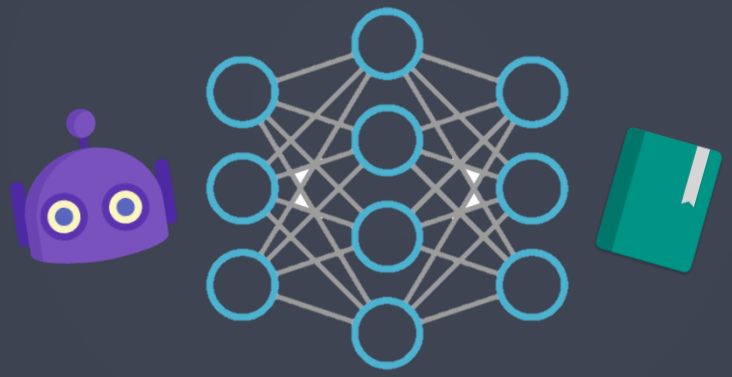
Deep Learning

Chapter 3 활성화 함수,오차 역전파, 경사하강법 (Activation Function, Back Propagation, Gradient Descent Algorithm)



START

Smart Media
스마트미디어인재개발원

학습목표

- 활성화 함수를 이해 할 수 있다.
- 오차역전파의 개념을 이해 할 수 있다.
- 다양한 경사하강법 종류를 알 수 있다.
- Keras를 활용해 다양한 경사하강법을 적용 할 수 있다.

활성화 함수(Activation Function)

활성화 함수(Activation Function)

- 신경망은 (선형회귀와 달리) 한 계층의 신호를 다음 계층으로 그대로 전달하지 않고 비선형적인 활성화 함수를 거친 후에 전달한다.

- 이렇게 하는 이유는 생물학적인 신경망을 모방한 것
  - 약한 신호는 전달하지 않고 어느 이상의 신호도 전달하지 않는 "S"자 형 곡선과 같이 "비선형적"인 반응을 한다고 생각했다.

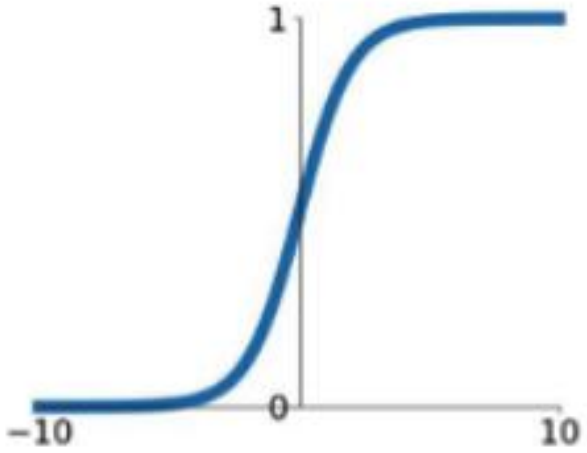
- 실제로 비선형의 활성화 함수를 도입한 신경망이 잘 동작하고 있다.

활성화 함수(Activation Function)

Sigmoid 함수

Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



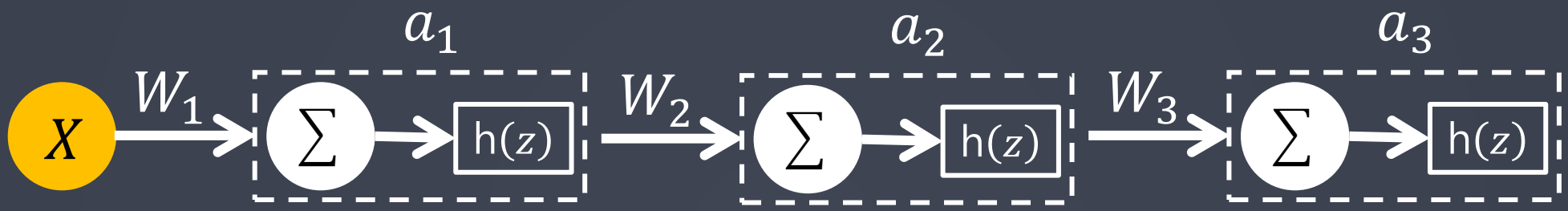
활성화 함수(Activation Function) 정리

1. 활성화 함수로 비선형 함수를 사용하는 이유

- 계단 함수(step)와 시그모이드 함수(sigmoid)는 비선형 함수이다.
- 활성화 함수로 선형함수 ex) $h(z) = cz$ 를 사용하면 중간층(은닉층)을 여러 개 구성한 효과를 살릴 수 없다.
- $y(z) = h(h(h(z))) = c * c * c * z = az$

활성화 함수(Activation Function) 정리

1. 활성화 함수로 비선형 함수를 사용하는 이유



✓ a1 = W1 * X * c

✓ a2 = W2 * a1 * c

✓ a3 = W3 * a2 * c

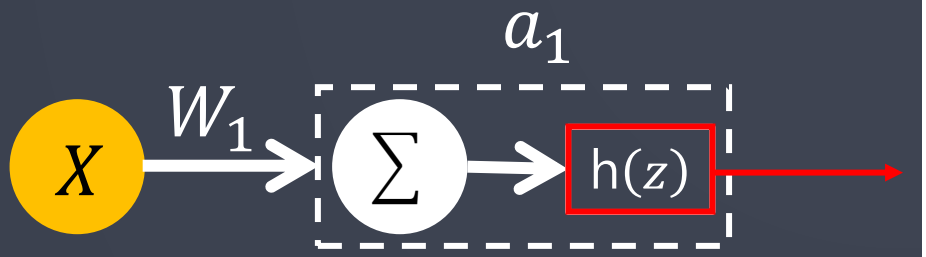
- y(z) = h(h(h(z))) = c * c * c * z = az
- a3 = W3 * W2 * W1 * X * c * c * c

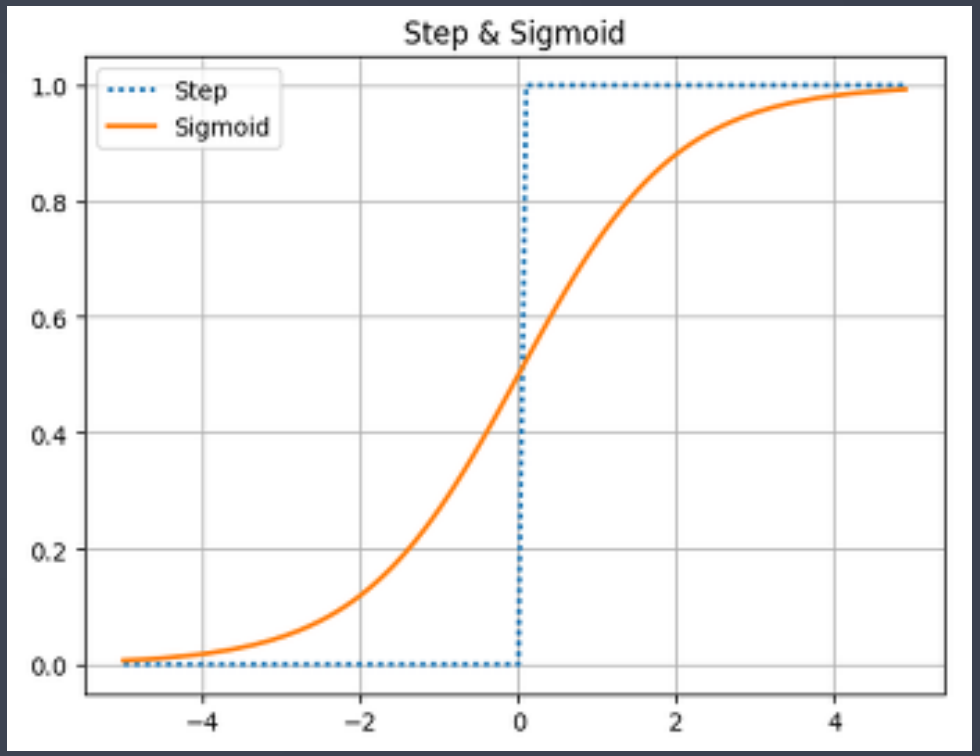
활성화 함수(Activation Function) 정리

2. Step와 Sigmoid의 차이

- 선형 모델이 학습하기위해서 경사하강법(cost 함수를 미분)을 사용.

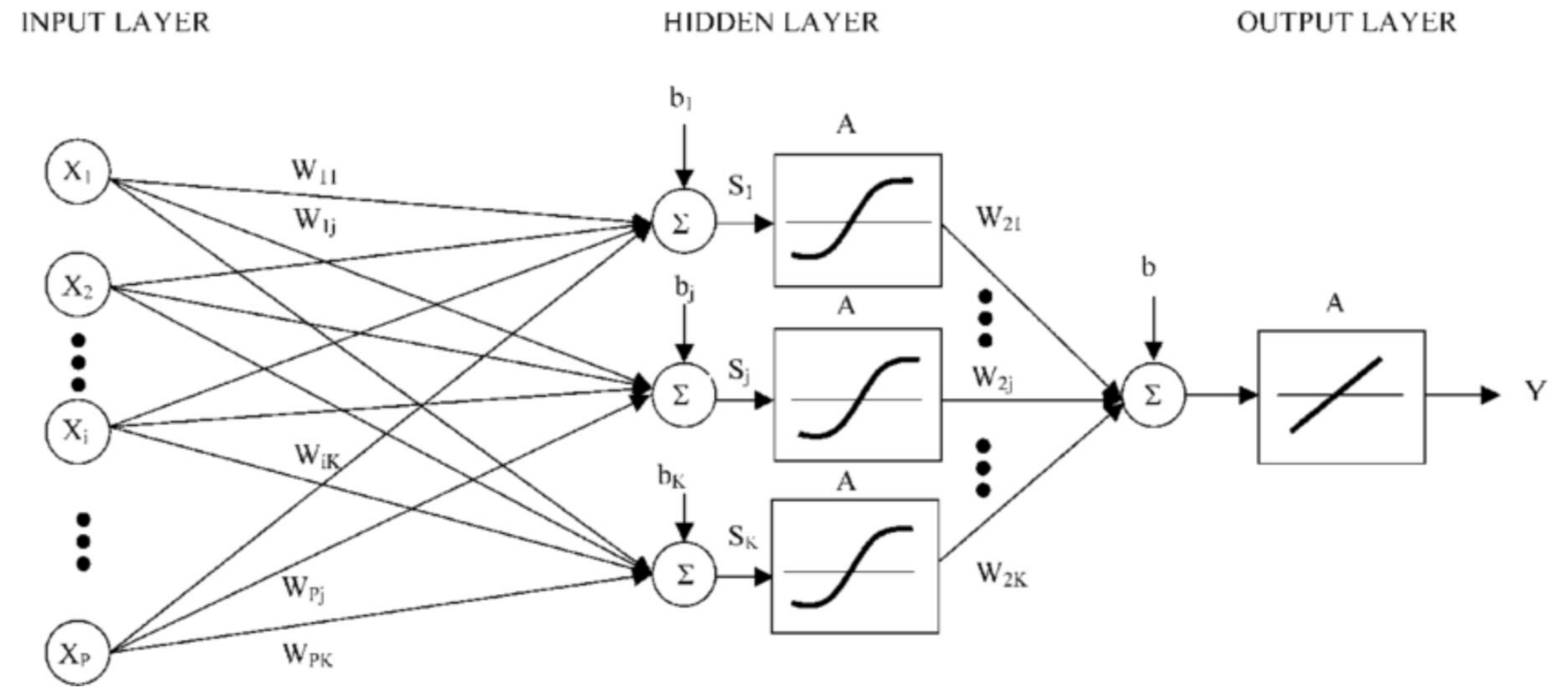
$$cost = \frac{1}{m}\sum_{i=1}^{m}(H(x_i) - y_i)^2$$





활성화 함수(Activation Function)

층에 따라 다른 활성화 함수를 사용 할 수 있다.



문제 유형에 따라 사용되는 활성화 함수와 오차함수 종류

유형	출력층 활성화 함수	오차함수
회귀	항등 함수 **linear**	제곱오차 **mean_squared_error**
이진 분류	로지스틱 함수 **sigmoid**	교차 엔트로피 **binary_crossentropy**
다클래스 분류	스프트맥스 함수 **softmax**	교차 엔트로피 **categorical_crossentropy**

문제 유형에 따라 사용되는 활성화 함수와 오차함수 종류

소프트맥스(softmax) 함수

$$y_k = \frac{\exp(a_k)}{\sum_{i=1}^{n} \exp(a_i)}$$

문제 유형에 따라 사용되는 활성화 함수와 오차함수 종류

소프트맥스(softmax) 함수

```
1 import numpy as np
2
3 def softmax(x):
4     e_x = np.exp(x-x.max())
5     return e_x/e_x.sum()
```

```
1 x = np.array([1.0,1.0,2.0])
2 x
```

```
array([1., 1., 2.])
```

```
1 y = softmax(x)
2 y
```

```
array([0.21194156, 0.21194156, 0.57611688])
```

```
1 y.sum()
```

```
1.0
```

Keras 활용 XOR 문제풀기

딥러닝(Deep Learning) 실습

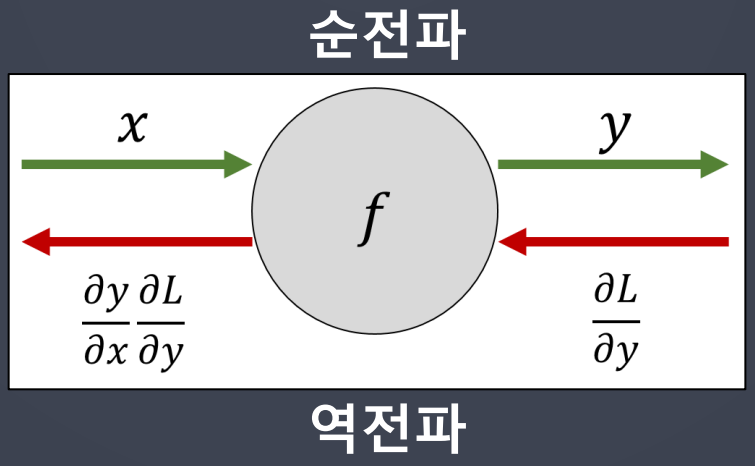
iris 데이터 신경망으로 풀기 (분류)

오차 역전파(Back Propagation)

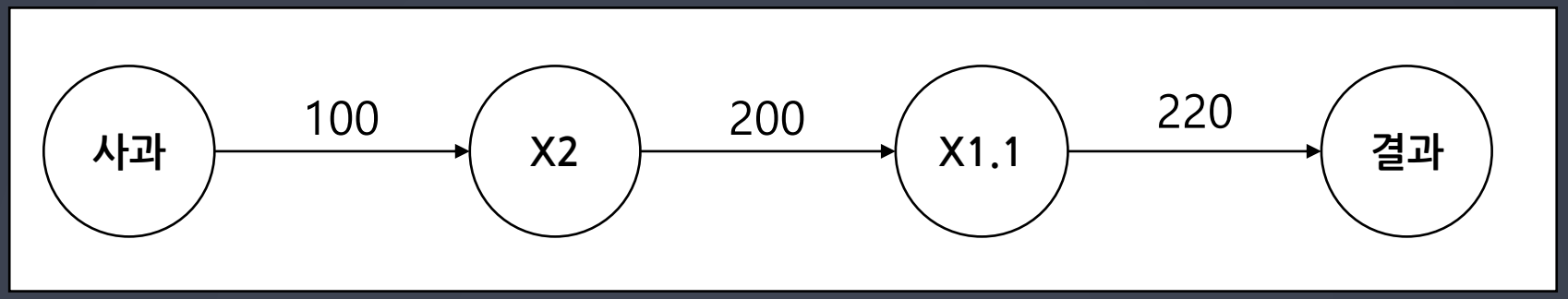
오차 역전파(Back Propagation)

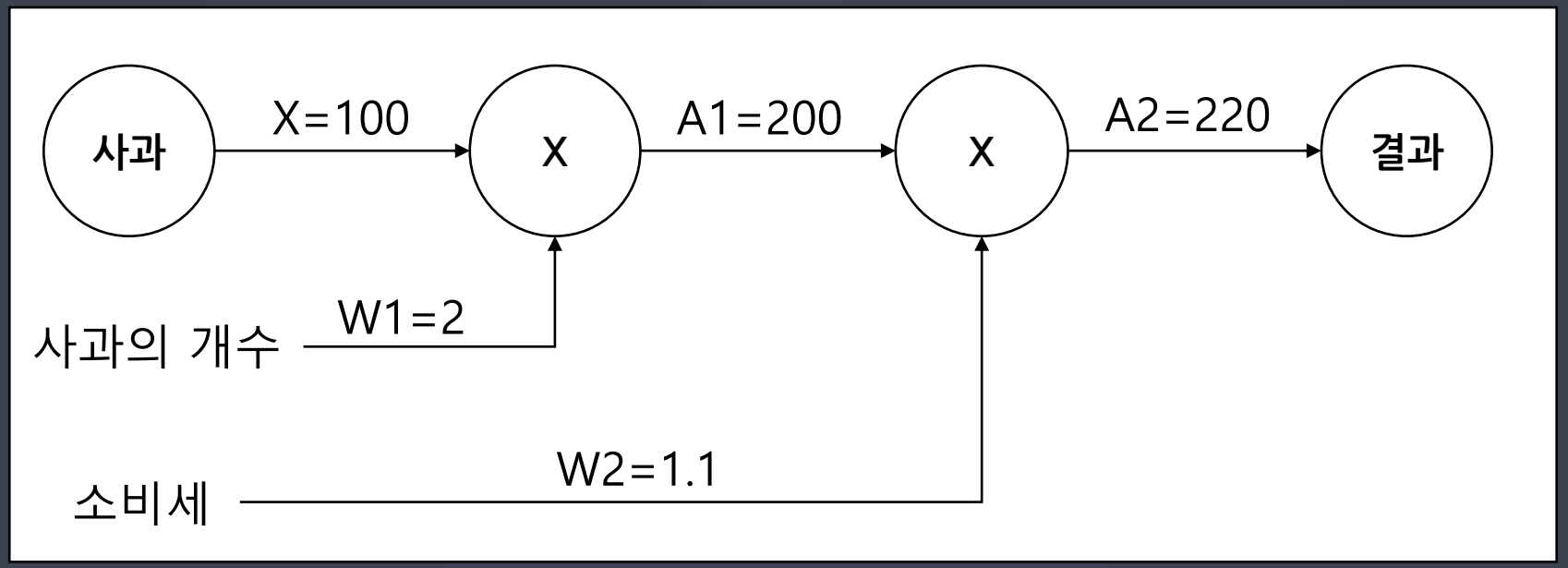
- 순전파 : 입력 데이터를 입력층에서부터 출력층까지 전파시키면서 출력값을 찾아가는 과정 -> 추론

- 역전파 : 에러를 출력층에서 입력층 쪽으로 전파시키면서 최적의 학습 결과를 찾아가는 것 -> 학습

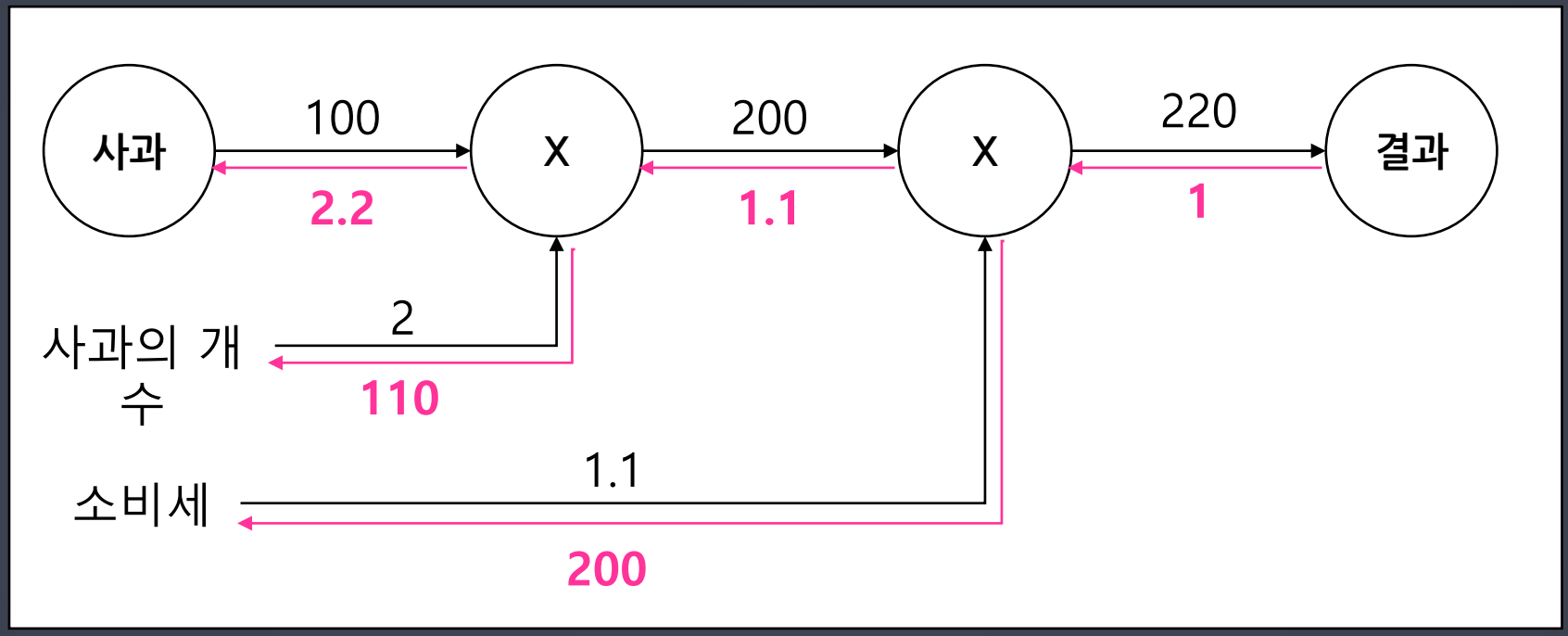


오차 역전파(Back Propagation)



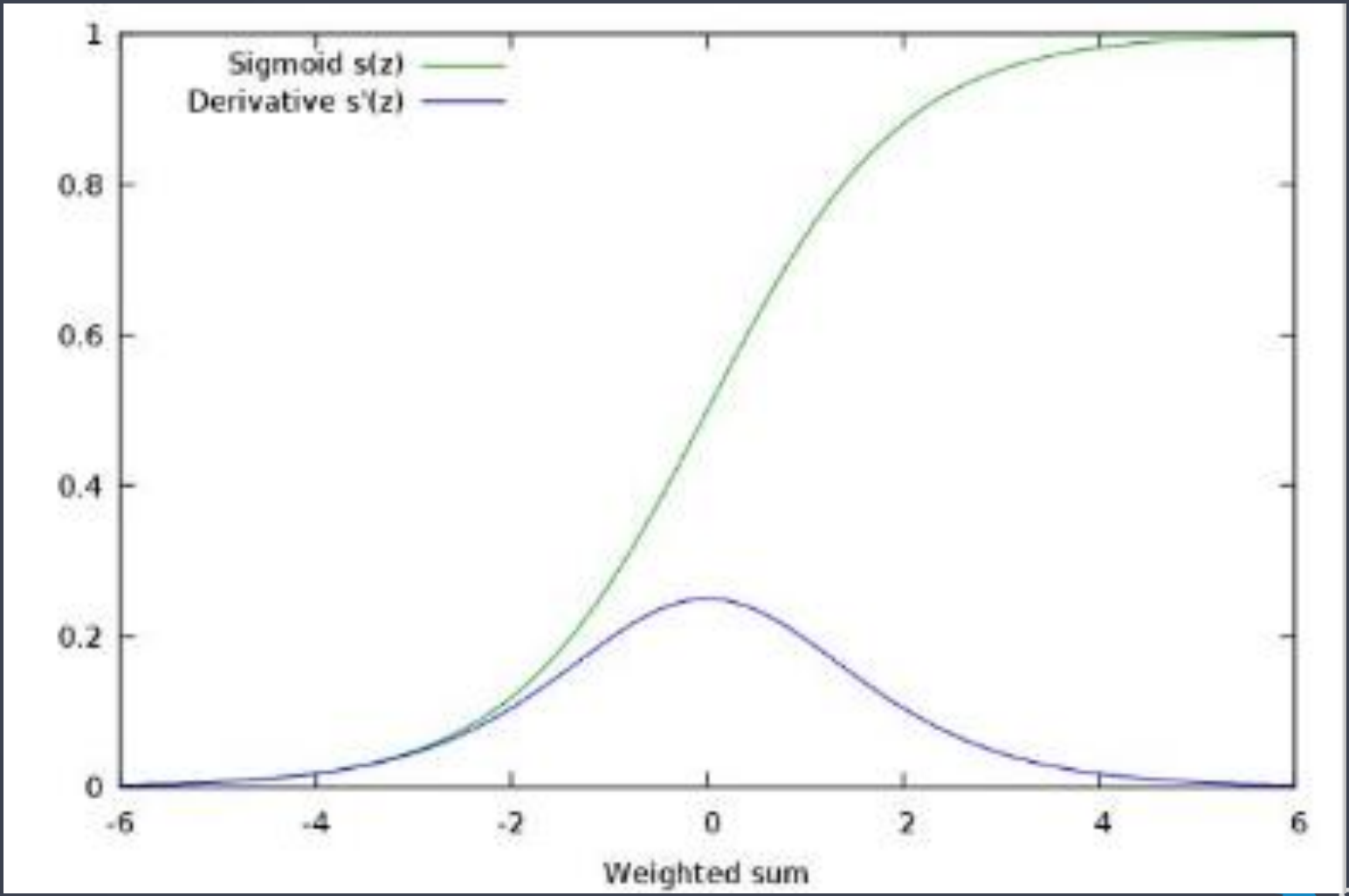


오차 역전파(Back Propagation)



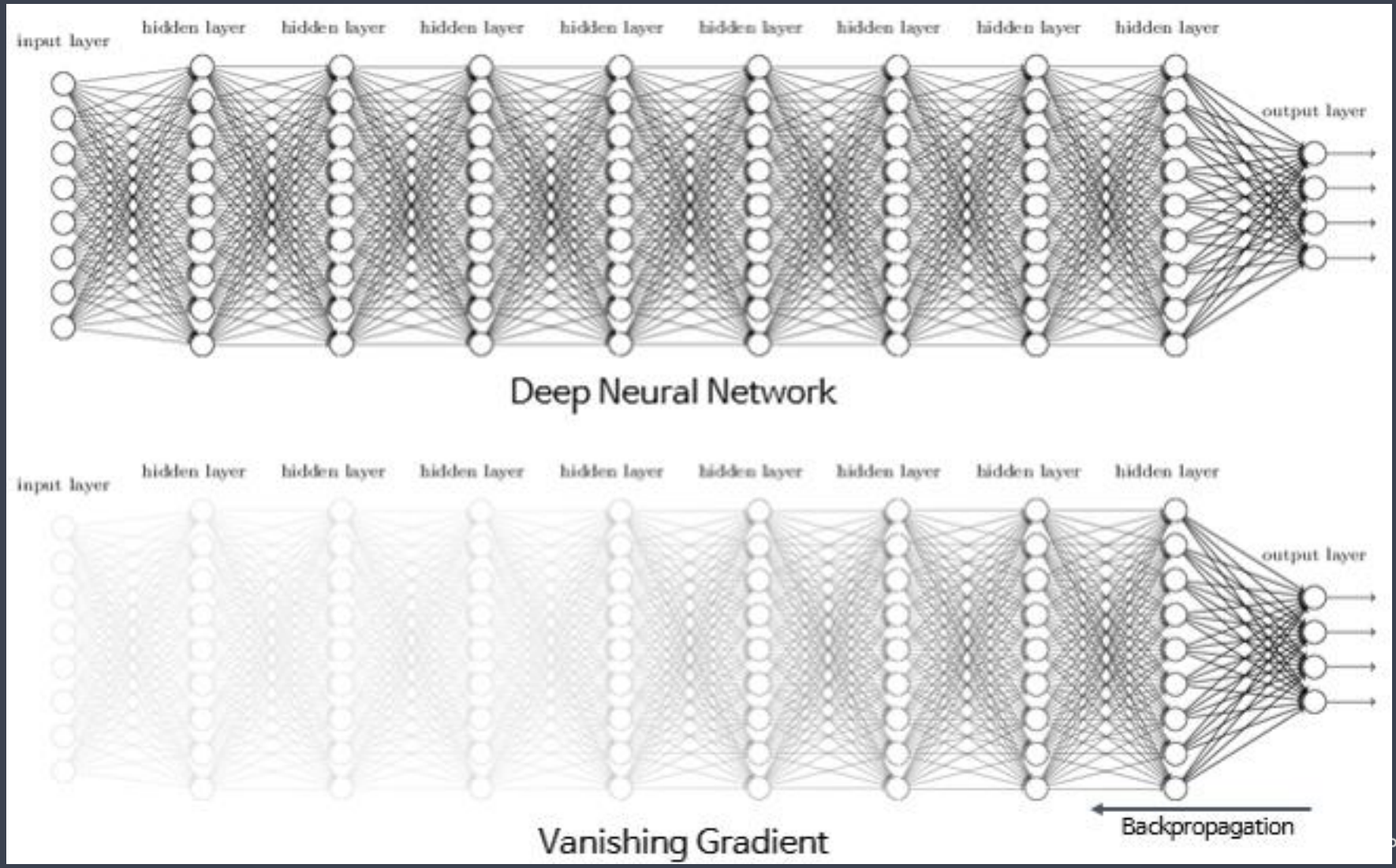
오차 역전파(Back Propagation)

Sigmoid 함수의 문제점



오차 역전파(Back Propagation)

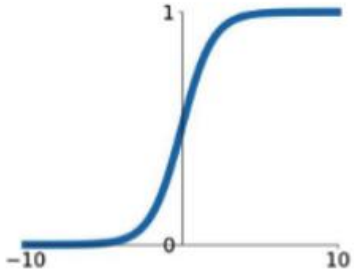
Sigmoid 함수의 문제점



오차 역전파(Back Propagation)

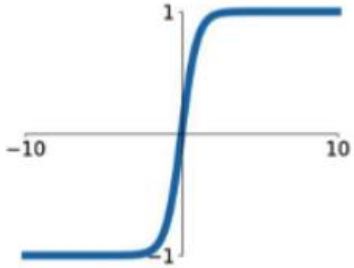
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



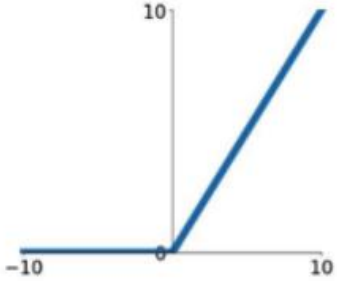
tanh

$$\tanh(x)$$



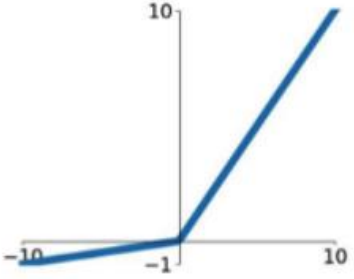
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$

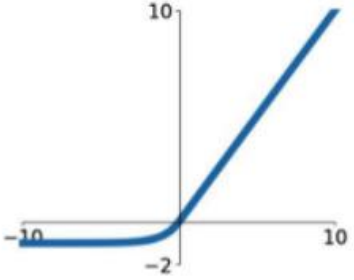


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

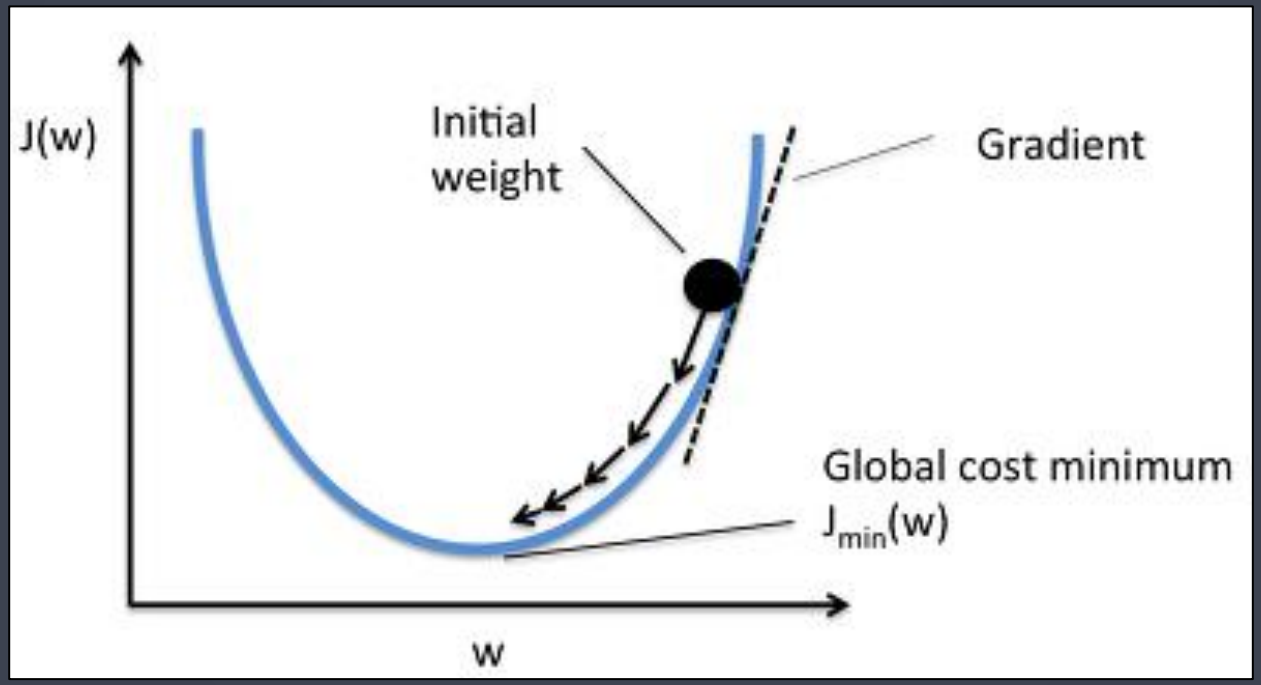
ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$

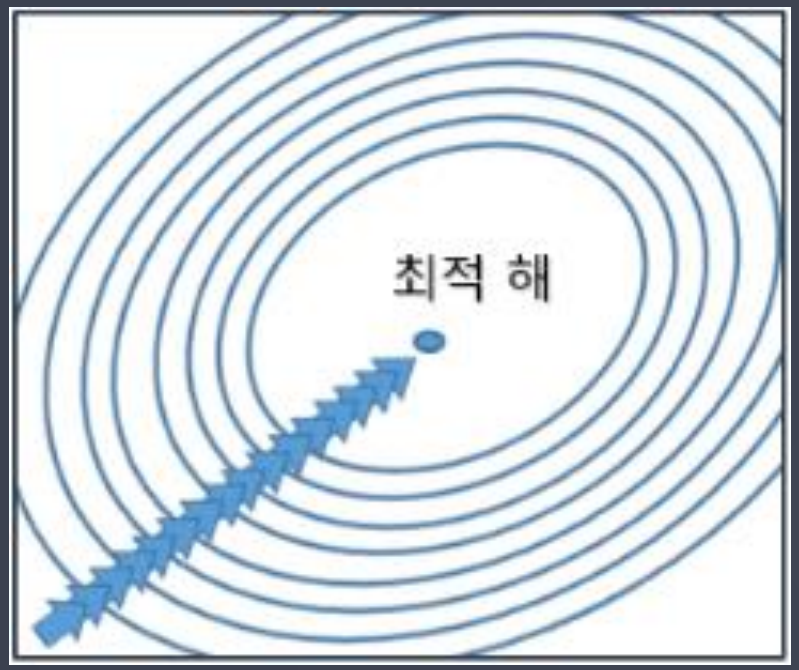


경사하강법(Gradient Descent Algorithm)

경사하강법(Gradient Descent Algorithm)

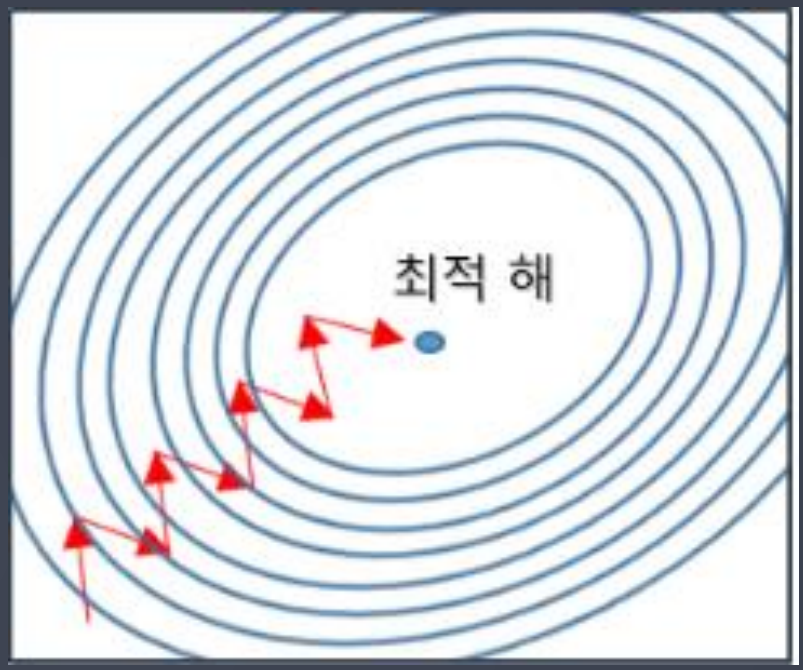


경사하강법(Gradient Descent Algorithm)



경사하강법
(Gradient Descent)

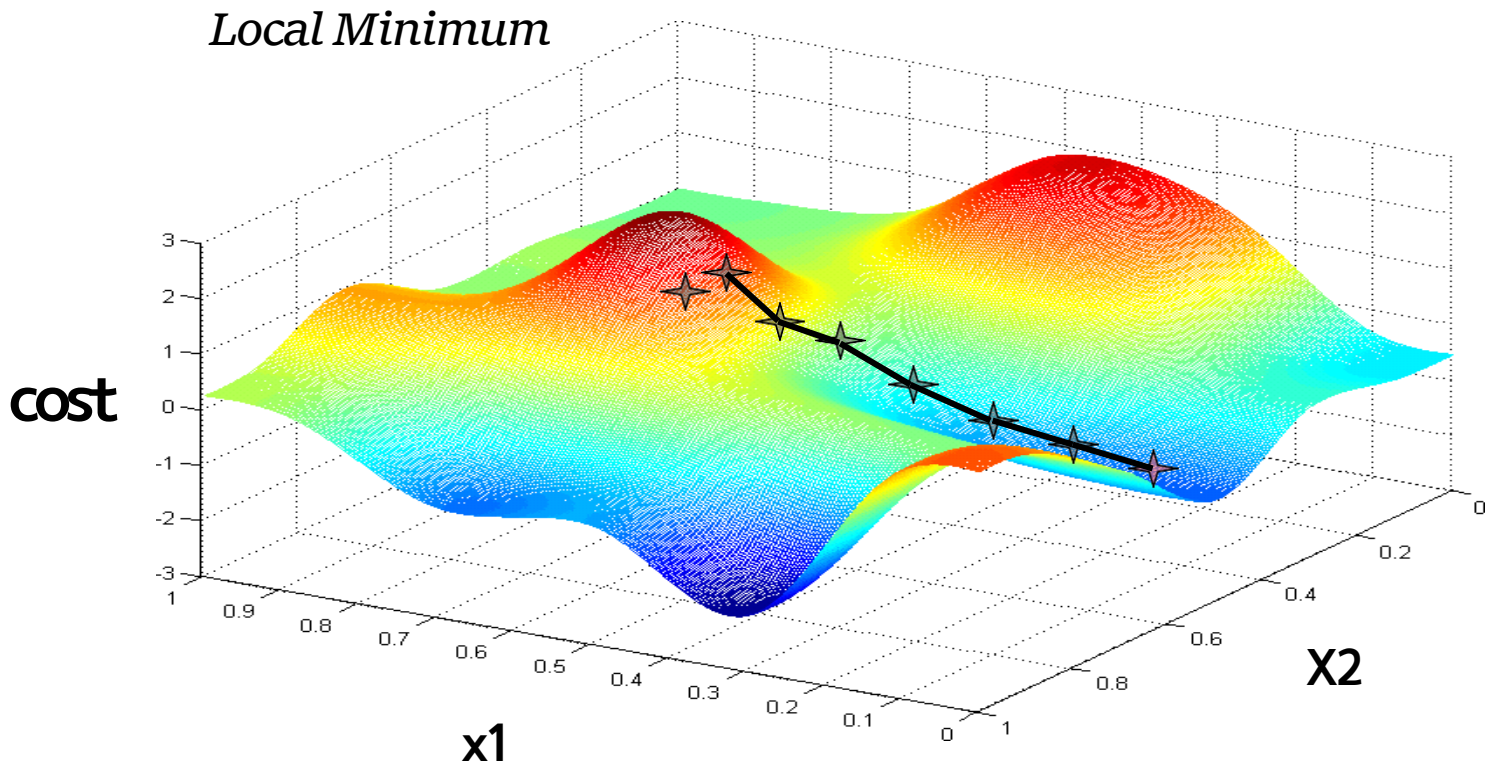
전체 데이터를 이용해 업데이트



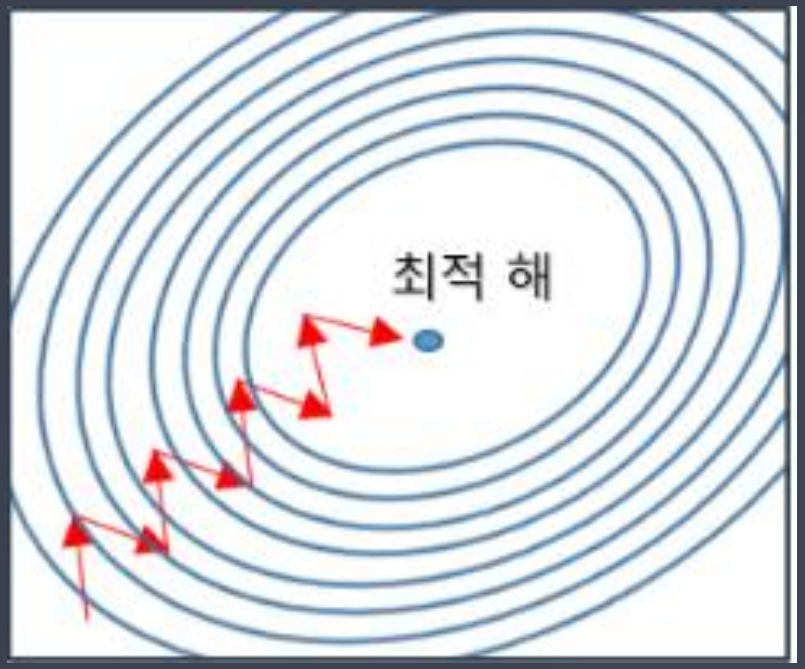
확률적경사하강법
(Stochastic Gradient Descent)

확률적으로 선택된 일부 데이터를 이용해 업데이트

경사하강법(Gradient Descent Algorithm)

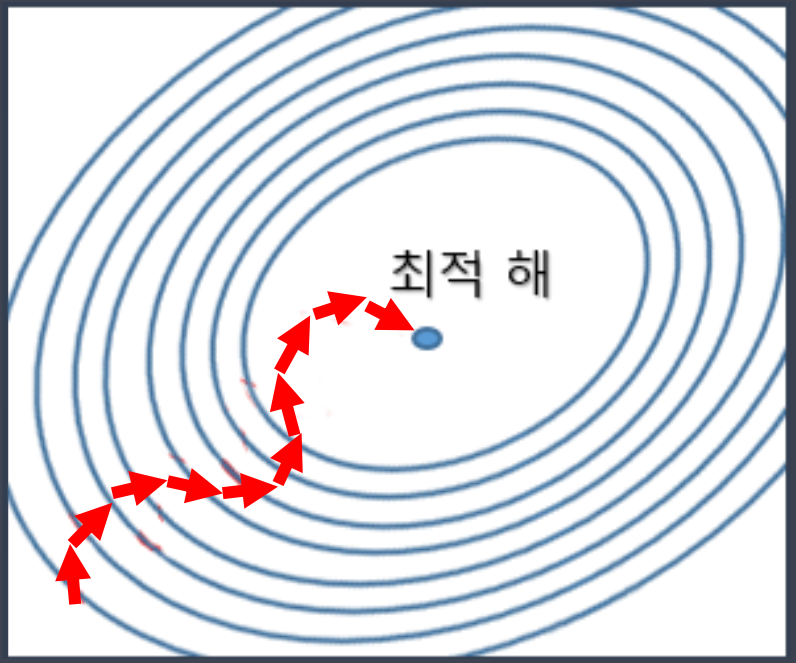


경사하강법(Gradient Descent Algorithm)



확률적경사하강법
(Stochastic Gradient Descent)

확률적으로 선택된 일부 데이터를 이용해 업데이트



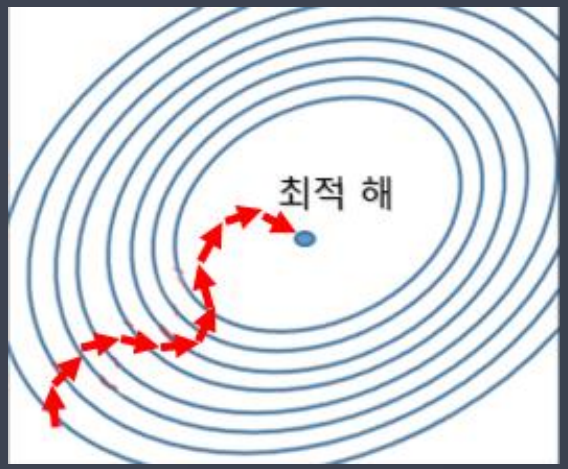
모멘텀
(Momentum)

경사 하강법에 관성을 적용해 업데이트

경사하강법(Gradient Descent Algorithm)

장·단점 (Momentum)

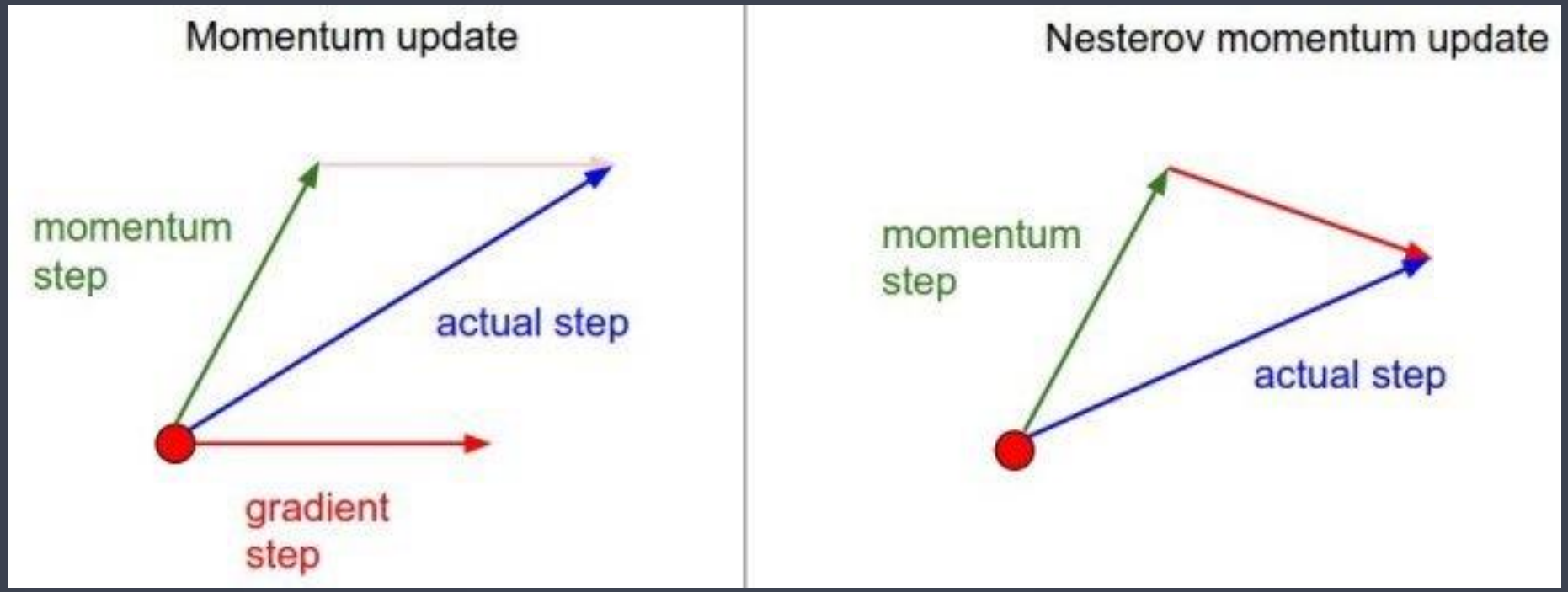
- 가중치를 수정하기전 이전 방향을 참고하여 업데이트.
- α는 Learning Rate, m은 momentum 계수 (보통 0.9)
- 지그재그 형태로 이동하는 현상이 줄어든다.



$$V(t) = m * V(t-1) - \alpha \frac{\partial}{\partial w} Cost(w)$$

$$W(t+1) = W(t) + V(t)$$

경사하강법(Gradient Descent Algorithm)



네스테로프 모멘텀

(Nesterov Accelrated Gradient)

개선된 모멘텀 방식

경사하강법(Gradient Descent Algorithm)

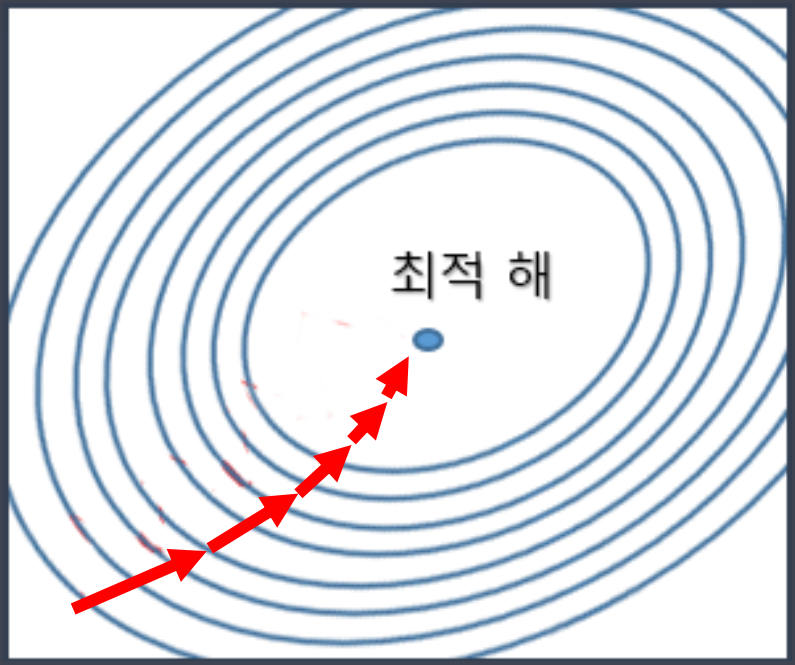
장·단점 (NAG)

- 업데이트시 모멘텀 방식으로 먼저 더한 다음 계산.
- 미리 해당 방향으로 이동한 뒤 그래디언트를 계산하는 효과.
- 불필요한 이동을 줄일 수 있다.

$$V(t) = m * V(t-1) - \alpha \frac{\partial}{\partial(w + m * V(t-1))} Cost(w)$$

$$W(t+1) = W(t) + V(t)$$

경사하강법(Gradient Descent Algorithm)



아다그리드

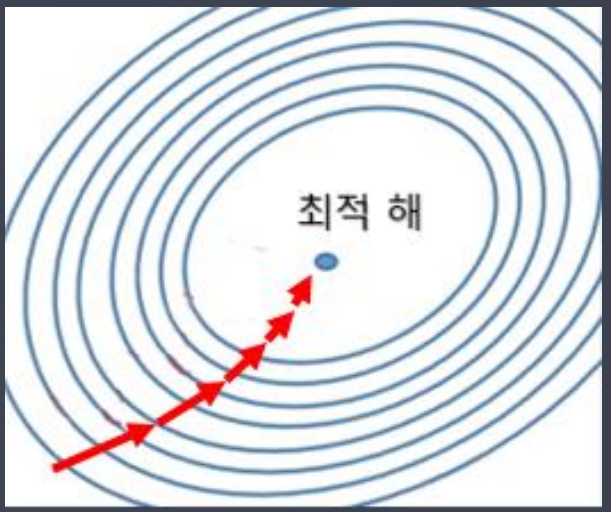
(Adaptive Gradient)

학습률 감소 방법을 적용해 업데이트

경사하강법(Gradient Descent Algorithm)

장·단점 (Adagard)

- 학습을 진행하면서 학습률을 점차 줄여가는 방법.
- 처음에는 크게 학습하다가 조금씩 작게 학습한다.
- 각각의 가중치에 맞춤형 값을 만든다.

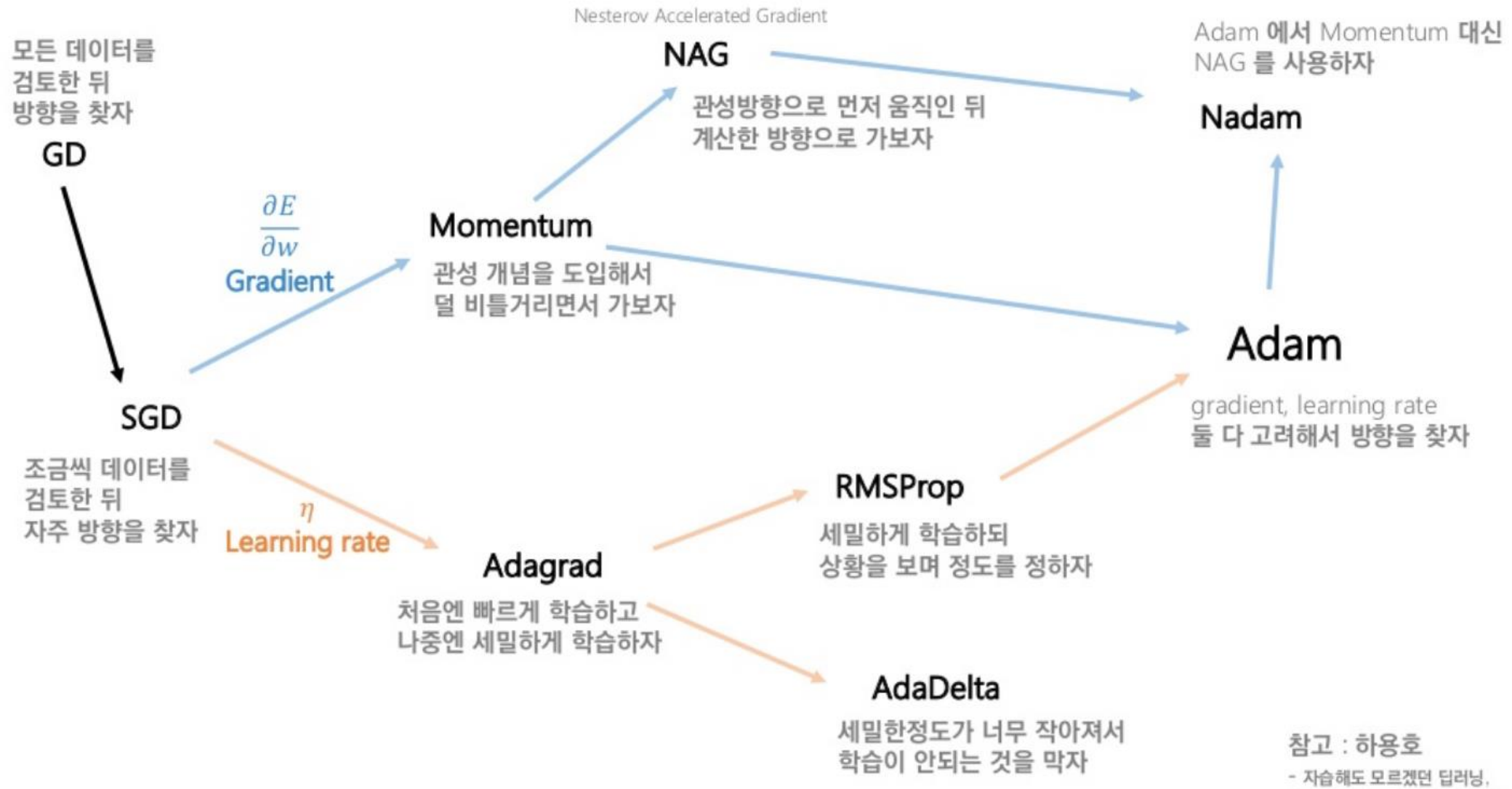


$$G(t) = G(t-1) + \left(\frac{\partial}{\partial w(t)} Cost(w(t))\right)^2$$

$$= \sum_{i=0}^{t} \left(\frac{\partial}{\partial w(i)} Cost(w(i))\right)^2$$

$$W(t+1) = W(t) - \alpha * \frac{1}{\sqrt{G(t)+\epsilon}} * \frac{\partial}{\partial w(i)} Cost(w(i))$$

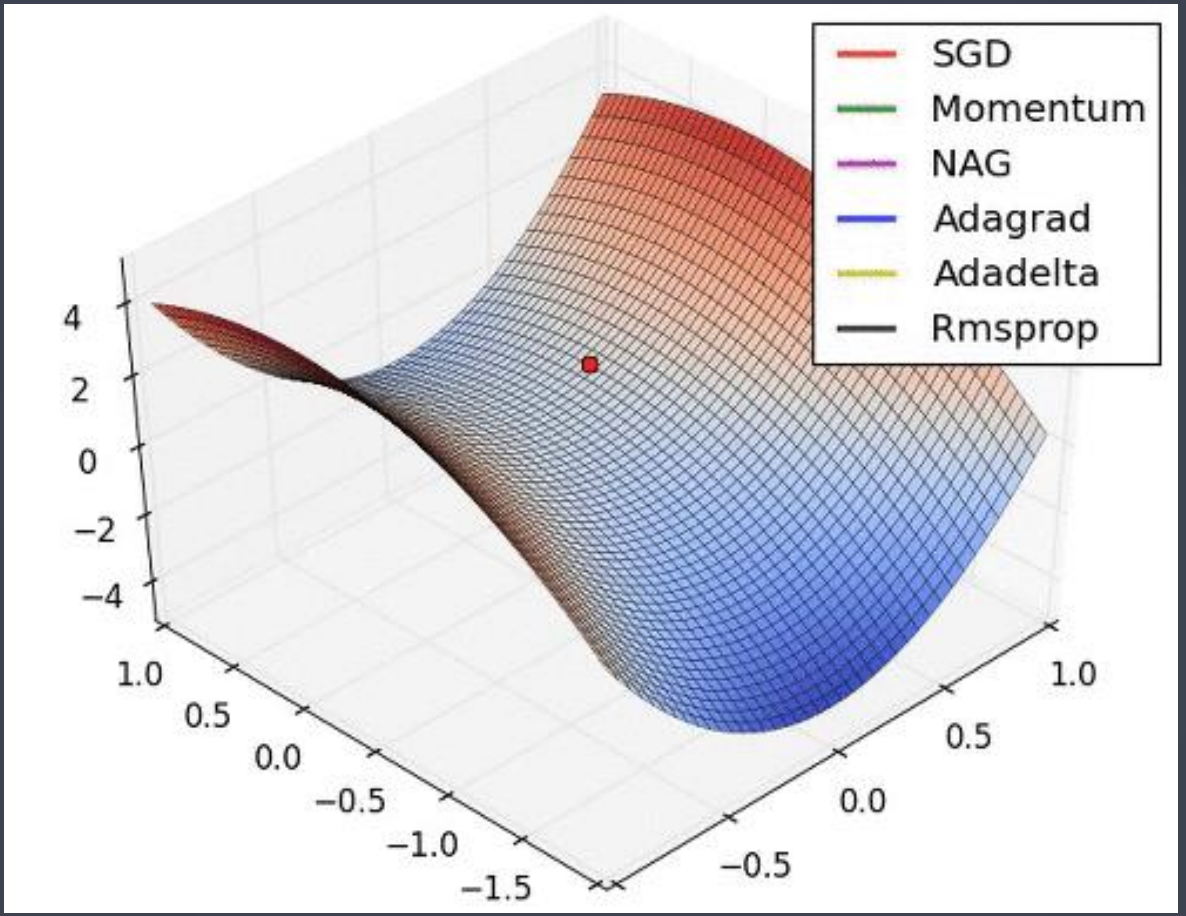
경사하강법(Gradient Descent Algorithm)



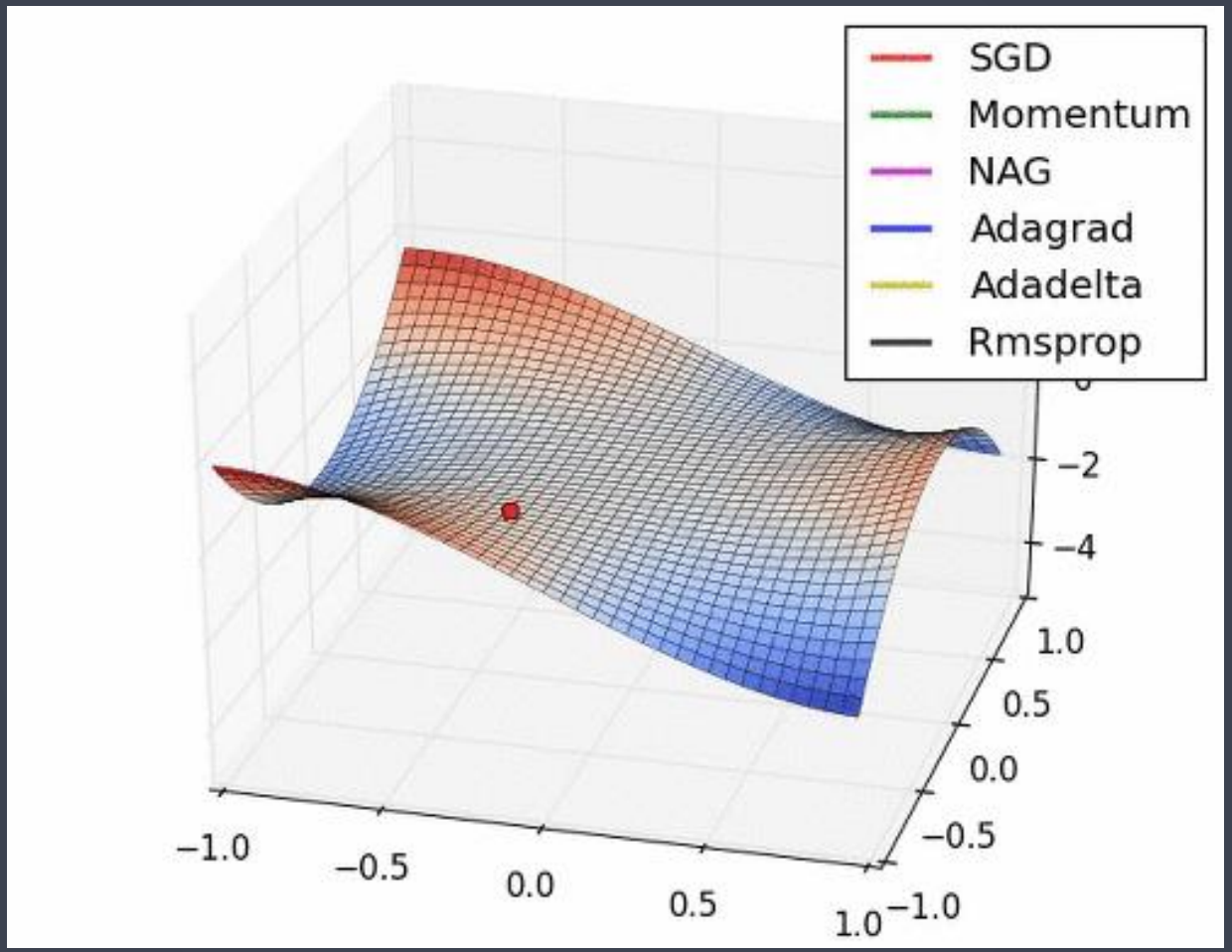
참고 : 하용호

- 자습해도 모르겠던 딥러닝,

경사하강법(Gradient Descent Algorithm)



경사하강법(Gradient Descent Algorithm)



경사하강법(Gradient Descent Algorithm)

Keras

```
from keras import optimizers

gradient_optimizer = optimizers.SGD(lr=0.01, momentum=0.9)

model.compile(loss='mean_squared_error',optimizer=gradient_optimizer)
```

Momentum

```
from keras import optimizers

gradient_optimizer = optimizers.SGD(lr=0.01, momentum=0.9, nesterov=True)

model.compile(loss='mean_squared_error',optimizer=gradient_optimizer)
```

NAG

```
from keras import optimizers

gradient_optimizer = optimizers.adam(lr=0.01)

model.complie(loss="mean_squared_error", optimizers=gradient_optimizer)
```

Adam

Adagard, RMSprop, Adam 등은 이름으로 지정가능

딥러닝(Deep Learning)

가중치 초기화

- LeCun 초기화 : 98년도에 얀 르쿤이 제기한 방법으로 최근에는 Xavier나 He 초기화 방식에 비해 덜 사용된다.
- Xavier 초기화 : 케라스에서는 glorot이라는 이름으로 되어있는데, 일반적으로는 Xavier Initialization이라고 알려져 있다.
- He 초기화 : 2015년에 제안한 가장 최신의 초기화 방식이다. 수식을 보면 Xavier Initialization을 조금 개선한 것인데, 경험적으로 더 좋은 결과를 내었다고 한다.

딥러닝(Deep Learning) 실습

Keras로 MNIST 손글씨 학습하기
+가중치, 활성화함수, 옵티마이저, 규제적용

딥러닝(Deep Learning) 실습

Keras활용 보스턴 주택 값 예측 신경망 실습

CRIM: 타운별 1인당 범죄율
ZN: 25,000 평방 피트가 넘는 다수 거주자를 위한 주거 지역 비율
INDUS: 타운당 비소매 사업 면적 비율
CHAS: 길이 강과 경계하면 1, 아니면 0
NOX: 산화질소 농도(0.1ppm 단위)
RM: 주택당 평균 방수
AGE: 1940년 이전에 건축된 주인이 거주하는 주택 비율
DIS: 보스턴의 5개 고용 센터까지 가중치 거리
RAD: 방사형 고속도로로의 접근성 지수
TAX: $10,000 당 최대 자산 가치 비율
PTRATIO: 타운별 학생-교사 비율
B: 아프라카계 미국인의 비율을 *Bk*라고 할 때, *1000 x (Bk - 0.63)^2* 의 값
LSTAT: 인구의 낮은 백분율
MEDV: 주인이 거주하는 주택 가격의 중간값($1,000 단위)

딥러닝(Deep Learning) 실습

Keras활용 보스턴 주택 값 예측 신경망 실습 +모델검증, 학습중단, 모델저장, 모델로드