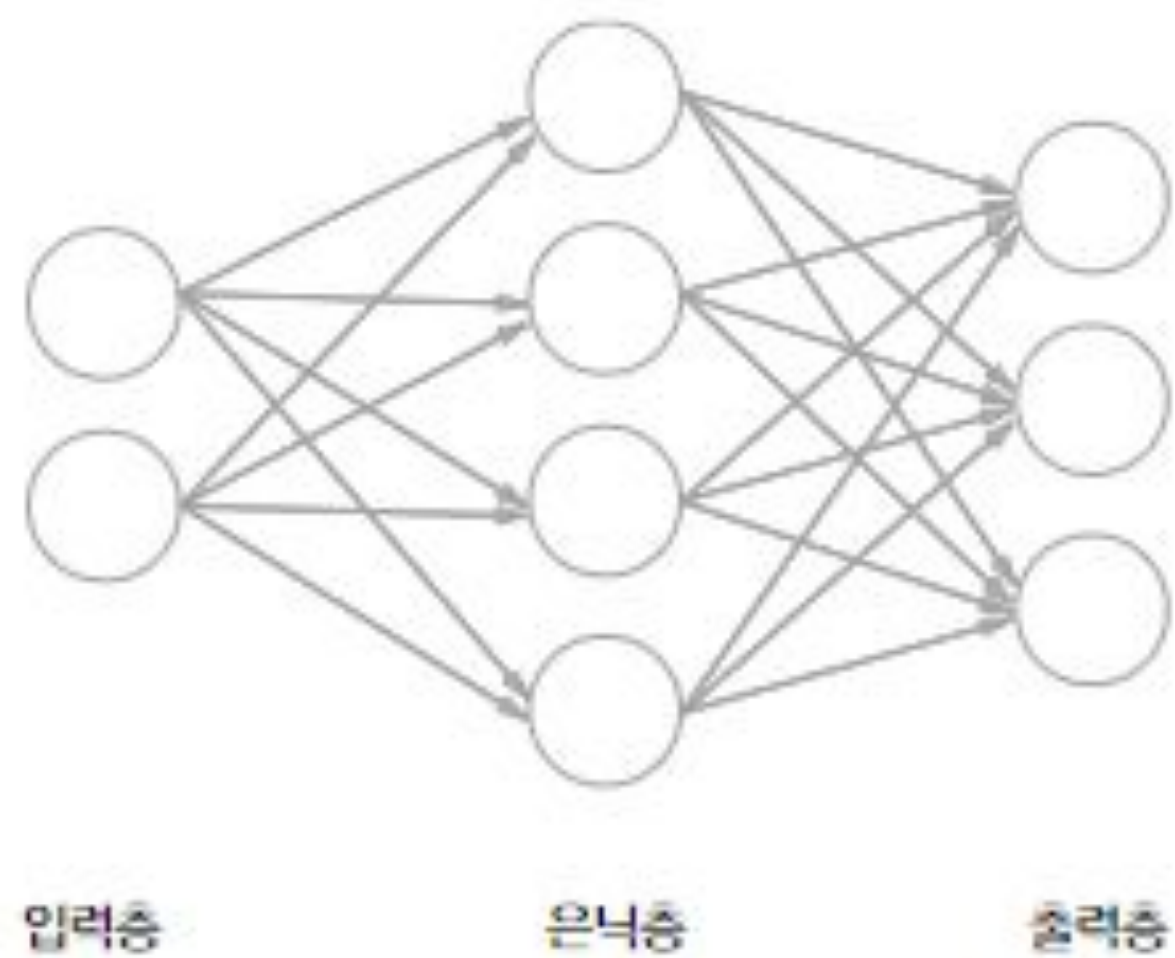
```
1  import numpy as np
2
3  # 벡터의 내적
4  a = np.array([1, 2, 3])
5  b = np.array([4, 5, 6])
6  print(f'내적 : {np.dot(a, b)}')
7
8  # 행렬의 곱
9  A = np.array([[1, 2], [3, 4]])
10 B = np.array([[5, 6], [7, 8]])
11 print(f'matmul: \n{np.matmul(A, B)}')
12 print(f'dot: \n{np.dot(A, B)}')
```

```
내적 : 32
matmul:
[[19 22]
 [43 50]]
dot:
[[19 22]
 [43 50]]
```


1.2 신경망의 학습과 추론

구조

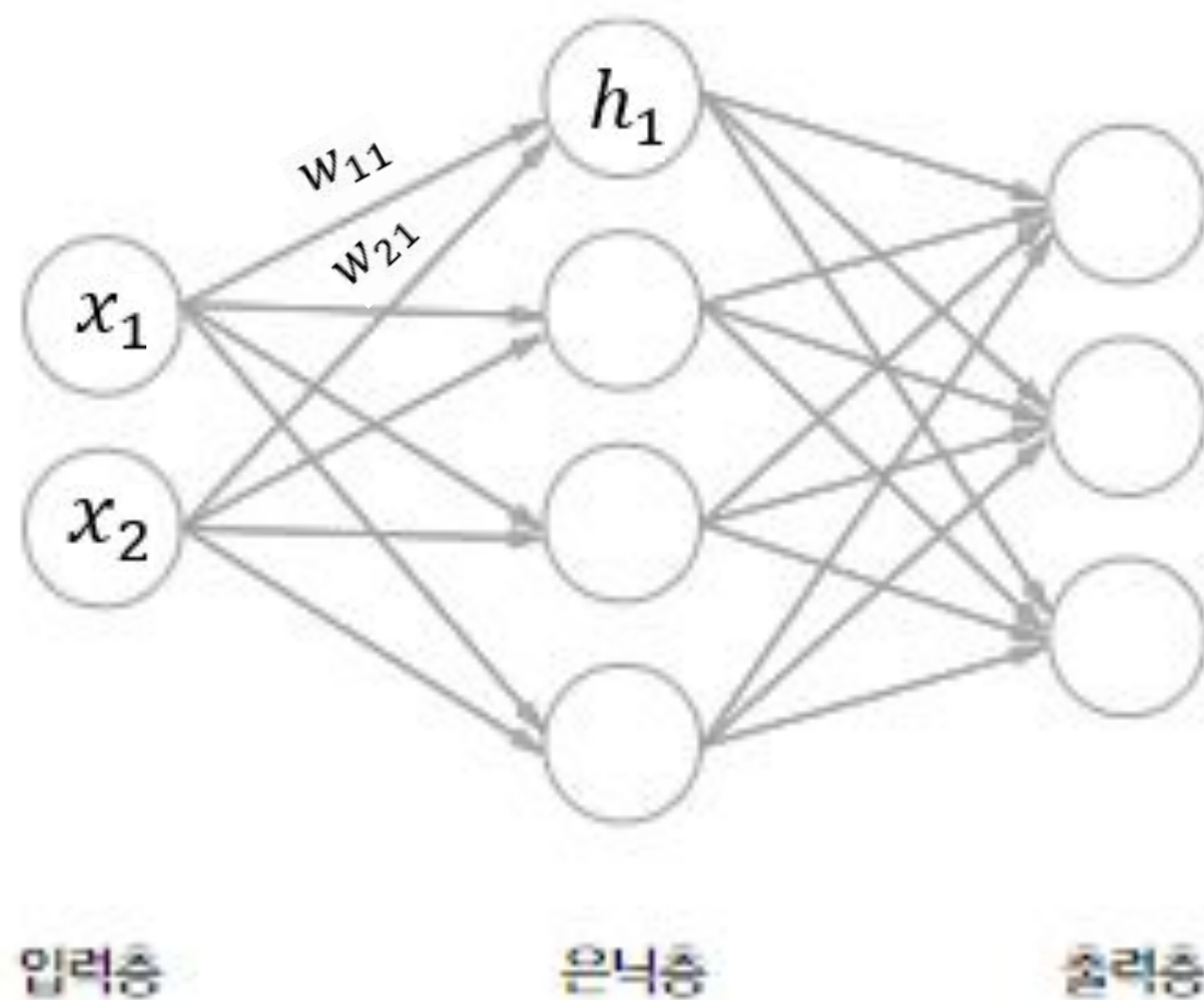
그림 1-7 신경망의 예



1.2 신경망의 학습과 추론

구조

그림 1-7 신경망의 예



$$(h_1, h_2, h_3, h_4) = (x_1, x_2) \begin{pmatrix} w_{11} & w_{12} & w_{13} & w_{14} \\ w_{21} & w_{22} & w_{23} & w_{24} \end{pmatrix} + (b_1, b_2, b_3, b_4) \quad [\text{식 1.3}]$$

$$\mathbf{h} = \mathbf{xW} + \mathbf{b}$$

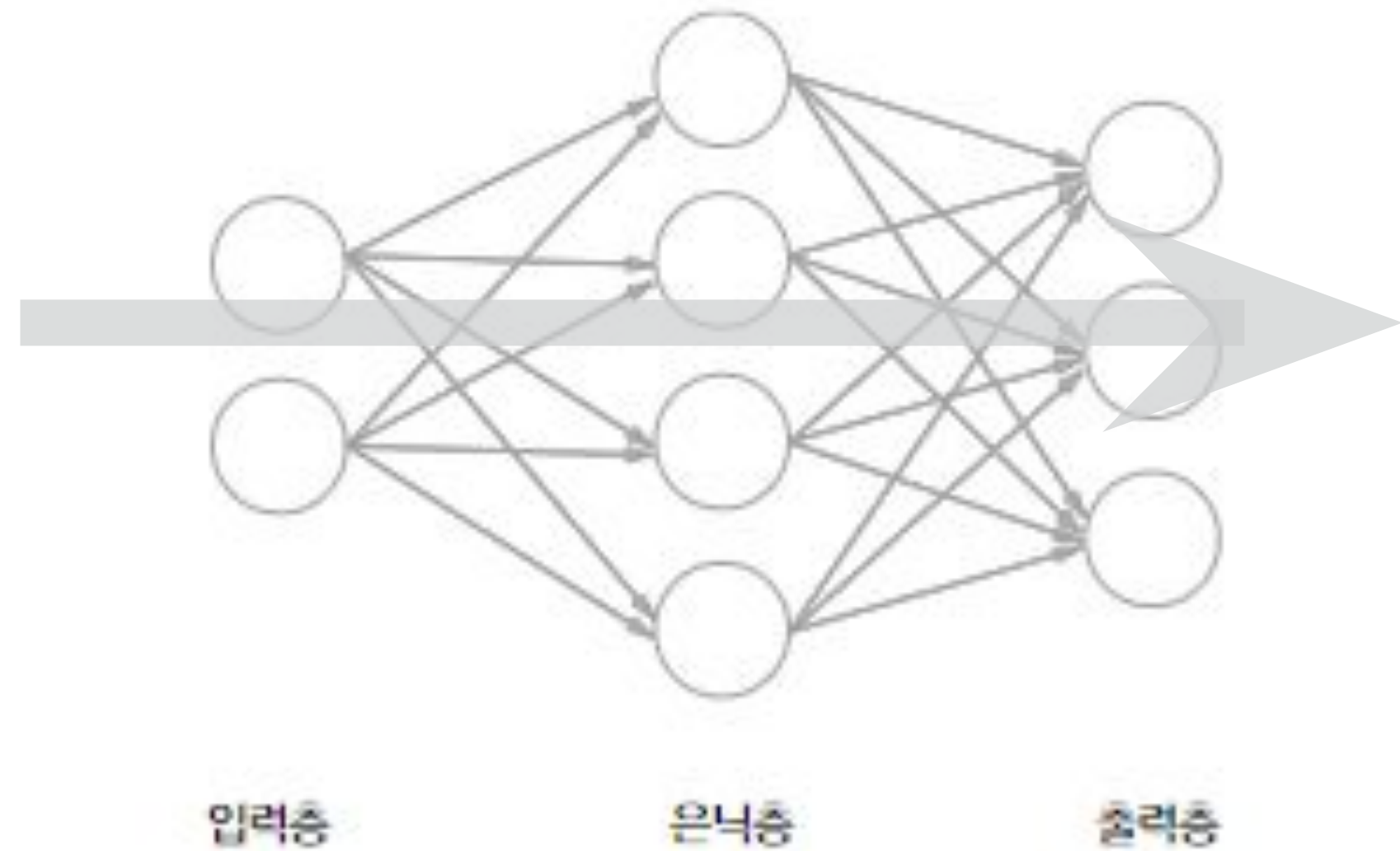
[식 1.4]

1.2 신경망의 학습과 추론

추론

- 순전파

그림 1-7 신경망의 예

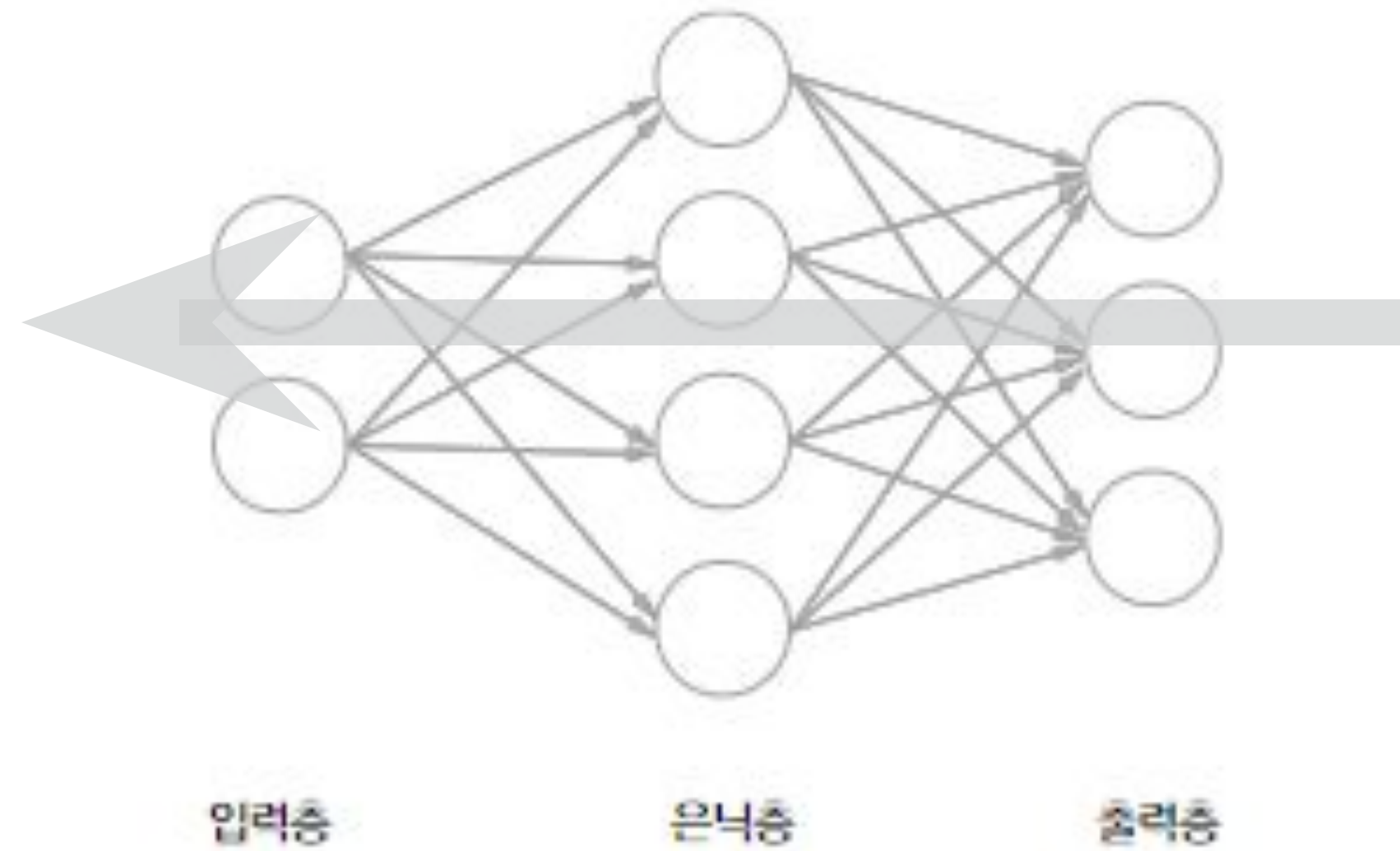


1.2 신경망의 학습과 추론

학습

- 역전파

그림 1-7 신경망의 예

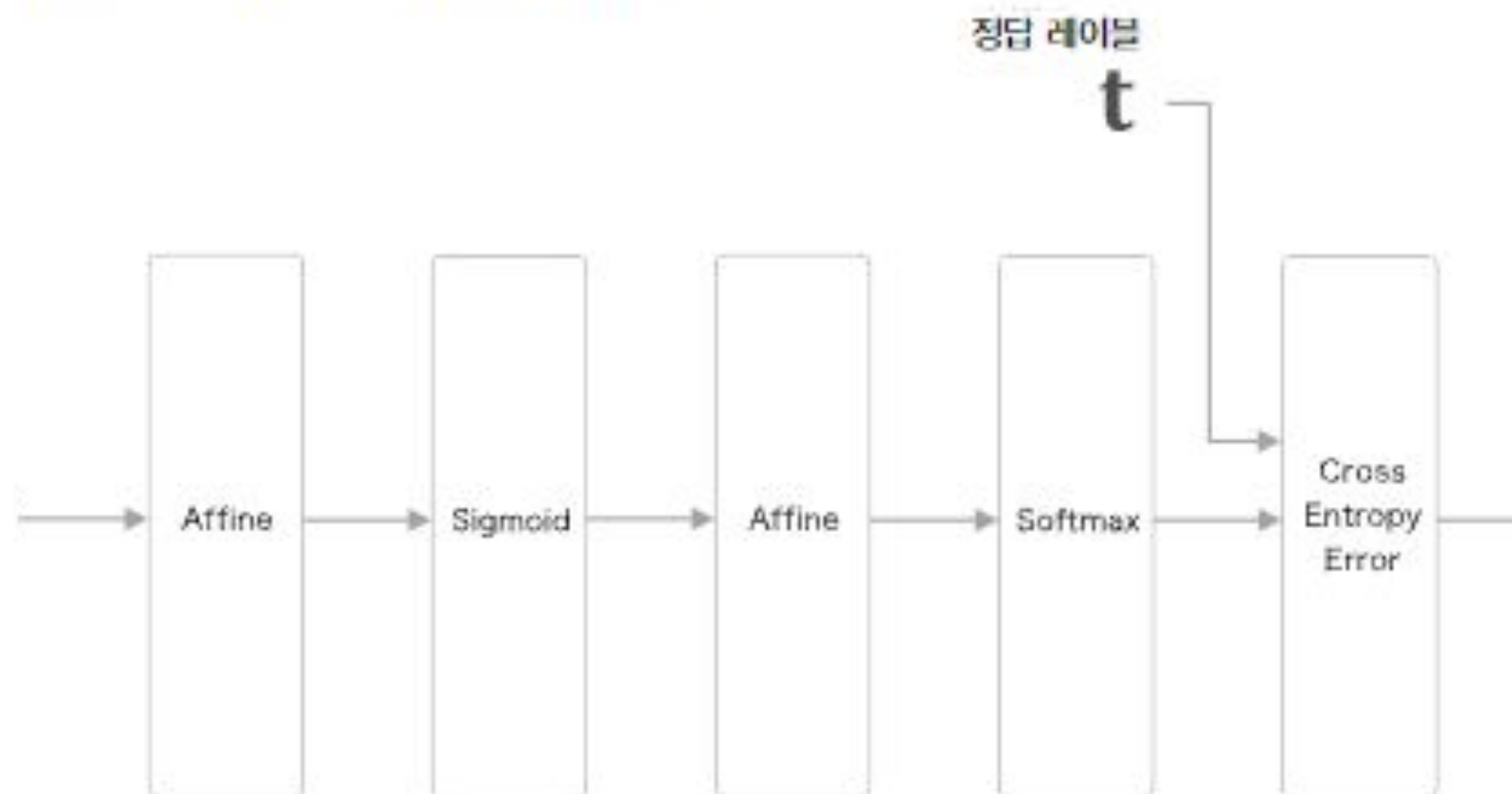


1.2 신경망의 학습과 추론

학습

- 손실 함수

1-12 손실 함수를 적용한 신경망의 계층 구성

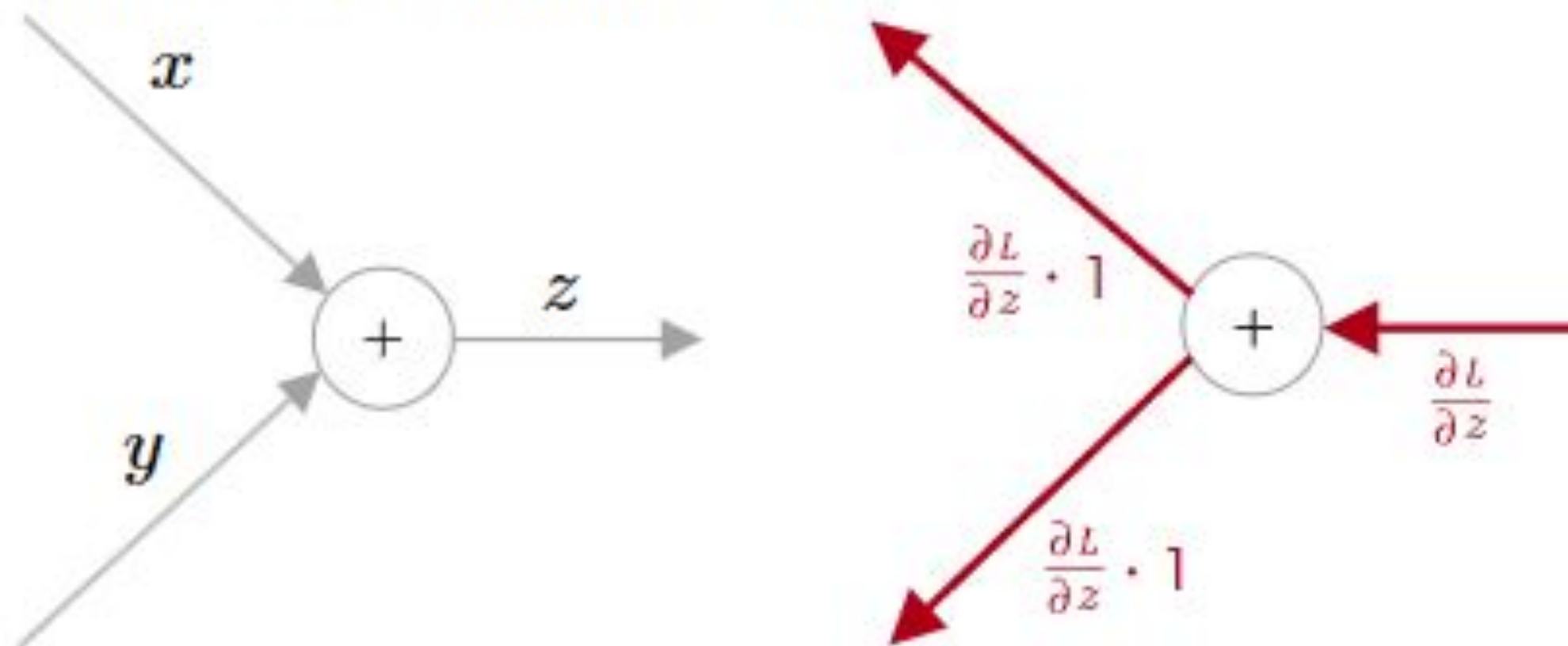


1.2 신경망의 학습과 추론

학습

- 계산그래프
 - 덧셈 노드

그림 1-18 덧셈 노드의 순전파(왼쪽)와 역전파(오른쪽)

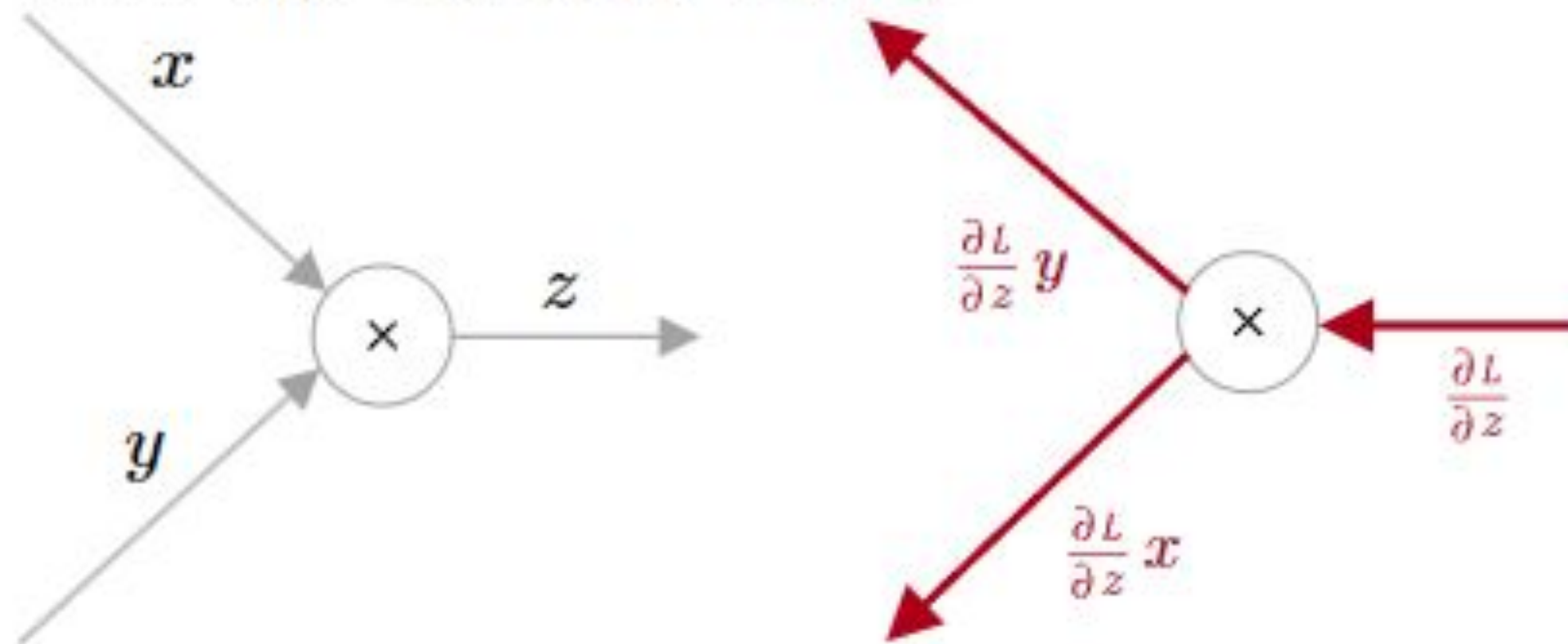


1.2 신경망의 학습과 추론

학습

- 계산그래프
 - 곱셈 노드

그림 1-19 곱셈 노드의 순전파(왼쪽)와 역전파(오른쪽)

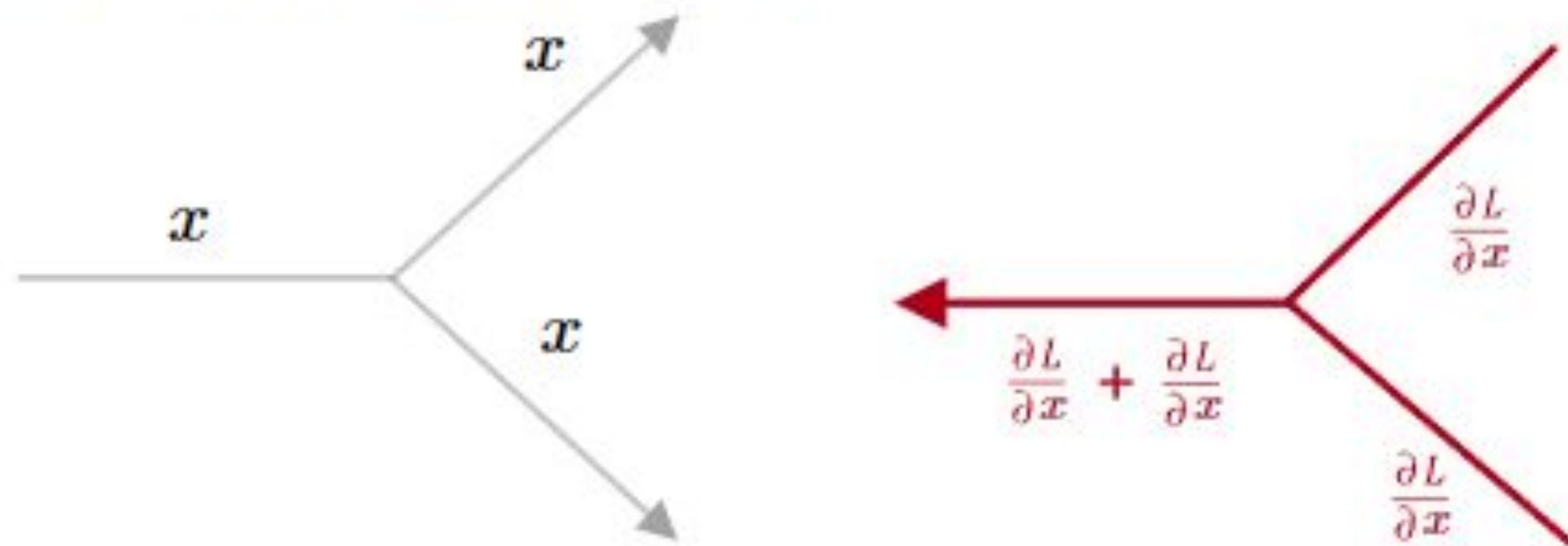


1.2 신경망의 학습과 추론

학습

- 계산그래프
 - 분기 노드

그림 1-20 분기 노드의 순전파(왼쪽)와 역전파(오른쪽)

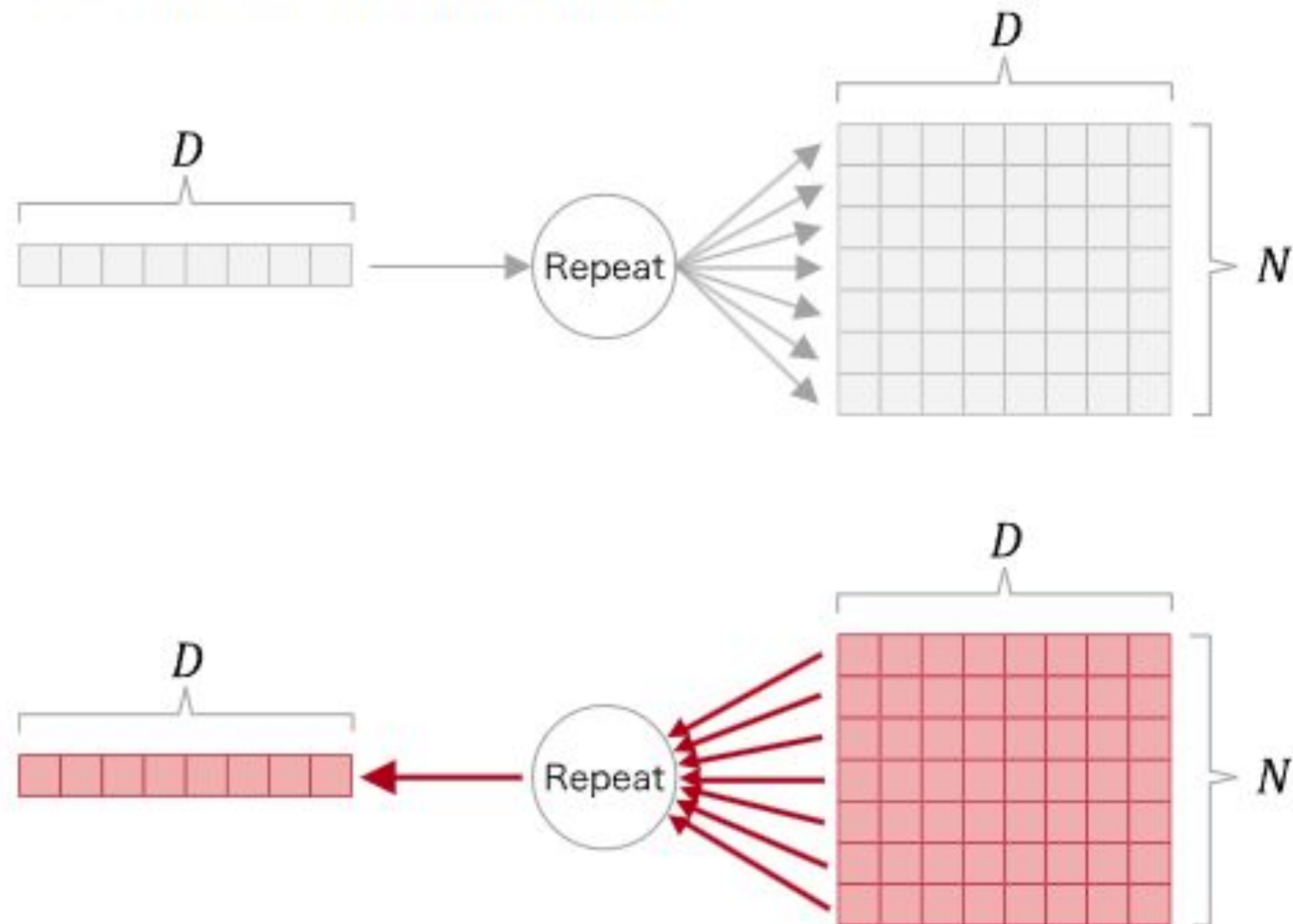


1.2 신경망의 학습과 추론

학습

- 계산그래프
 - Repeat노드

그림 1-21 Repeat 노드의 순전파(위)와 역전파(아래)

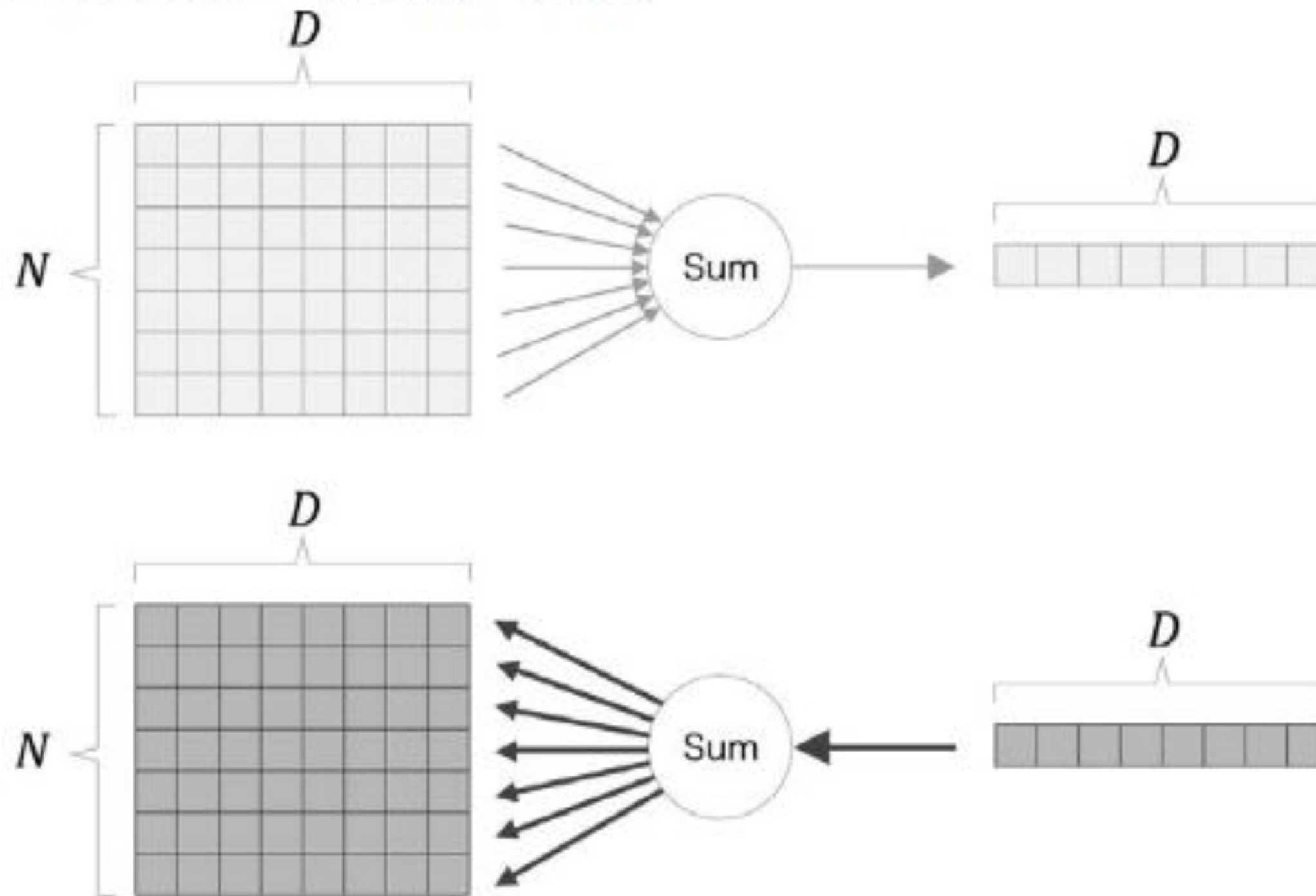


1.2 신경망의 학습과 추론

학습

- 계산그래프
 - Sum 노드

그림 1-22 Sum 노드의 순전파(위)와 역전파(아래)

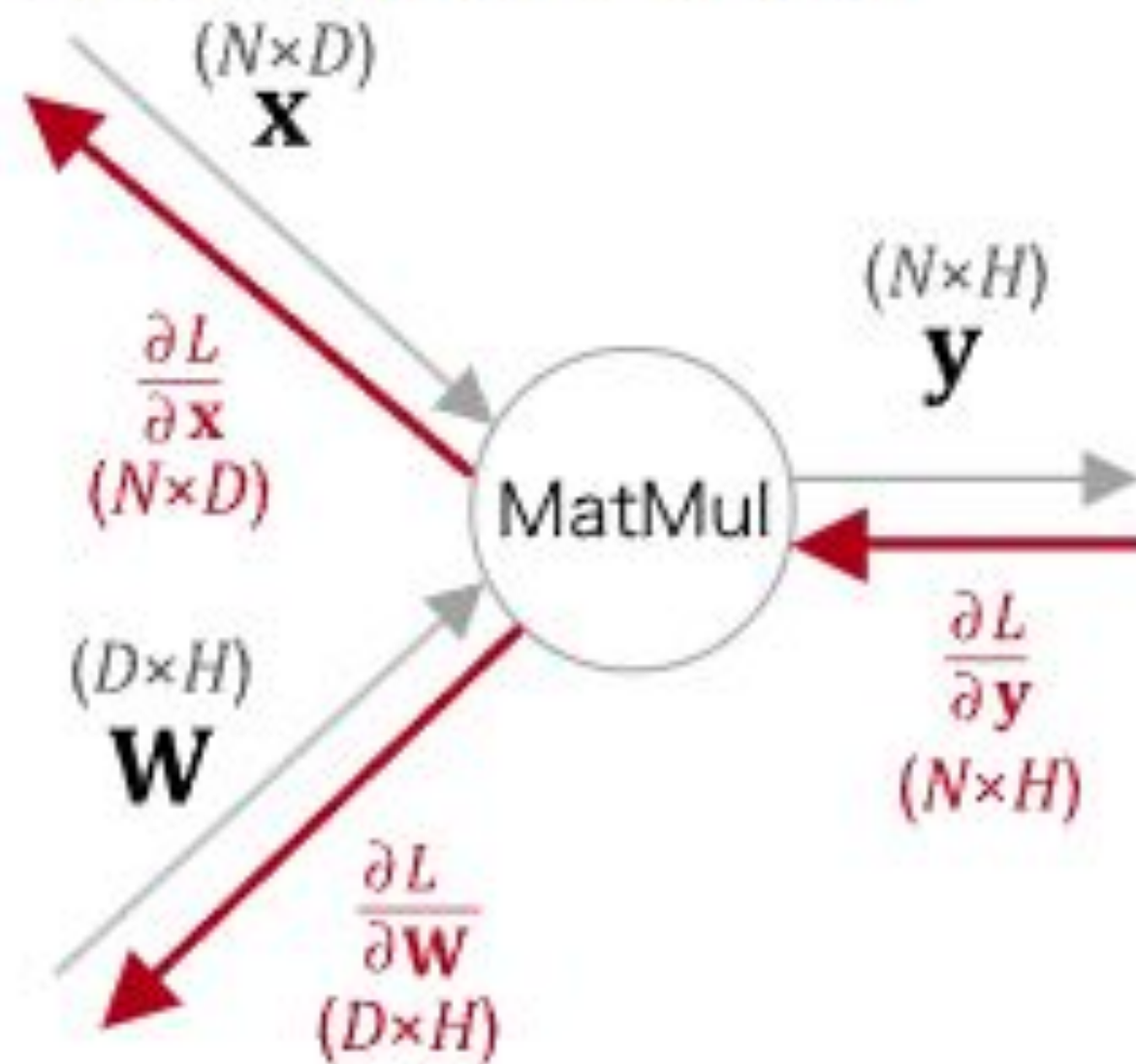


1.2 신경망의 학습과 추론

학습

- 계산그래프
 - MatMul 노드

그림 1-25 MatMul 노드의 역전파

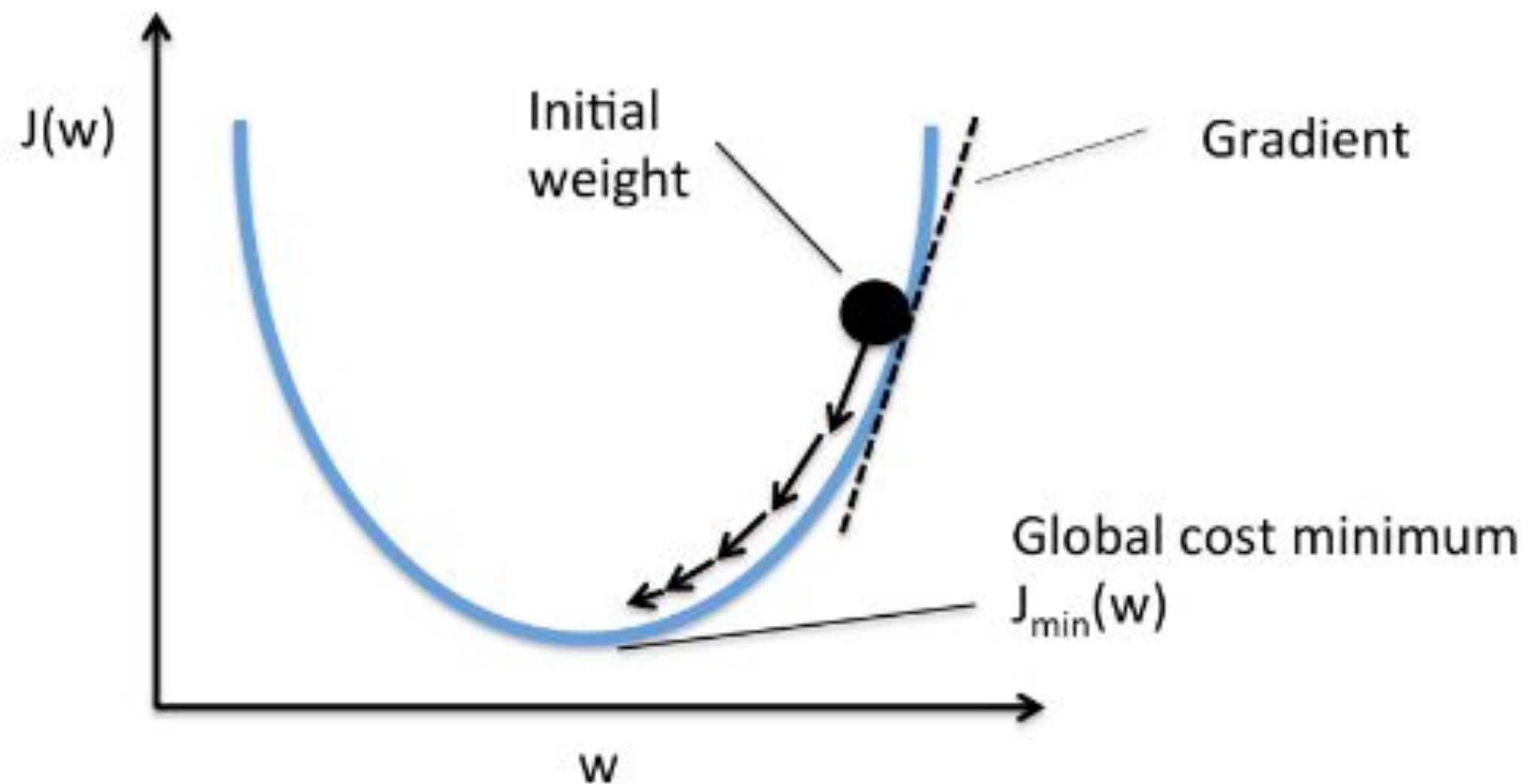


1.2 신경망의 학습과 추론

학습

- 가중치 갱신
 - SGD

$$\mathbf{W} \leftarrow \mathbf{W} - \eta \frac{\partial L}{\partial \mathbf{W}}$$



1.2 신경망의 학습과 추론

학습

- 전체 과정

- **1단계: 미니배치**

훈련 데이터 중에서 무작위로 다수의 데이터를 골라낸다.

- **2단계: 기울기 계산**

오차역전파법으로 각 가중치 매개변수에 대한 손실 함수의 기울기를 구한다.

- **3단계: 매개변수 갱신**

기울기를 사용하여 가중치 매개변수를 갱신한다.

- **4단계: 반복**

1~3단계를 필요한 만큼 반복한다.

2. 자연어와 단어의 분산

표현

2.1 자연어 처리란

자연어 처리

- 컴퓨터에게 단어를 어떻게 이해시킬까?



2.2 자연어 처리 기법

자연어 처리 기법

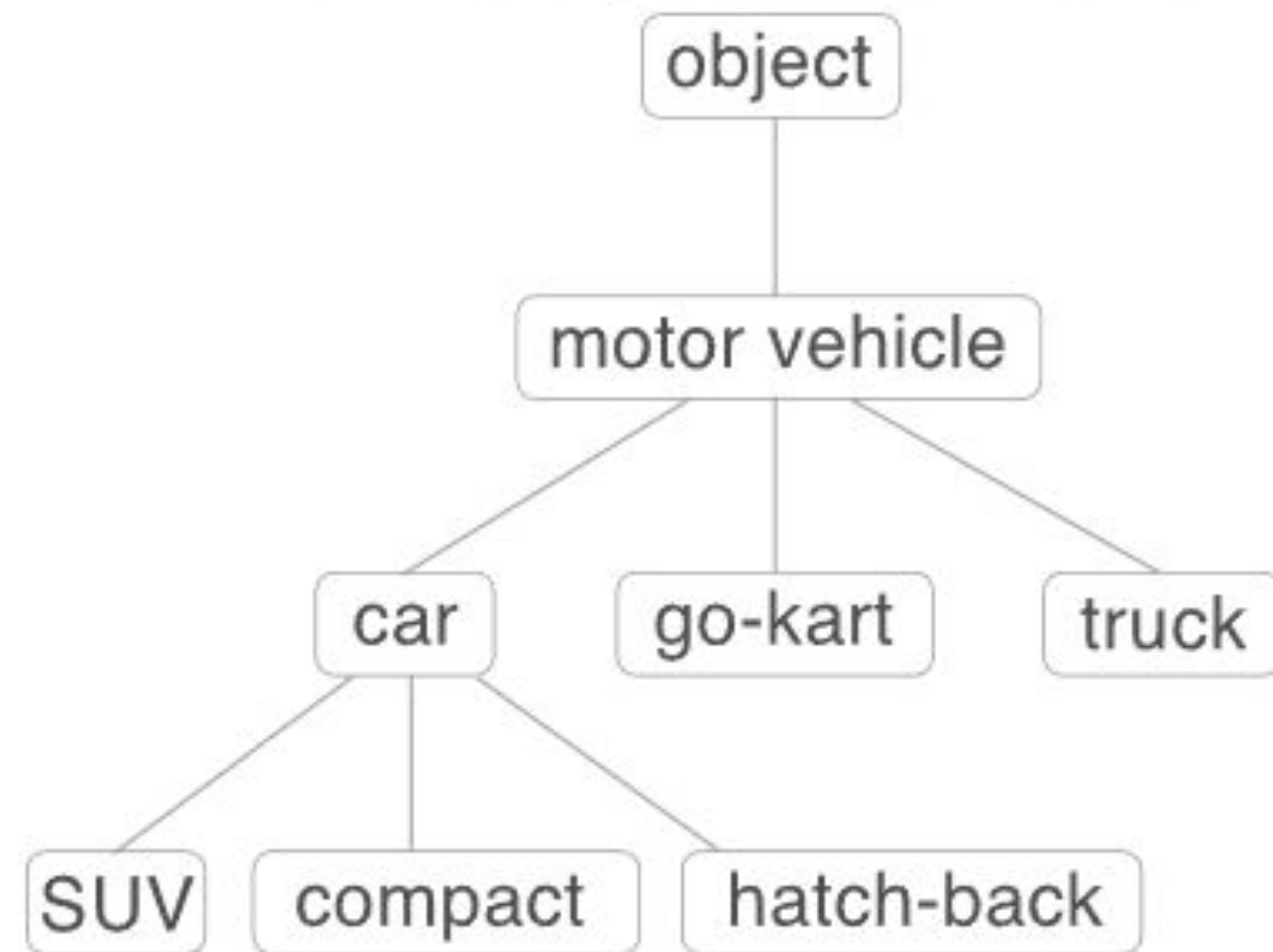
1. 시소러스
2. 통계 기반 기법
- ~~3. 추론 기반 기법~~

2.2 자연어 처리 기법

자연어 처리 기법

1. 시소러스

그림 2-2 단어들을 의미의 상·하위 관계에 기초해 그래프로 표현한다(문헌 [14]를 참고하여 그림).



2.2 자연어 처리 기법

자연어 처리 기법

2. 통계 기반 기법

- 말뭉치(corpus)

text = 'You say goodbye and I say hello.'

2.2 자연어 처리 기법

자연어 처리 기법

2. 통계 기반 기법

- 말뭉치(corpus)

```
1 text = 'You say goodbye and I say hello.'
2 text = text.lower()
3 text = text.replace('.', ' .')
4 words = text.split()
5 print(words)
```

```
['you', 'say', 'goodbye', 'and', 'i', 'say', 'hello', '.']
```

```
10 for word in words:
11     if word not in word_to_id:
12         new_id = len(word_to_id)
13         word_to_id[word] = new_id
14         id_to_word[new_id] = word
15 print(word_to_id)
```

```
{'you': 0, 'say': 1, 'goodbye': 2, 'and': 3, 'i': 4, 'hello': 5, '.': 6}
```

2.2 자연어 처리 기법

자연어 처리 기법

2. 통계 기반 기법

- 말뭉치(corpus)

```
19 corpus = [word_to_id[w] for w in words]
20 corpus = np.array(corpus)
21 print(corpus)
```

```
[0 1 2 3 4 1 5 6]
```

2.2 자연어 처리 기법

자연어 처리 기법

2. 통계 기반 기법

- 단어의 분산 표현

그림 2-6 단어 "say"의 맥락에 포함되는 단어의 빈도를 표로 정리한다.

you say goodbye and i say hello .

	you	say	goodbye	and	i	hello	.
say	1	0	1	0	1	1	0

2.2 자연어 처리 기법

자연어 처리 기법

2. 통계 기반 기법

- 분포 가설

그림 2-3 윈도우 크기가 2인 '맥락'의 예. 단어 "goodbye"에 주목한다면, 그 좌우의 두 단어(총 네 단어)를 맥락으로 이용한다.

you say goodbye and i say hello.



2.2 자연어 처리 기법

자연어 처리 기법

2. 통계 기반 기법

- 분포 가설
 - 동시 발생 행렬

그림 2-7 모든 단어 각각의 맥락에 해당하는 단어의 빈도를 세어 표로 정리한다.

	you	say	goodbye	and	i	hello	.
you	0	1	0	0	0	0	0
say	1	0	1	0	1	1	0
goodbye	0	1	0	1	0	0	0
and	0	0	1	0	1	0	0
i	0	1	0	1	0	0	0
hello	0	1	0	0	0	0	1
.	0	0	0	0	0	1	0

2.2 자연어 처리 기법

자연어 처리 기법

2. 통계 기반 기법

- 분포 가설
 - 벡터 간 유사도

$$\text{similarity}(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|} = \frac{x_1 y_1 + \cdots + x_n y_n}{\sqrt{x_1^2 + \cdots + x_n^2} \sqrt{y_1^2 + \cdots + y_n^2}}$$

[식 2.1]

2.2 자연어 처리 기법

자연어 처리 기법

2. 통계 기반 기법

- 분포 가설
 - 점별 상호정보량
 - 양의 상호정보량

$$\text{PMI}(x, y) = \log_2 \frac{P(x, y)}{P(x)P(y)} = \log_2 \frac{\frac{C(x, y)}{N}}{\frac{C(x)}{N} \frac{C(y)}{N}} = \log_2 \frac{C(x, y) \cdot N}{C(x)C(y)} \quad [\text{식 2.3}]$$

$$\text{PPMI}(x, y) = \max(0, \text{PMI}(x, y)) \quad [\text{식 2.6}]$$

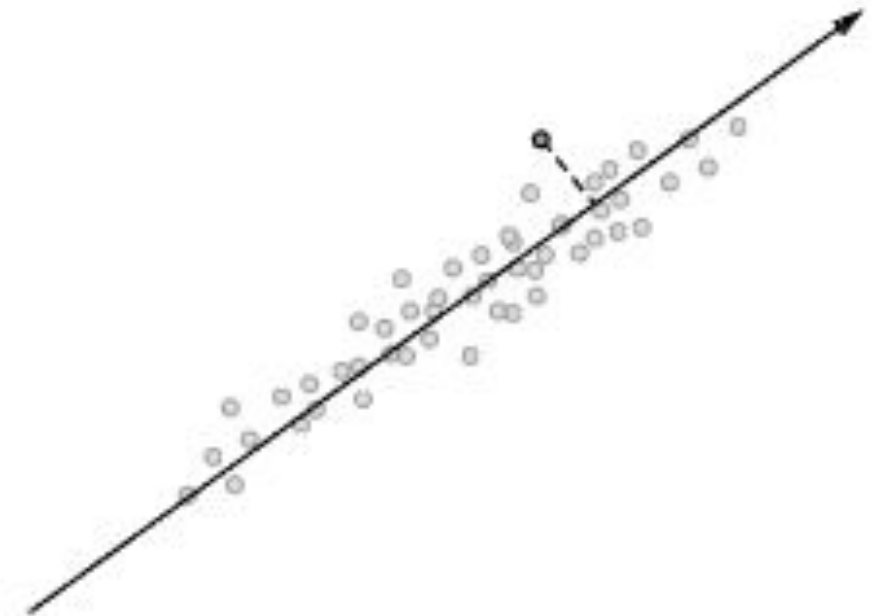
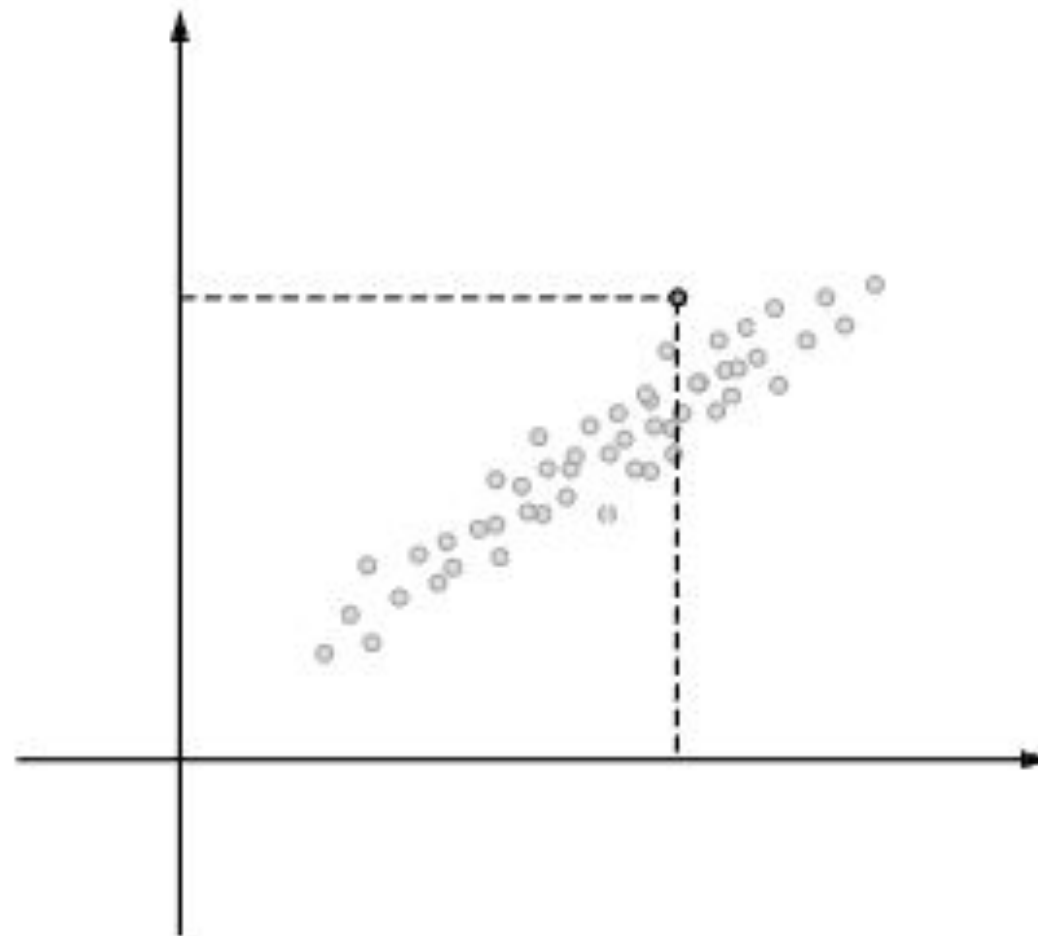
2.2 자연어 처리 기법

자연어 처리 기법

2. 통계 기반 기법

- 분포 가설
 - 차원 감소

그림 2-8 그림으로 이해하는 차원 감소: 2차원 데이터를 1차원으로 표현하기 위해 중요한 축(데이터를 넓게 분포시키는 축)을 찾는다.



2.3 특잇값 분해(SVD)

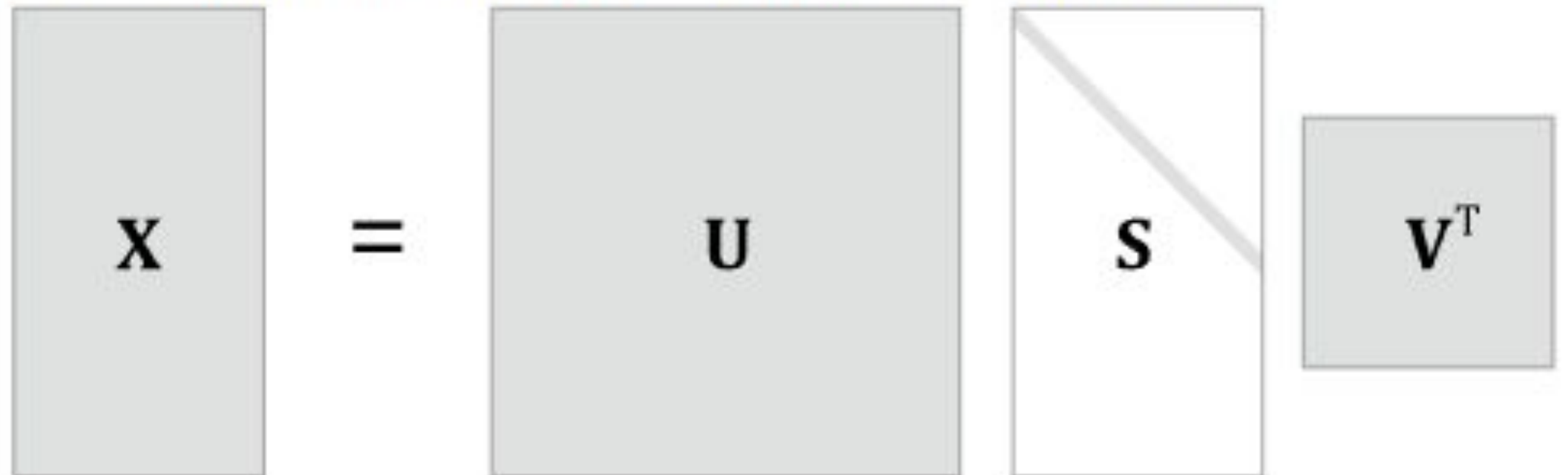
SVD

- 원리

$$X = USV^T$$

[식 2.7]

그림 2-9 SVD에 의한 행렬의 변환(행렬의 '흰 부분'은 원소가 0임을 뜻함)



2.3 특잇값 분해(SVD)

SVD

- 원리

$$A = \begin{bmatrix} | & | & | & \cdots & | & \cdots & | \\ u_1 & u_2 & u_3 & \cdots & u_M & \cdots & u_N \\ | & | & | & \cdots & | & \cdots & | \end{bmatrix} \begin{bmatrix} \boxed{\sigma_1} & 0 & 0 & \cdots & 0 \\ 0 & \boxed{\sigma_2} & 0 & \cdots & 0 \\ 0 & 0 & \boxed{\sigma_3} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \boxed{\sigma_M} \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} \boxed{v_1^T} \\ \boxed{v_2^T} \\ \vdots \\ \boxed{v_M^T} \end{bmatrix}$$

$$A = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \cdots + \sigma_M u_M v_M^T$$

2.3 특잇값 분해(SVD)

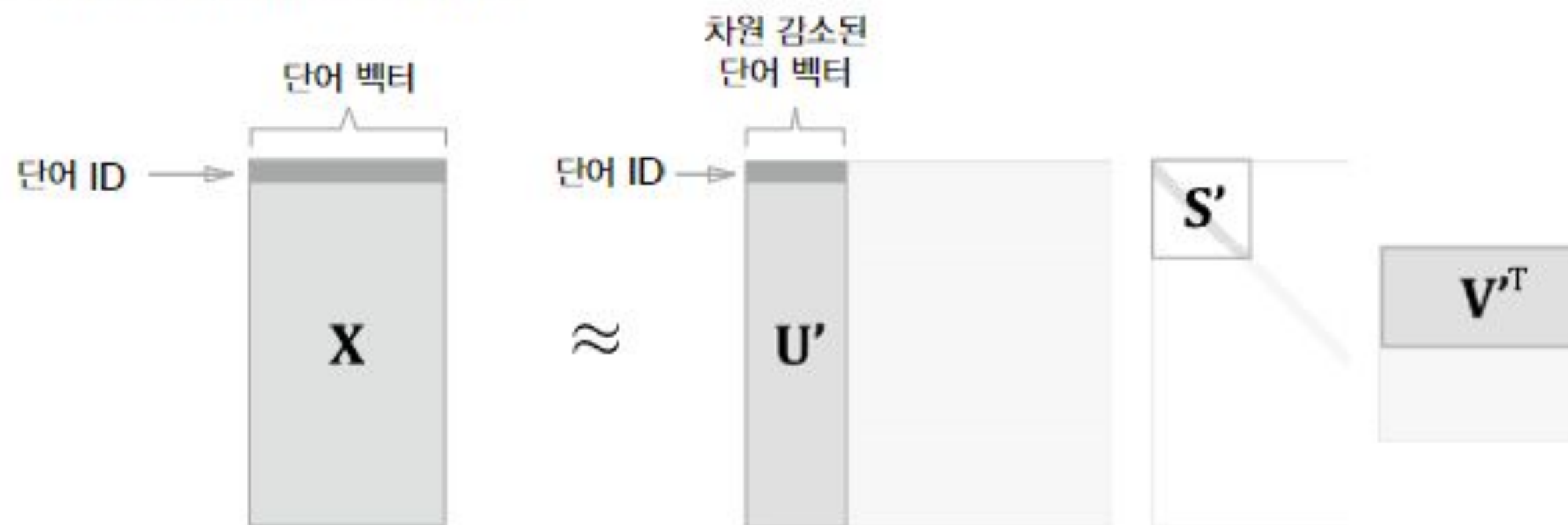
SVD

- 원리

$$X = USV^T$$

[식 2.7]

그림 2-10 SVD에 의한 차원 감소



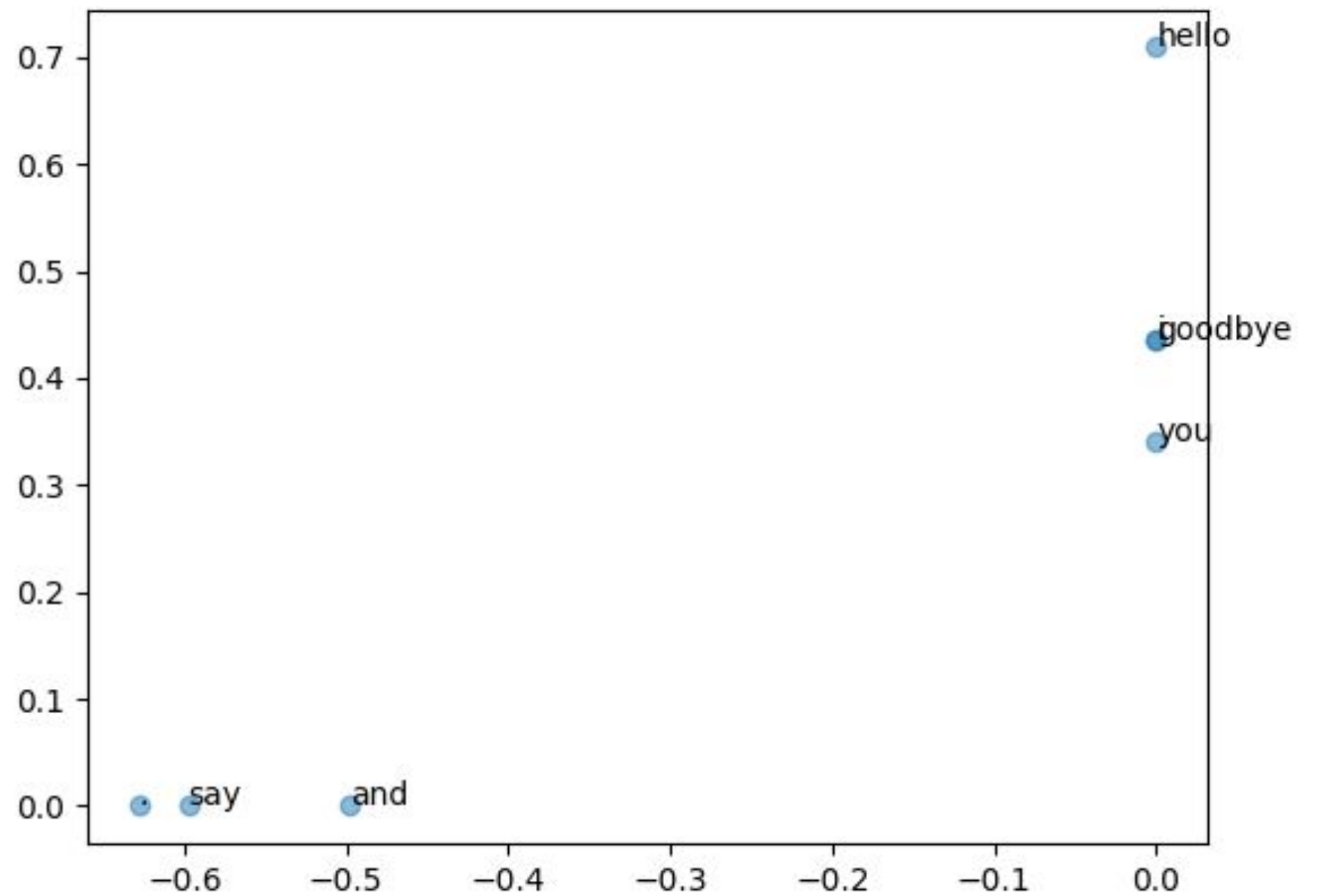
$$X = USV^T \approx U' S' V'^T$$

2.3 특잇값 분해(SVD)

SVD

- 시각화

text = 'You say goodbye and I say hello.'



2.3 특잇값 분해(SVD)

SVD

- PTB 데이터셋 활용

```
[query] you
i: 0.6605389714241028
'd: 0.5774868130683899
we: 0.5752034187316895
do: 0.5684781074523926
'11: 0.5566216707229614

[query] year
last: 0.6321343779563904
month: 0.62895268201828
earlier: 0.6102904081344604
next: 0.6037287712097168
quarter: 0.5836564898490906

[query] car
auto: 0.6354261636734009
luxury: 0.5820189118385315
vehicle: 0.5271400809288025
cars: 0.5269667506217957
automobiles: 0.5156604051589966

[query] toyota
motor: 0.7507732510566711
nissan: 0.6563388705253601
motors: 0.6372733116149902
lexus: 0.6046193838119507
honda: 0.6033986210823059
```

2.4 정리

신경망

- 신경망의 구성
- 신경망의 역할

자연어

- 자연어 처리의 정의
- 자연어 처리 기법
- SVD

QnA