

PART 1

단층 퍼셉트론(SLP)

1장 회귀분석

2장 이진판단

3장 선택분류

딥러닝 & 강화학습 담당
이재화 강사

이 장에서 다룰 내용

1. 단층 퍼셉트론 신경망 구조
2. 텐서 연산, 미니배치, 하이퍼파라미터, One-hot 벡터 표현
3. 신경망의 세 가지 기본 출력 유형
4. 회귀 분석, 평균제곱오차(MSE) 손실함수
5. 경사하강법, 역전파, 편미분, 손실 기울기
6. 전복 고리 수 추정 신경망 구현과 실행

실습에 사용할 데이터셋 : Abalone Dataset

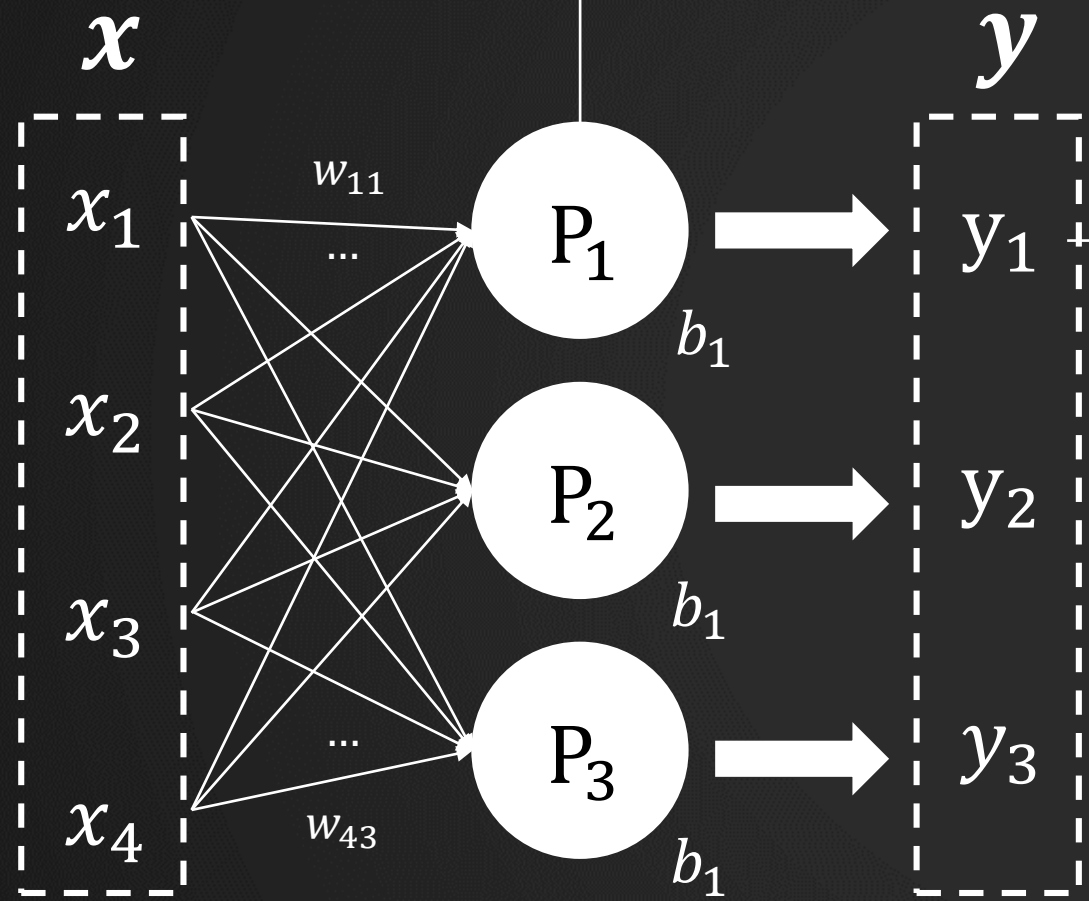
접근 방법 : Google ▶ 'abalone dataset' 검색 ▶ kaggle에서 다운로드

1.1 단층 퍼셉트론 신경망 구조

이용

$W_{11}, W_{21}, W_{31}, W_{41}$ b_1

$$y_1 = x_1w_{11} + x_2w_{21} + x_3w_{31} + x_4w_{41} + b_1$$



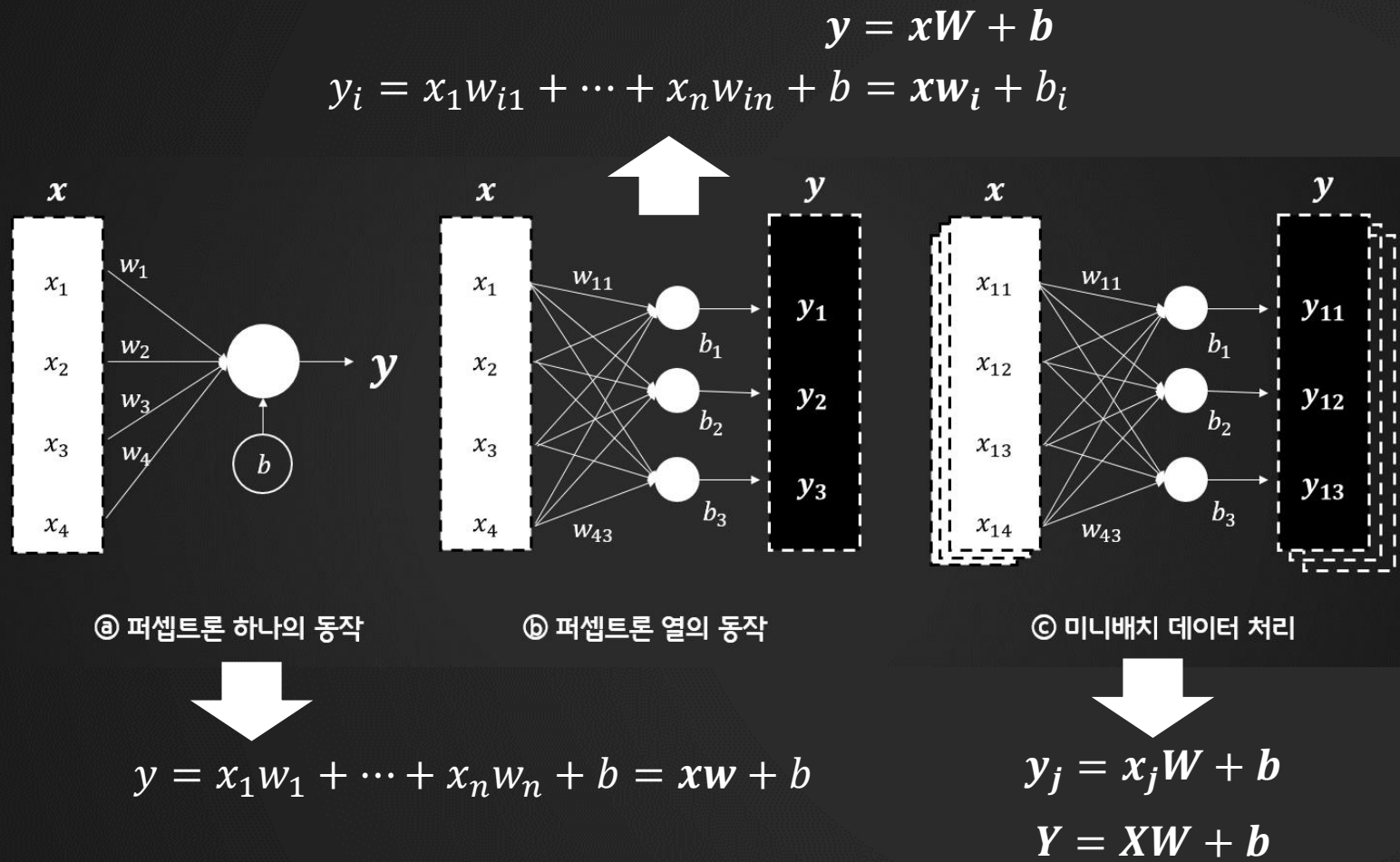
단층 퍼셉트론은

일련의 퍼셉트론을 한 줄로 배치
입력 벡터 하나로 부터 출력 벡터 하나를 얻음
입력 벡터만 공유할 뿐, 각자의 파라미터에 따라 독립적 활동
비록 간단한 구조이지만 여러 고급 신경망 구조의 기본 요소

파라미터란(parameter)?

학습 과정 중에 끊임없이 변경되어 가면서
퍼셉트론의 동작 특성을 결정하는 값들

1.2 텐서 연산과 미니배치의 활용



※ 입력 성분의 일차식으로 표현되는 과정을 선형연산, 일차식으로 나타낼 수 없는 연산을 비선형 연산 (활성화 함수)

하이퍼파라미터(hyper parameter)?

에폭 수나 미니배치 크기처럼 학습과정에서
변경되지 않으면서 신경망 구조나 학습 결과에
영향을 미치는 요인

에폭(epoch)?

학습데이터 전체에 대한 한 차례 처리를
‘에폭’이라고 부릅니다.

미니배치란(minibatch)?

딥러닝에서는 신경망이 하나의 데이터를
여러 데이터로 나눠 한꺼번에 처리하는데
이를 ‘미니배치’라 합니다.

1.3 신경망의 세 가지 기본 출력 유형과 회귀분석

넌 올해 몇 살이니?

백 두 살?
아니 만으로는 102세!

㉠ 회귀 분석

점심식사는 했니?

응! 먹었어!

㉢ 이진 판단

너무 졸린데 오늘 공부는
여기까지 할까..?

- ① 바로 잔다.
- ② 30분만 더 공부한다.
- ③ 커피를 마시고 온다.
- ④ 노래를 듣고 온다.

㉡ 선택 분류

회귀(regression)?

통계학에서는 연속형 변수 사이의 모형을
구한뒤 적합도를 측정하는 분석방법을 의미

1.4 전복의 고리 수 추정 문제

전복의 나이를 추정함으로써
다양한 정보를 유추 혹은 검증할 수 있다.

species.nibr.go.kr > rehome > redlist > redlist_view ▼

기존 적색종 종정보 - 한국의 멸종위기종

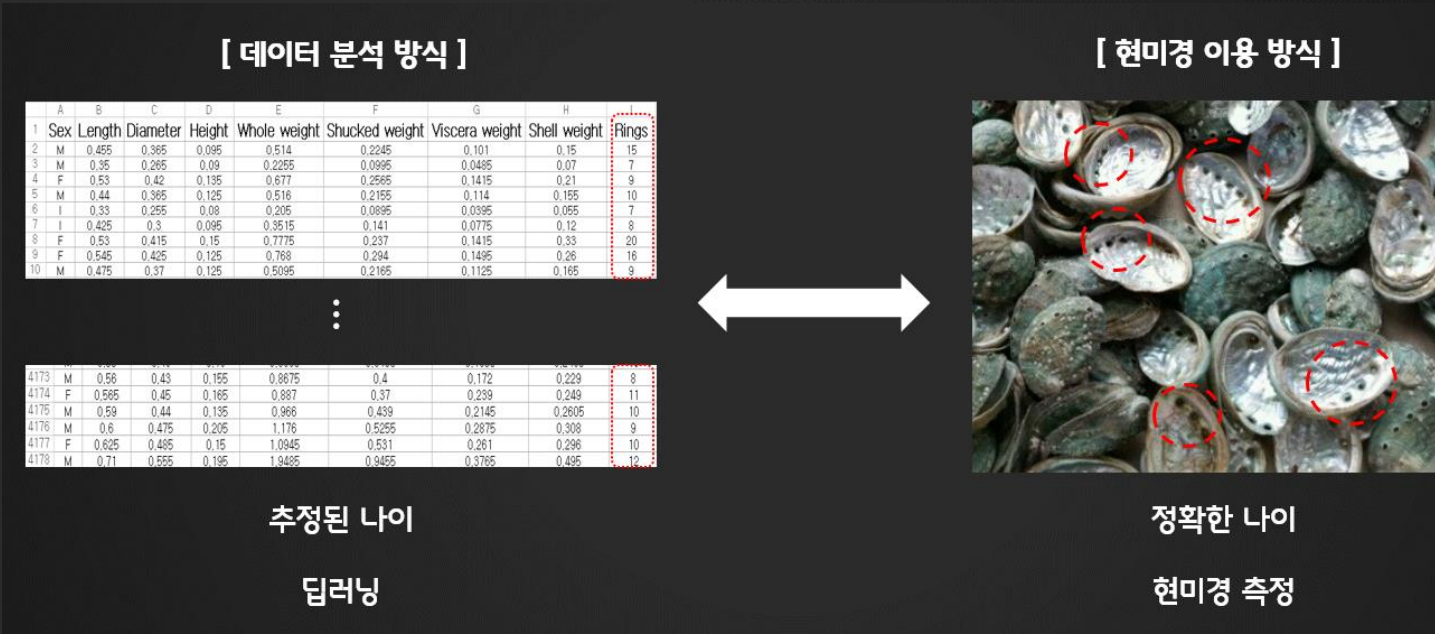
“적색 대형 전복류다. 제주도 연안에만 분포하는 제한된 서식 범위를 보인다. 서식처의

news.joins.com > article ▼

[라이프 트렌드] “세계적 명품 인정받은 완도 전복, 해외 소비자 ...

2018. 9. 11. - 전복의 품질뿐 아니라 서식 환경, 인권까지 면밀히 들여다보는 것이다. 향후 참전복

을 계속 양식할 것이라 가정에서 이 인증을 반드시 받아야겠다고 ...



1.5 회귀 분석과 평균제곱오차(MSE) 손실 함수



내가 학습을 잘 했는지
평가해줘!

너의 추정값이 얼마나
정확한지 평가할 MSE 와,
널 더 성장시키기 위한 손실함수가
필요해!



평균제곱오차(mean squared error)?

출력 각 성분에 대한 추정값과 정답 사이의 차이인 오차를 제공한 뒤
모두 합해 전체 성분 수로 나눈 값

손실함수(loss function)?

딥러닝에서는 평균제곱오차값이 항상 0 이상이며 추정이 정확해질 수록
값이 작아지는 성질이 있으면서 미분도 가능한 평가지표를 정의.
이를 최소화 하기위한것이 목표, 이러한 성질을 손실함수

Part 1. 단층 퍼셉트론

1.5 회귀 분석과 평균제곱오차(MSE) 손실 함수

[표] 메달 수 예측 게임의 정답 역할을 할 2016 하계 올림픽 국가별 획득 메달 수

| 순위 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | ... |
|----|-----|----|----|-----|----|----|-----|----|-----|
| 국가 | 미국 | 영국 | 중국 | 러시아 | 독일 | 일본 | 프랑스 | 한국 | ... |
| 금 | 46 | 27 | 26 | 19 | 17 | 12 | 10 | 9 | ... |
| 은 | 37 | 23 | 18 | 17 | 10 | 8 | 18 | 3 | ... |
| 동 | 38 | 17 | 26 | 19 | 15 | 21 | 14 | 9 | ... |
| 합계 | 121 | 67 | 70 | 55 | 42 | 41 | 42 | 21 | ... |

내가 생각하는 한.중.일 의
매달 예측 수는
한국 25개 / 중국 65개 / 일본 32개
야.

ROBOT_A

제가 생각하는 한.중.일의
매달 예측 수는
한국 20개 / 중국 80개 / 일본 35개
입니다.

‘에이림’의
메달 수 예측에 대한 평균제곱오차

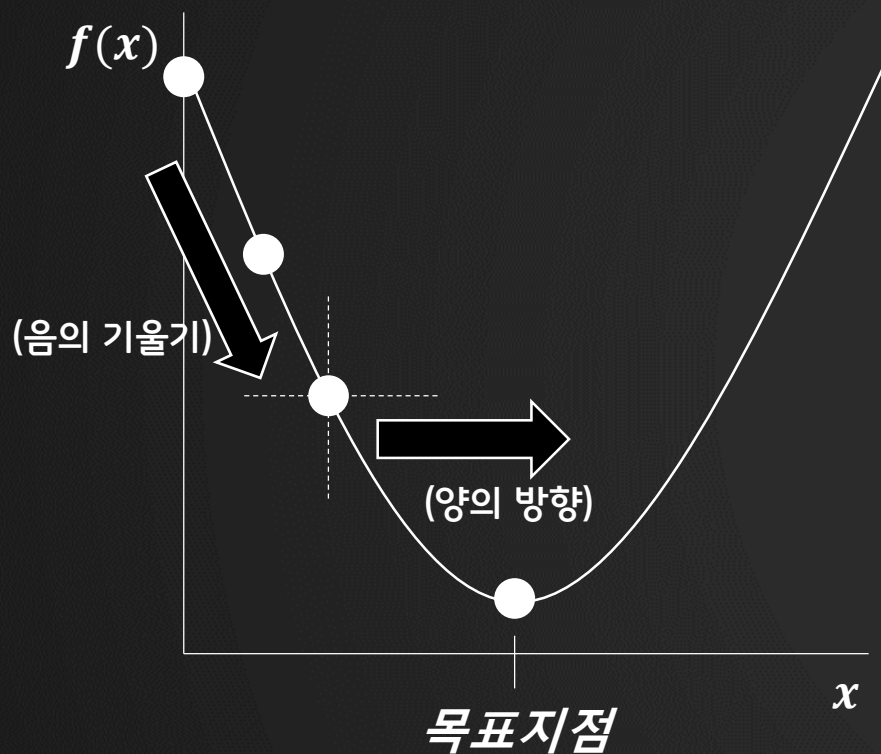
$$\frac{(25-21)^2+(65-70)^2+(32-41)^2}{3} = \frac{16+25+81}{3} = \frac{122}{3}$$



ROBOT_A의
메달 수 예측에 대한 평균제곱오차

$$\frac{(20-21)^2+(80-70)^2+(35-41)^2}{3} = \frac{1+100+36}{3} = \frac{137}{3}$$

1.6 경사하강법과 역전파



2차원 그래프로 표현한 경사하강법의 원리

경사하강법(gradient descent algorithm)?

기울기에 따라 함숫값이 낮아지는 방향으로 이동하는 기법.

순전파(forward propagation)?

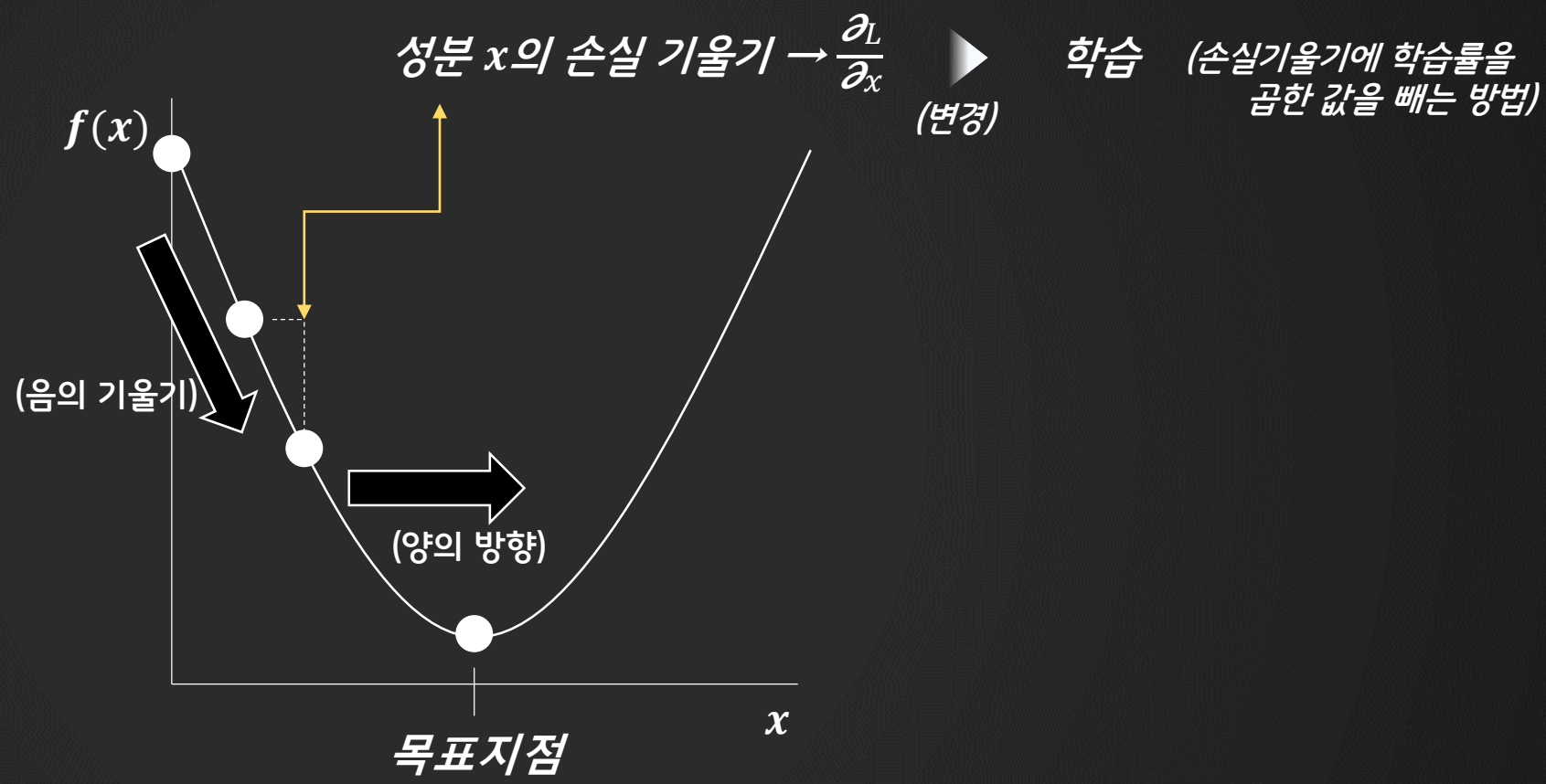
입력 데이터에 대해 신경망 구조를 따라가면서 현재의 파라미터값들을 이용해 손실함숫값을 계산하는 과정

역전파(backward propagation)?

순전파의 계산과정을 역순으로 거슬러 가면서 손실 함숫값에 직간접적으로 영향을 미친 모든 성분에 대해 손실 기울기를 계산하는 과정

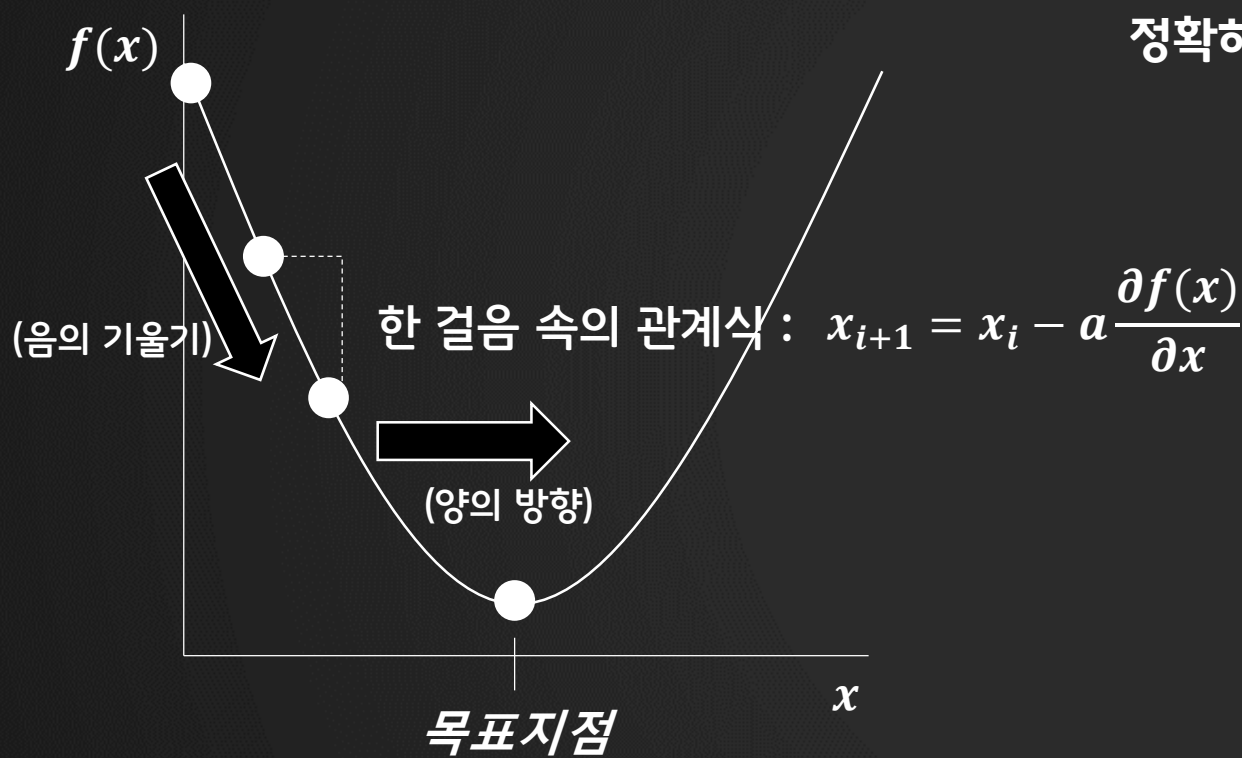
$$\text{성분 } x \text{의 손실 기울기} \rightarrow \frac{\partial L}{\partial x}$$

1.6 경사하강법과 역전파



2차원 그래프로 표현한 경사하강법의 원리

1.6 경사하강법과 역전파



2차원 그래프로 표현한 경사하강법의 원리

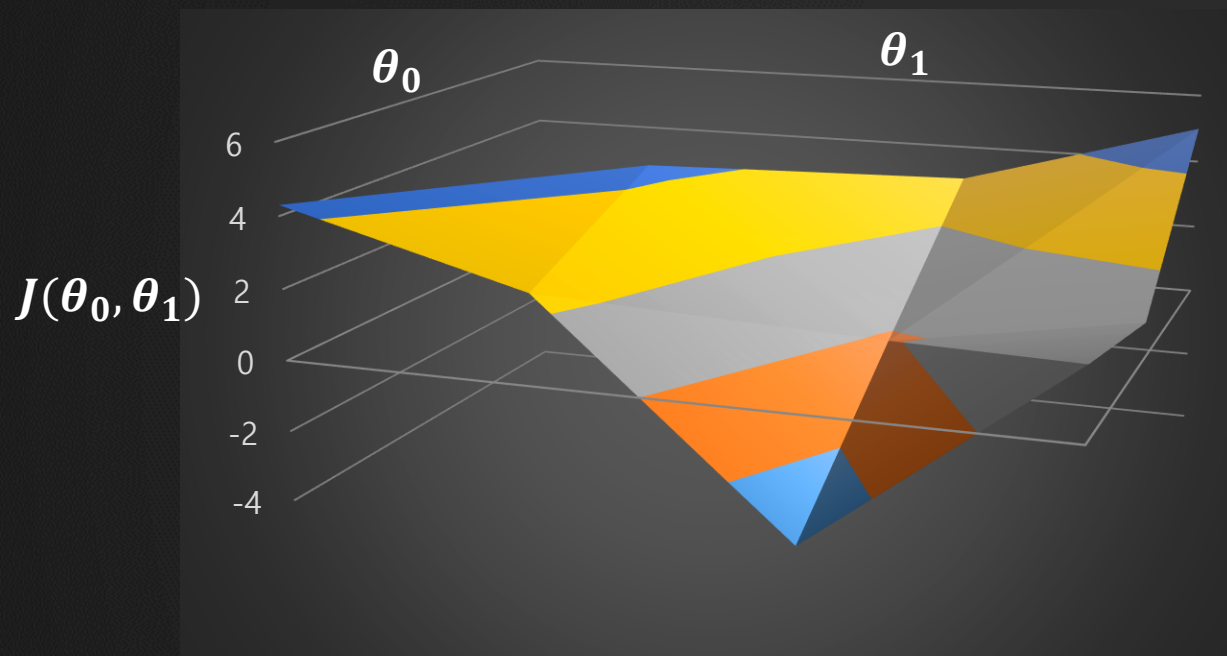
학습률(learning rate)?

학습률은 임의의 양수값을 사용할 수 있지만 값이 클 수록 목표 근처에서 정확하게 바닥을 찾는 능력이 무너지고, 값이 작을 수록 바닥 지점에 접근하는 시간이 더 오래걸림.

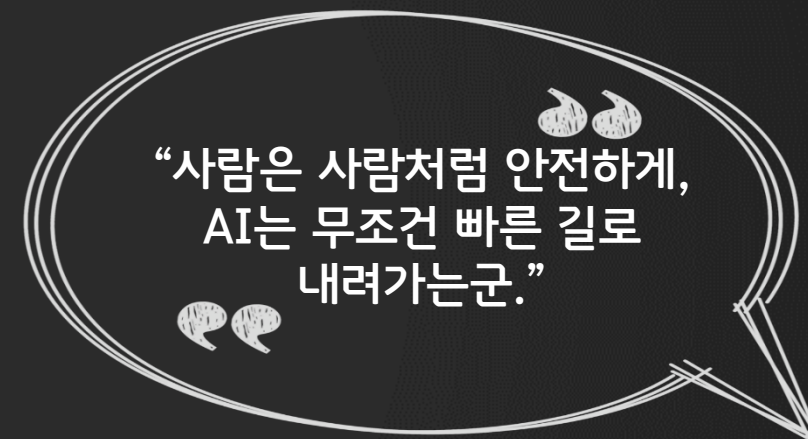
편미분(partial derivative)?

x 를 제외한 다른 변수 모두를 상수로 간주하고 미분하는
간소화된 미분법

1.6 경사하강법과 역전파

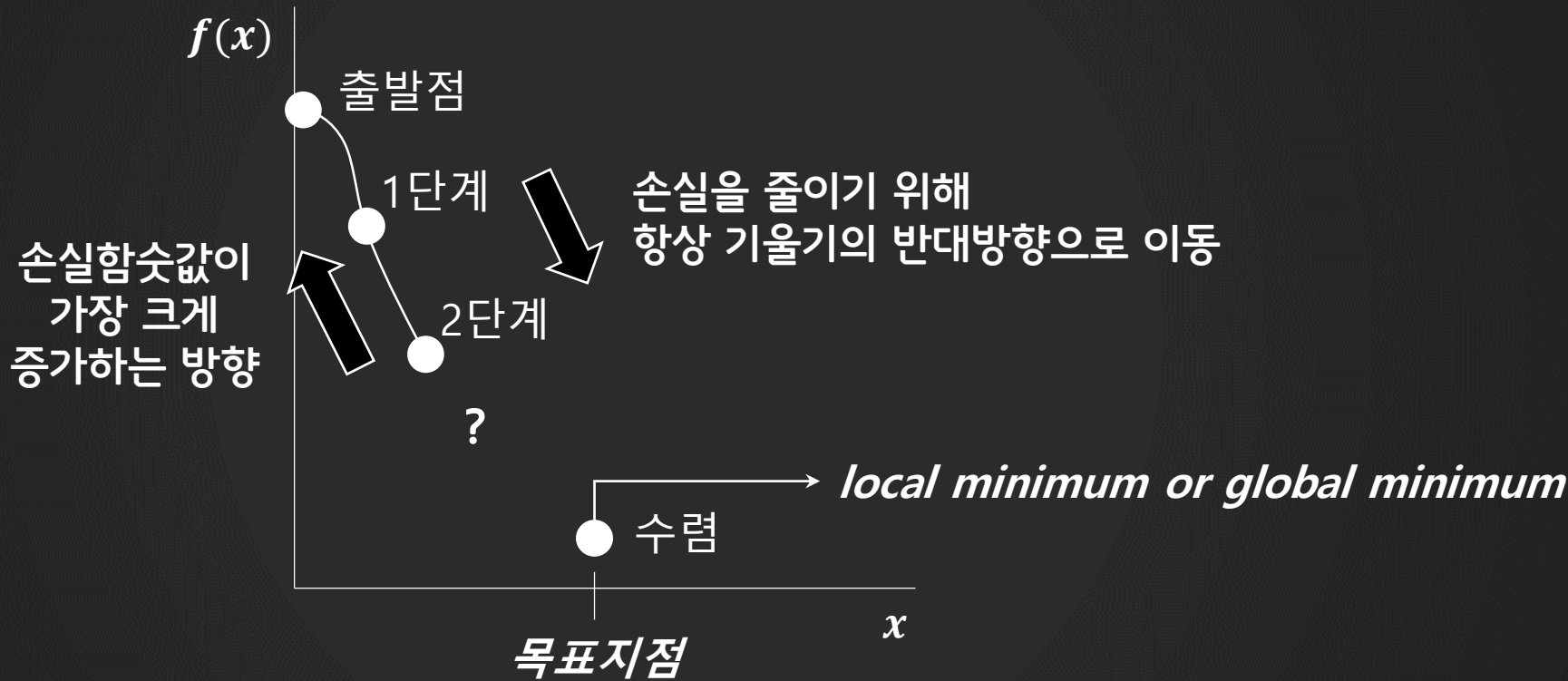


3차원 지형도에 비유해 살펴보는 경사하강법



1.6 경사하강법과 역전파

한 걸음 속의 관계식 : $x_{i+1} = x_i - a \frac{\partial f(x)}{\partial x}$

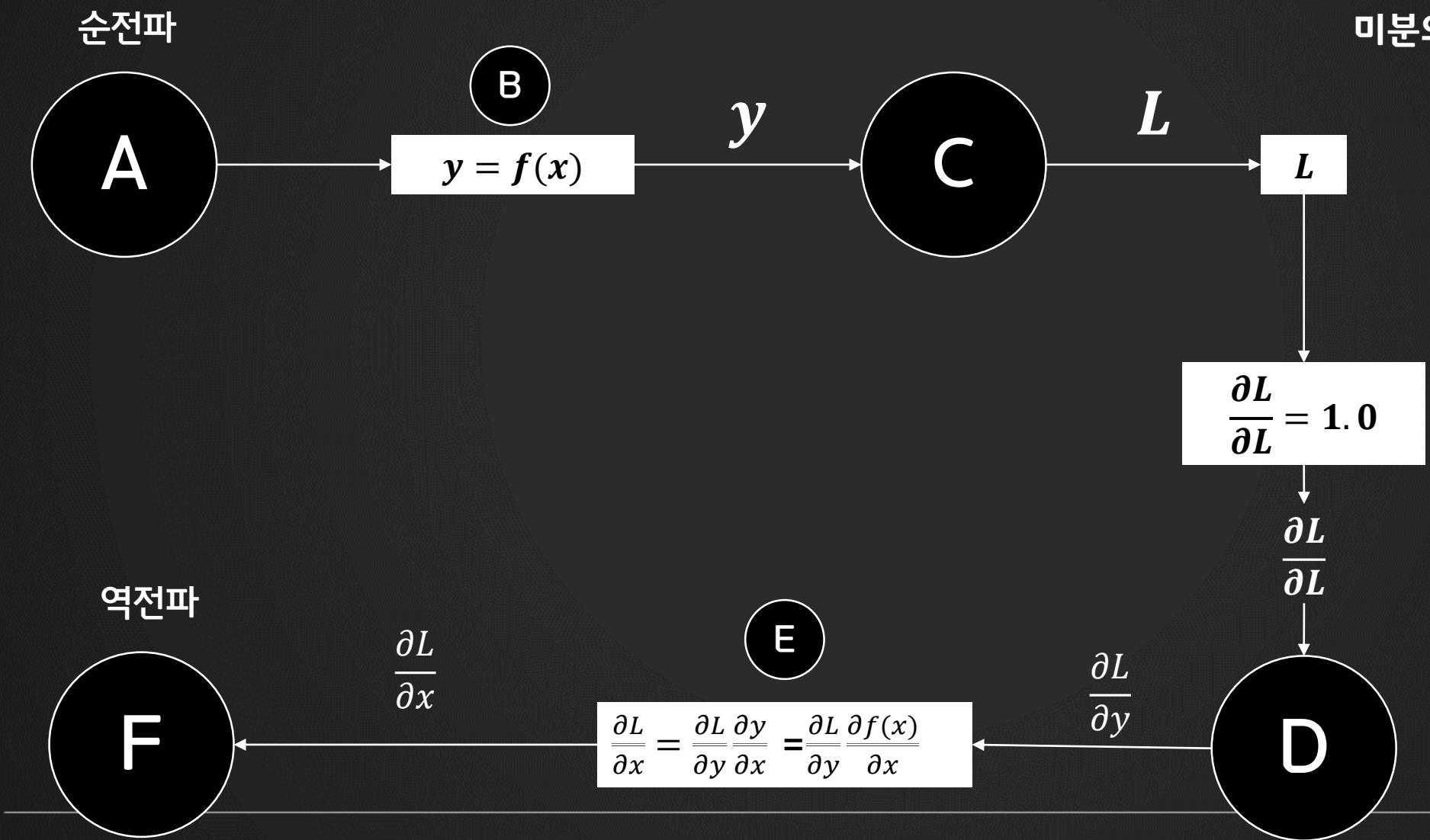


2차원 그래프로 표현한 경사하강법의 원리

1.7 편미분과 손실 기울기의 계산

편미분의 연쇄적 관계(chain rule)?

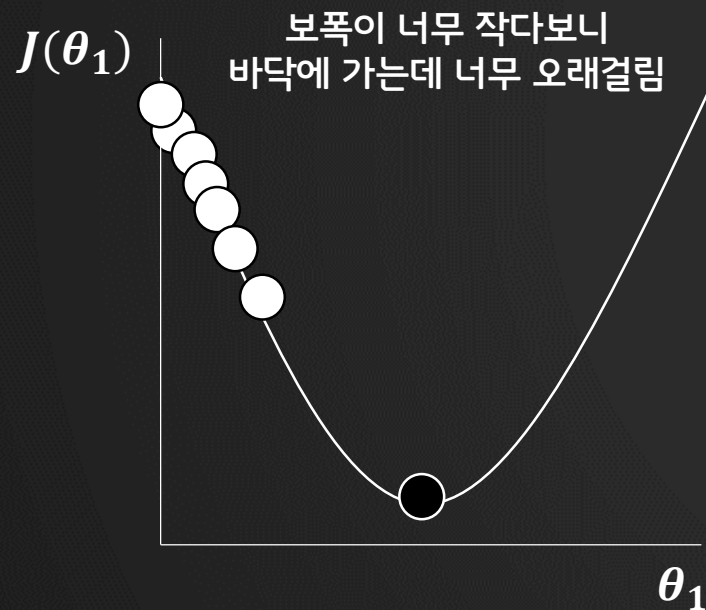
미분의 기본성질인 $\frac{\partial L}{\partial x} = \frac{\partial L}{\partial y} \frac{\partial y}{\partial x}$



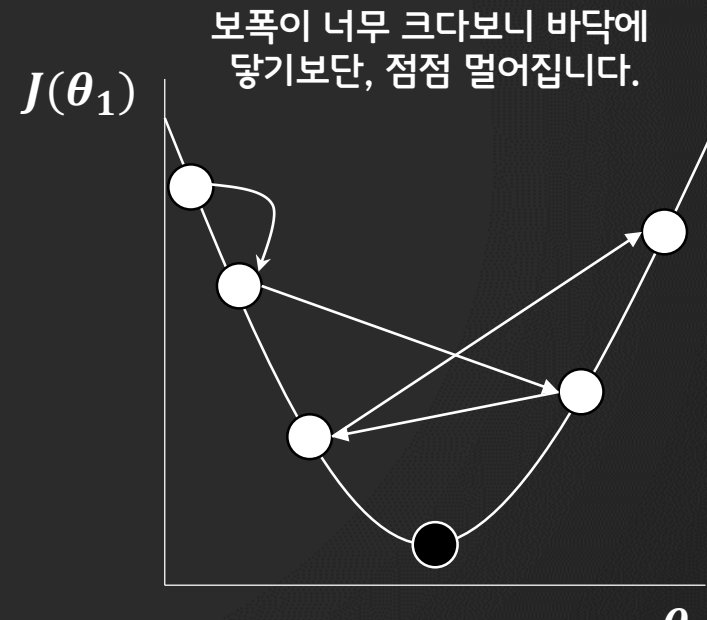
1.7 편미분과 손실 기울기의 계산

신경망의 목적은

“손실함수가 최솟값(오차가 최소)일 때의 파라미터를 찾아 올바른 학습결과를 내는 것”



㉑ 지나치게 작은 학습률



㉒ 지나치게 큰 학습률 θ_1

※ Adam : 고정된 학습률의 단점을 보완하기 위하여 학습 초반에는 큰 학습률을 사용하고, 바닥점에 가까워 질 수 록 학습률을 줄이는 기법.

1.8 하이퍼파라미터



㉠ 외부에서 주어지는 값
(재료 투입구)



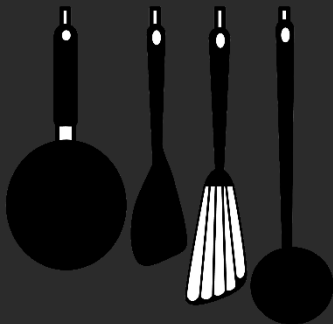
㉢ 각종 중간 계산 결과
(생산 결과물)

이전 맛에서는 너무
싱거웠고, 그에 따라
짭맛과 단맛이 필요해.



㉡ 파라미터
(가변 제어 장치)

소금 0.15 스푼 추가
설탕 1.15 스푼 추가



㉣ 하이퍼파라미터
(고정 장치)

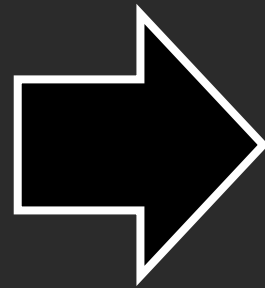
※ 오늘의 요리 기구들 ※
0.005 프라이팬
100 뒤집개
(요리 중간에 변경 불가
단, 다음 요리 시작때는
변경 가능)

딥러닝 모델에 등장하는 네 가지 값들

1.9 비선형 정보와 원-핫 벡터 표현



SEX = I SEX = M SEX = F



| 유충 | 수컷 | 암컷 |
|----|----|----|
| 0 | 1 | 2 |

[전북 성별 정보의 비선형 정보 표현]

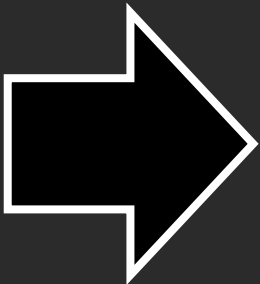
“성별정보를 0,1,2 라고 표현하는 것은
과연 타당한 선택일까?”



1.9 비선형 정보와 원-핫 벡터 표현



SEX = I SEX = M SEX = F



[전복 성별 정보의 원-핫 벡터 표현]

| 유충 | 수컷 | 암컷 |
|--------------------------------------|--------------------------------------|--------------------------------------|
| infant = 1 male = 0 female = 0 | infant = 0 male = 1 female = 0 | infant = 0 male = 0 female = 1 |

기존 입력크기 8 (‘성별’(1) + 선형적 특징값 7가지)

| <i>sex</i> | <i>Length</i> | <i>Diameter</i> | <i>Height</i> | ... | <i>Shell weight</i> |
|------------|---------------|-----------------|---------------|-----|---------------------|
|------------|---------------|-----------------|---------------|-----|---------------------|

“이렇게 표현하면 신경망도 이 특징값이
뭘 의미하는지 파악할 수 있겠다!”



개선된 입력크기 10 (원 - 핫 벡터 (3) + 선형적 특징값 7가지)

| <i>infant</i> | <i>male</i> | <i>female</i> | <i>Length</i> | <i>Diameter</i> | <i>Height</i> | ... | <i>Shell weight</i> |
|---------------|-------------|---------------|---------------|-----------------|---------------|-----|---------------------|
|---------------|-------------|---------------|---------------|-----------------|---------------|-----|---------------------|

※ 지금까지의 설명 요약

전복 고리 수 추정 문제

입력 벡터 크기 : 10 (3 + 7)

출력 벡터 크기 : 1

미니배치 크기 : mb_size

퍼셉트론의 수 : 1

단층 퍼셉트론

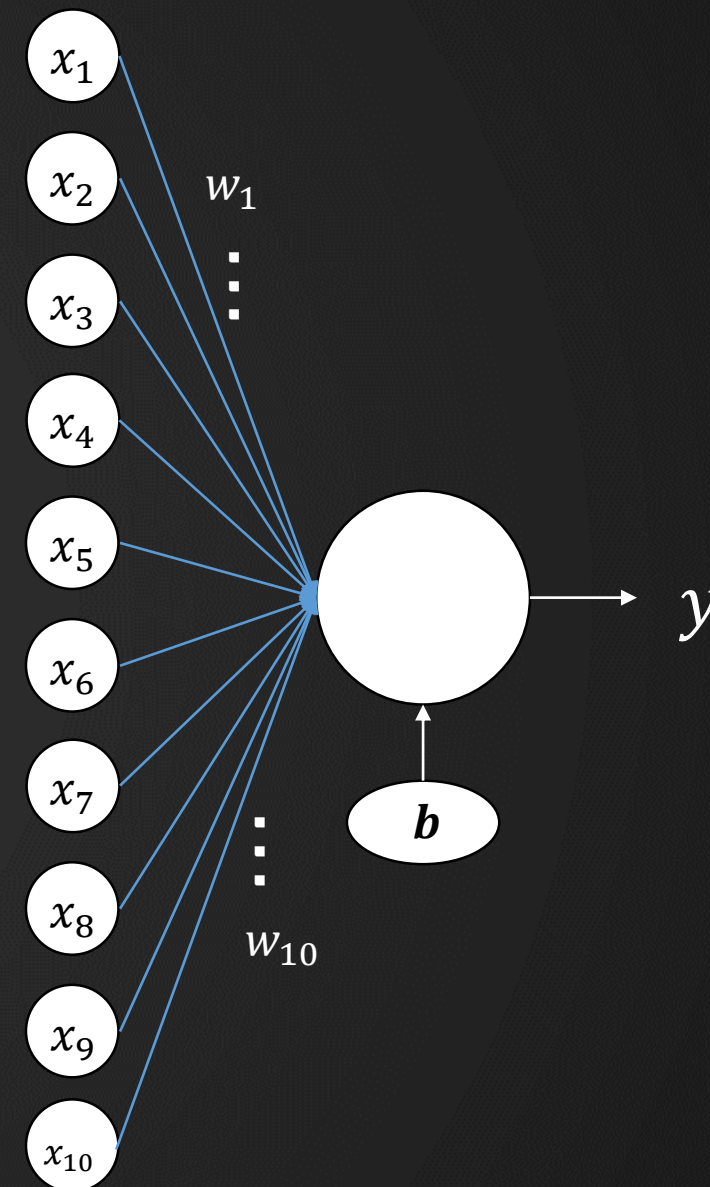
입력 : [mb_size, 10]

출력 벡터 크기 : [mb_size, 1]

미니배치 크기 : mb_size

가중치 정보 : [10, 1]

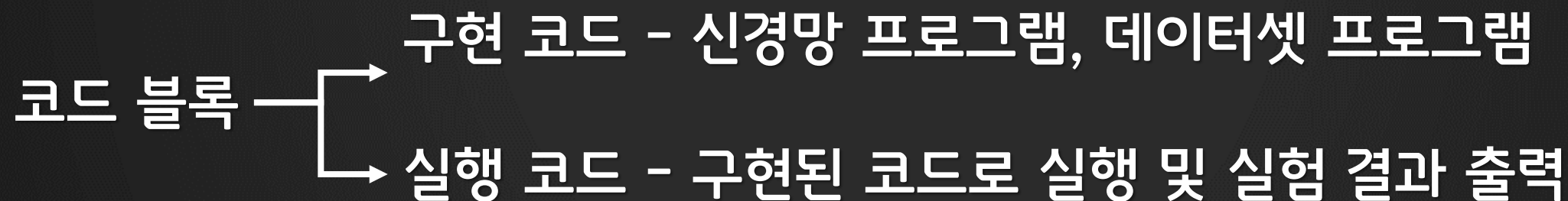
편향 정보 : [1]



1.10 구현하기 : 전복 고리 수 추정 신경망

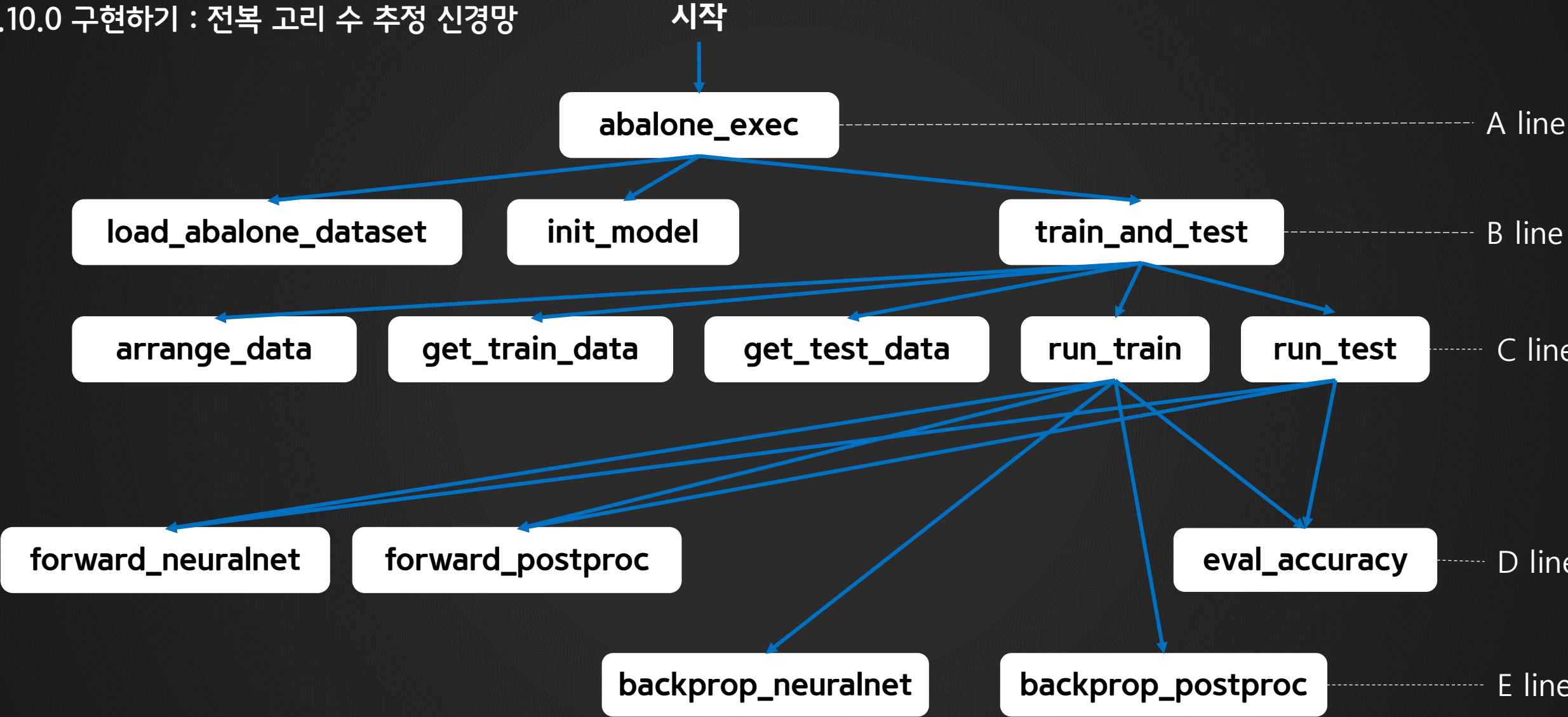
전복의 고리 수를 추정하는 프로그램을 파이썬으로 구축해보기
(코드 블록 단위로 진행)

※ 코드 블록이란?



★ 코드 블록이 지나치게 늘어나지 않도록 코드를 최대한 재활용

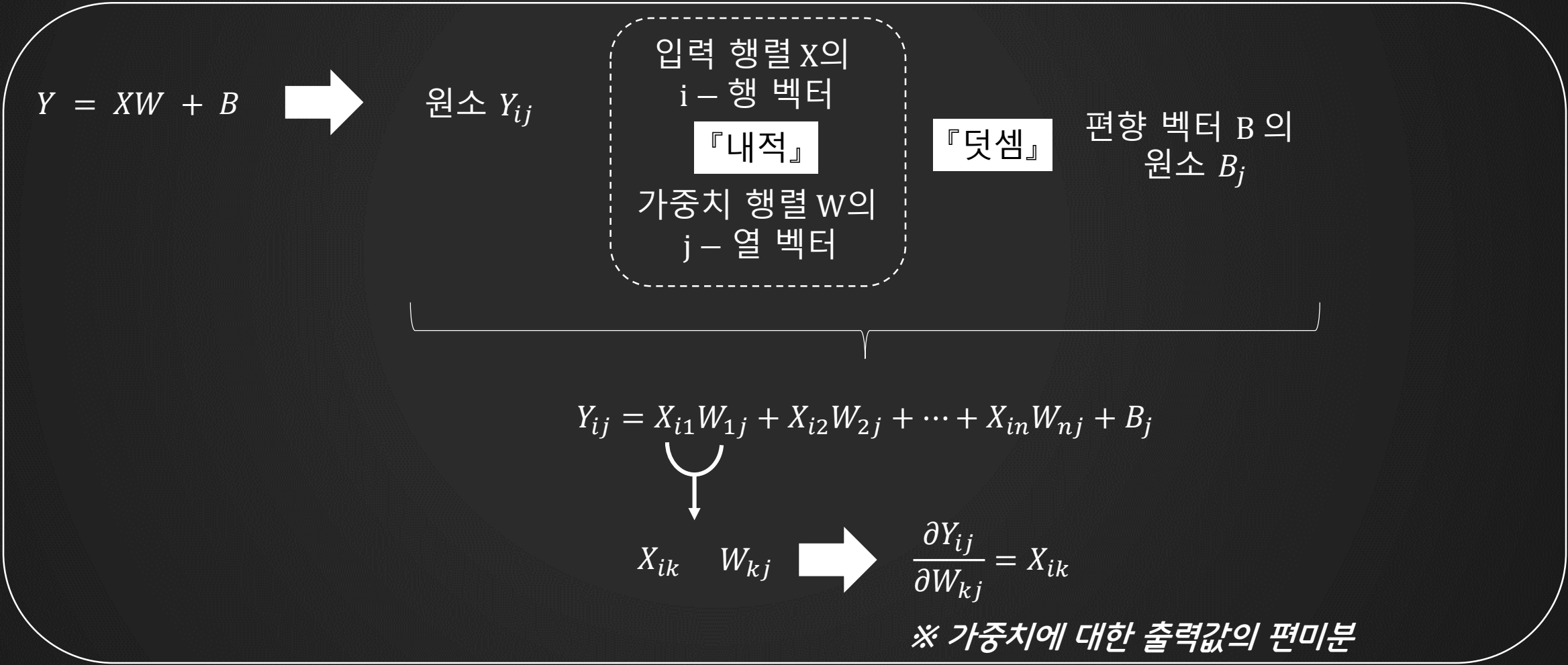
1.10.0 구현하기 : 전복 고리 수 추정 신경망



※ 가중치와 편향의 손실 기울기 (1)

$$\frac{\partial L}{\partial W} \rightarrow \frac{\partial L}{\partial W} = \frac{\partial L}{\partial Y} \frac{\partial Y}{\partial W}$$

$$\frac{\partial Y}{\partial W}$$



※ 가중치와 편향의 손실 기울기 (2)

$$\frac{\partial L}{\partial W} \rightarrow \frac{\partial L}{\partial W} = \frac{\partial L}{\partial Y} \frac{\partial Y}{\partial W}$$

$$\frac{\partial L}{\partial Y}$$

Y_{ij} 성분의 손실 기울기

$$\frac{\partial L}{\partial Y_{ij}} = G_{ij}$$

Part 1. 단층 퍼셉트론

※ 가중치와 편향의 손실 기울기 (3)

$\frac{\partial L}{\partial W} \rightarrow \frac{\partial L}{\partial W} = \frac{\partial L}{\partial Y} \frac{\partial Y}{\partial W}$

W_{Kj} 성분이 L 에 미치는 영향
(Y_{ij} 를 통해)

$\frac{\partial L}{\partial Y_{ij}} \frac{\partial Y_{ij}}{\partial W_{kj}} = G_{ij} X_{ik}$

Y 의 j -열 벡터를 이루는
다수의 원소들
($Y_{1j}, Y_{2j}, \dots, Y_{mj}$)

W_{Kj} 의 손실 기울기

(이들 경로로 각각을 통해 얻어지는
기울기 성분을 모두 합)

$(G_{1j} X_{1k} + G_{2j} X_{2k} + \dots + G_{mj} X_{mk})$

(행렬곱셈에
알맞게 재구성)

X 대신 행과 열을 뒤바꾼 전치행렬
 $T = X^T$

최종적인
가중치에 대한
손실함수 수식

$\frac{\partial L}{\partial W_{kj}} = T_{k1} G_{1j} + T_{k2} G_{2j} + \dots + T_{km} G_{mj}$
 $\frac{\partial L}{\partial W} = T G = X^T G$

Part 1. 단층 퍼셉트론

※ 가중치와 편향의 손실 기울기 (1)

$\frac{\partial L}{\partial B} \rightarrow \frac{\partial L}{\partial B} = \frac{\partial L}{\partial Y} \frac{\partial Y}{\partial B}$

Y_{ij} 성분의 손실 기울기

$$\frac{\partial L}{\partial Y_{ij}} = G_{ij}$$

$$Y = XW + B$$
$$Y_{ij} = X_{i1}W_{1j} + X_{i2}W_{2j} + \cdots + X_{in}W_{nj} + B_j$$

편미분의 성질로 인한
연산결과 1 출력

$$\frac{\partial Y_{ij}}{\partial B_j} = 1$$

$$\frac{\partial L}{\partial Y_{ij}} \frac{\partial Y_{ij}}{\partial B_j} = G_{ij}$$

다수의 원소들
($Y_{1j}, Y_{2j}, \dots, Y_{mj}$)

$$\frac{\partial L}{\partial Y_{ij}} \quad \frac{\partial Y_{ij}}{\partial B_j} = G_{ij}$$

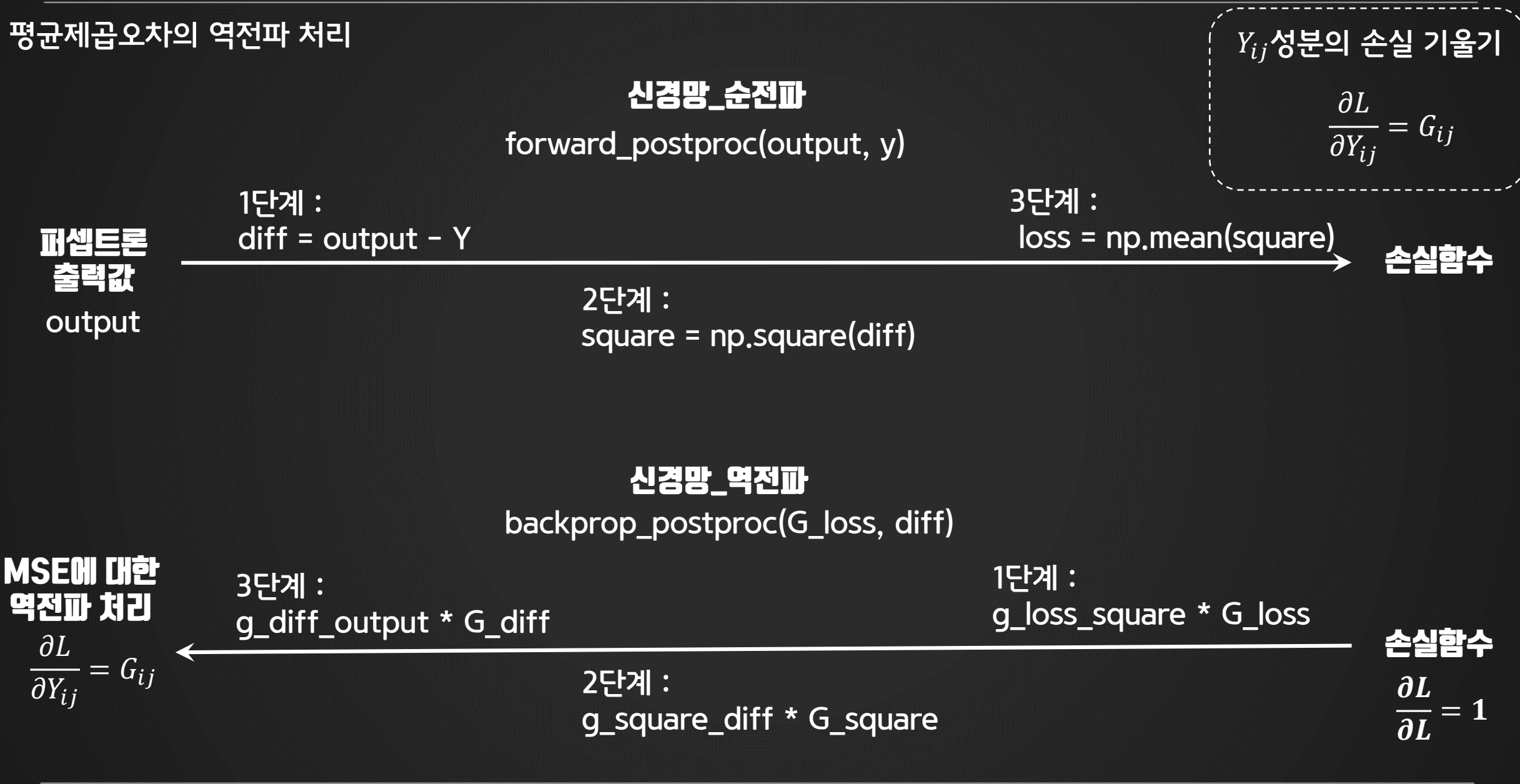
손실함숫값에 영향

$$\frac{\partial L}{\partial B_j} = G_{1j} + G_{2j} + \cdots + G_{mj}$$

이 계산은 1로 채워진 행렬과 G의 곱으로 구할 수도 있겠지만 구현 코드에서와 같이 단순히 G의 각 행의 합을 구하는 간단한 방법으로도 구할 수 있습니다.

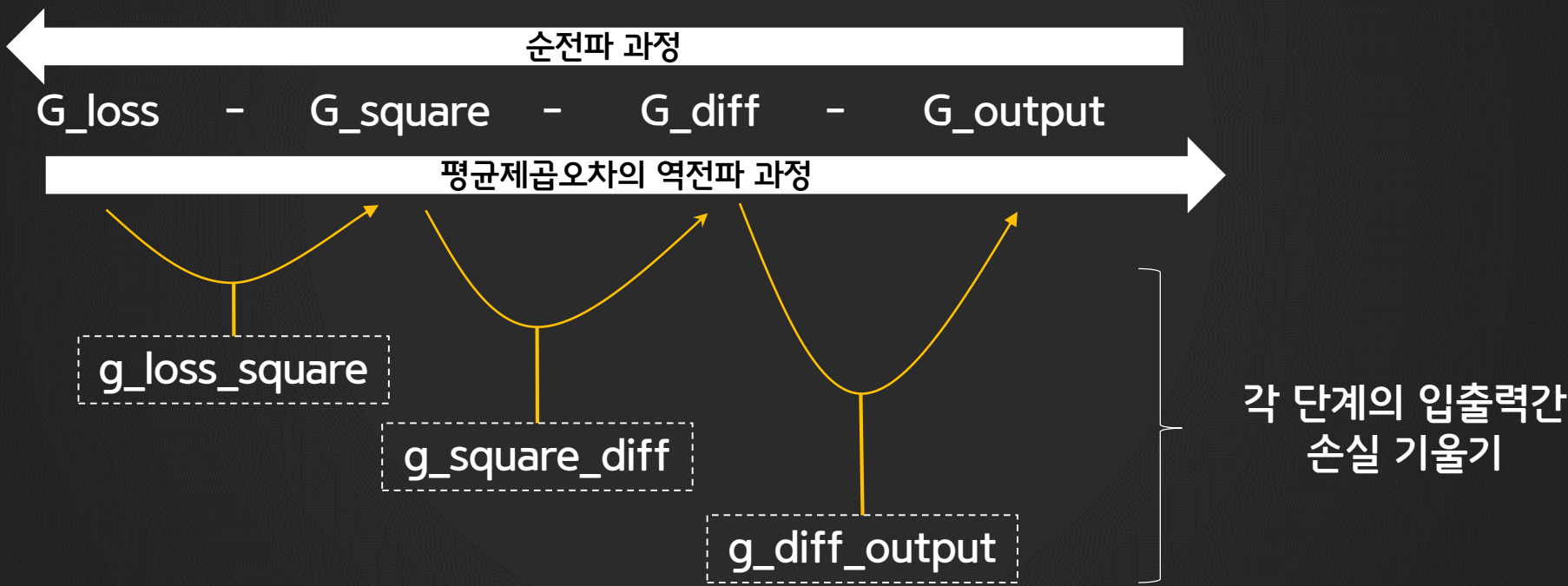
Part 1. 단층 퍼셉트론

※ 평균제곱오차의 역전파 처리



※ 평균제곱오차의 역전파 처리

순전파의 역순으로 평균 / 제곱 / 오차에 대한 역전파 처리를 차례로 수행
backprop_postproc(G_loss, diff)



※ 손실 기울기의 연쇄적 계산

Part 1. 단층 퍼셉트론

※ 평균제곱오차의 역전파 처리

※ loss만 스칼라 값일 뿐 나머지는 모두 [미니배치 크기, 출력벡터 크기], 즉 [N,M]의 크기를 갖는다.

g_loss_square 평균연산 $L = \frac{\sum_{i=1}^M \sum_{j=1}^N square_{ij}}{MN} \quad \blacktriangleright \quad \frac{\partial L}{\partial square_{ij}} = \frac{1}{MN}$

python code : `g_loss_square = np.ones(shape) / np.prod(shape)`

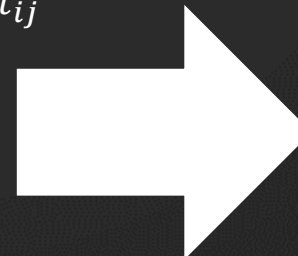
g_square_diff 제곱 연산 $square_{ij} = diff_{ij}^2 \quad \blacktriangleright \quad \frac{\partial square_{ij}}{\partial diff_{ij}} = 2diff_{ij}$

python code : `g_square_diff = 2 * diff`

g_diff_output 편차 연산 $diff_{ij} = output_{ij} - y_{ij} \quad \blacktriangleright \quad \frac{\partial diff_{ij}}{\partial output_{ij}} = 1$

python code : `g_diff_output = 1`

연쇄적 계산을 통한
G_output 도출



`G_square = g_loss_square * G_loss`
`G_diff = g_square_diff * G_square`
`G_output = g_diff_output * G_diff`