딥러닝
(Deep Learning)

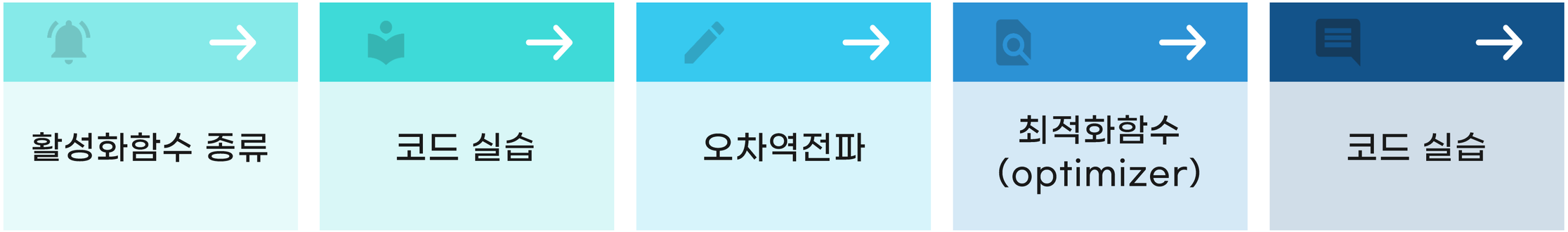
최성우

학습목표

- 활성화함수의 종류를 알 수 있다.
- 오차역전파의 개념을 알 수 있다.
- 경사하강법의 종류를 알 수 있다.
- 최적화함수의 종류를 알 수 있다.

수업 흐름도

스마트인재개발원
Smart Human Resources Development



스마트인재개발원
Smart Human Resources Development

활성화함수 종류

자극에 대한 반응 여부와 그 정도를 결정하는 함수

활성화함수(Activation function)

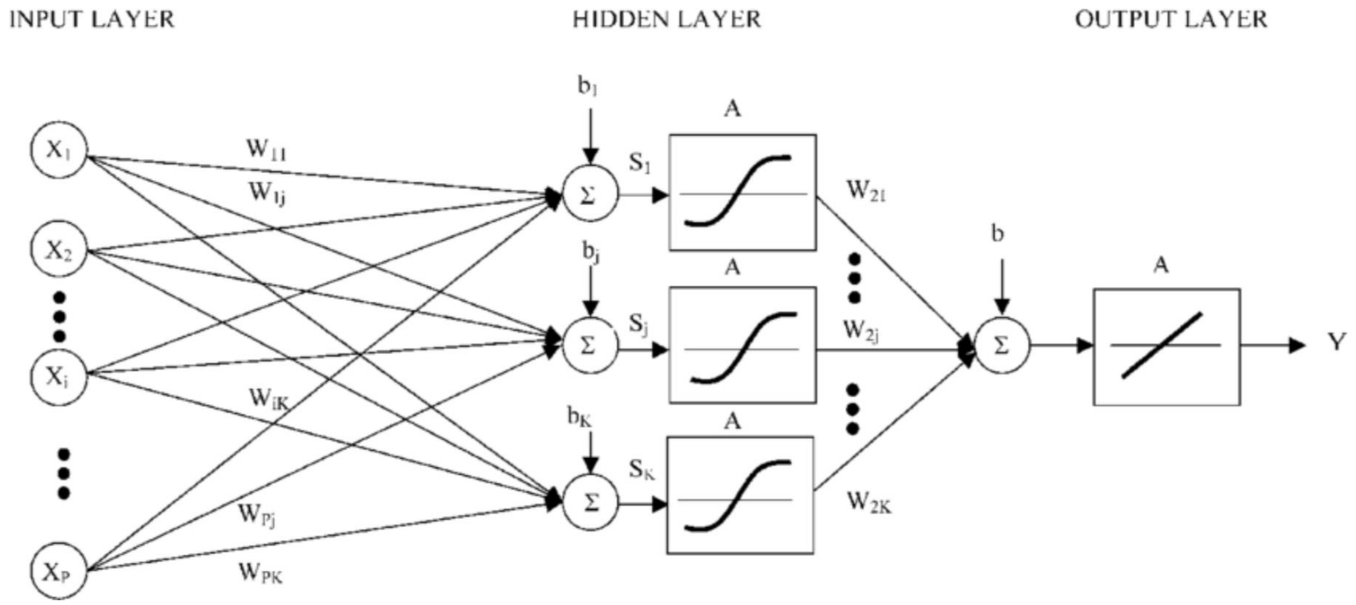
활성화함수

- 인공 신경망은 인간의 신경망을 본따 한 층의 신호를 다음 층으로 그대로 전달하지 않고 활성화 함수를 거친 후에 전달함
- 사람의 신경망 속 뉴런들도 모든 자극을 전부 다음 뉴런으로 전달하는 것이 아니라 역치 이상의 자극만 전달하게 됨
- 활성화 함수는 이런 부분까지 사람과 유사하게 구현하여 사람처럼 사고하고 행동하는 인공지능 기술을 실현하기 위해 도입됨
- 또한 선형모델을 기반으로 하는 딥러닝 신경망에서 분류 문제를 해결하기 위해서 비선형 활성화 함수가 필요함

활성화함수(Activation function)

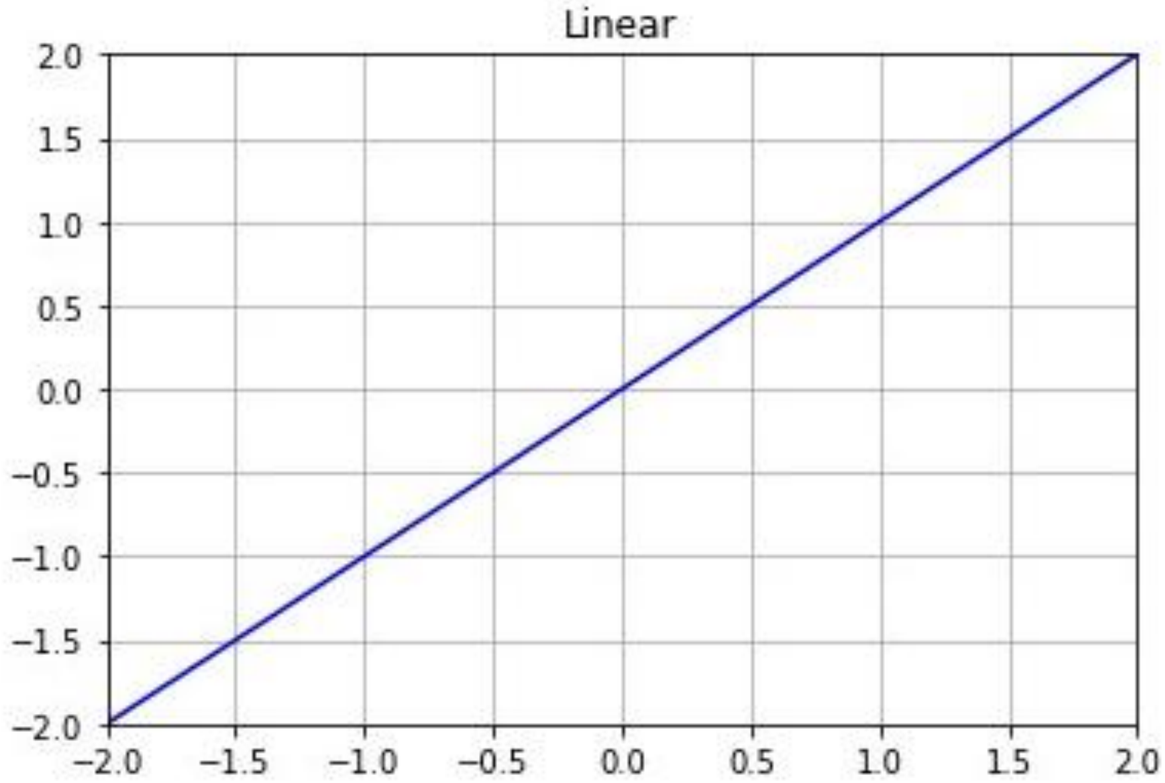
스마트인재개발원
Smart Human Resources Development

층에 따라 다른 활성화 함수를 사용할 수 있다



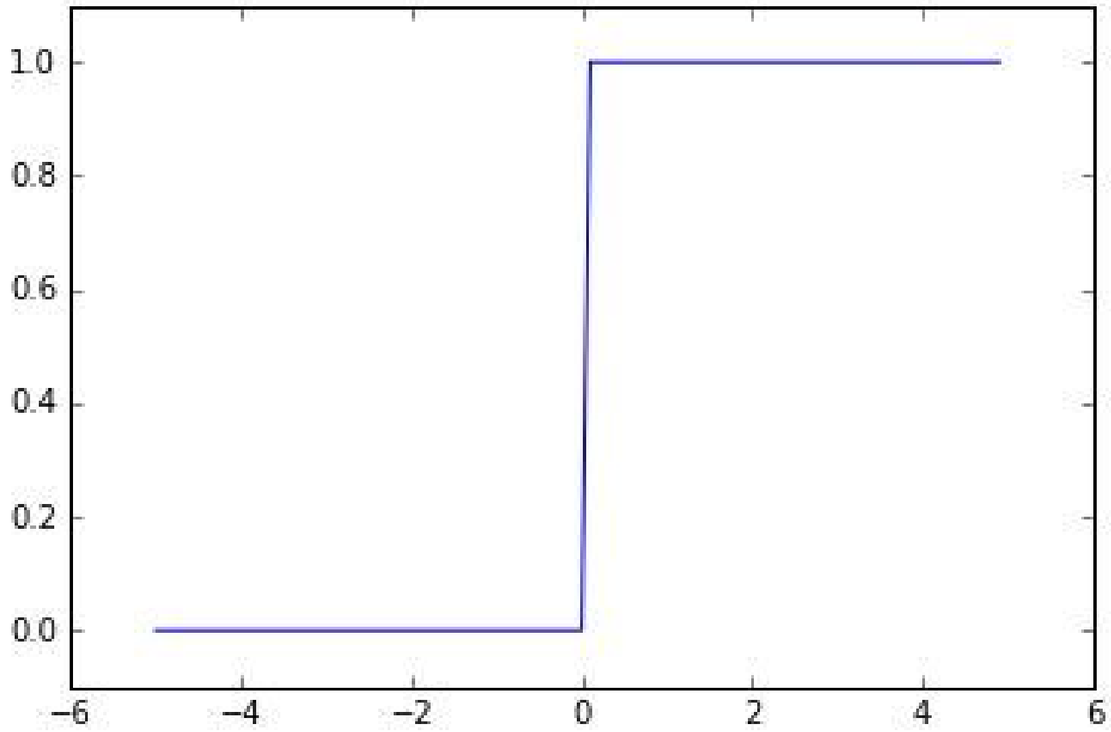
활성화함수(Activation function)

Linear function(선형함수=항등함수) → 회귀



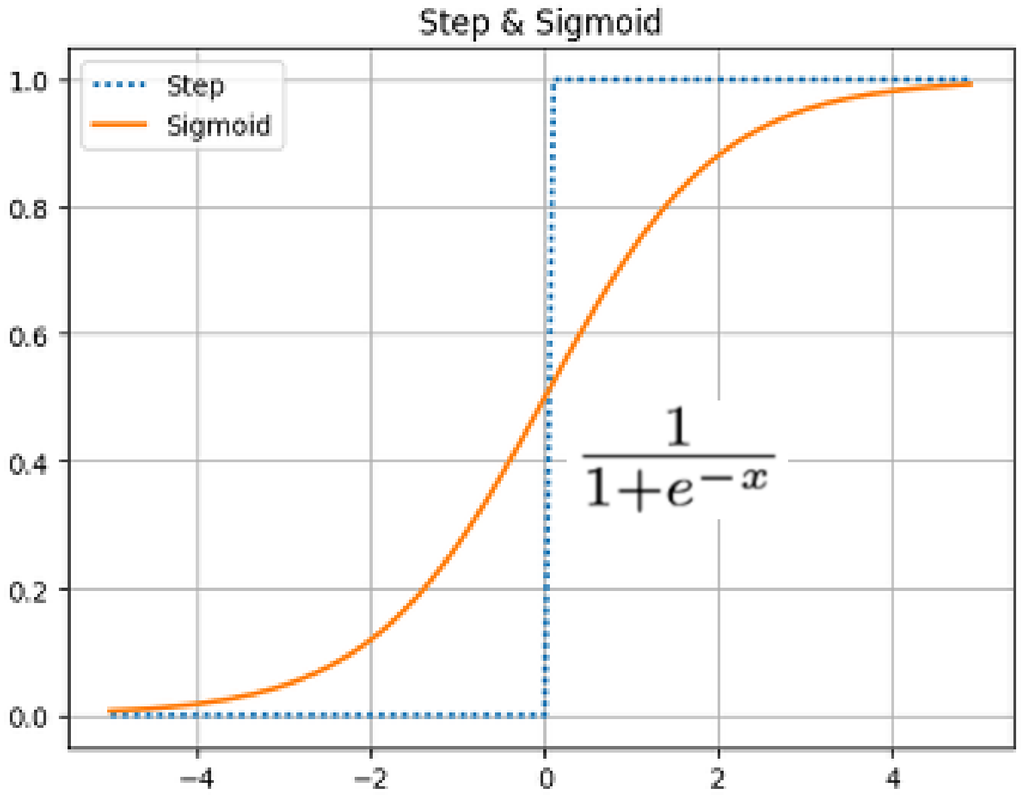
활성화함수(Activation function)

Step function(계단 함수) → 분류의 초기 활성화 함수



활성화함수(Activation function)

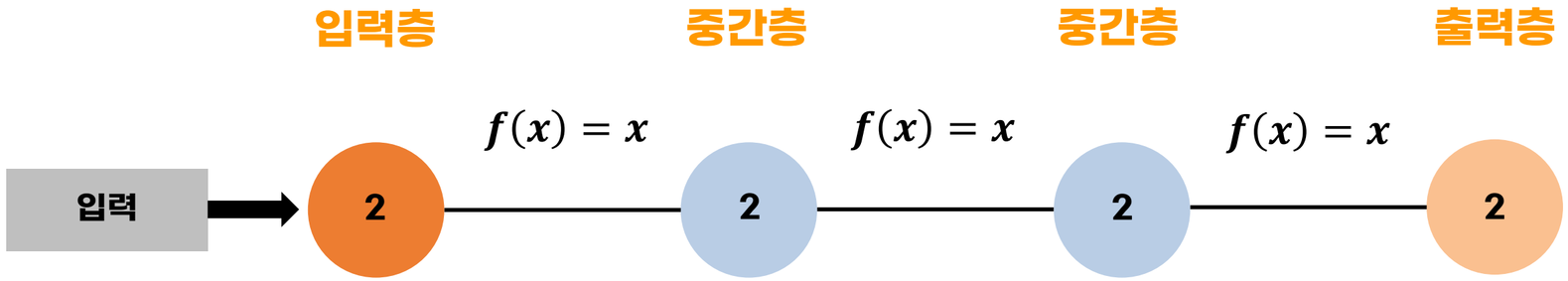
Sigmoid 함수 → 이진분류

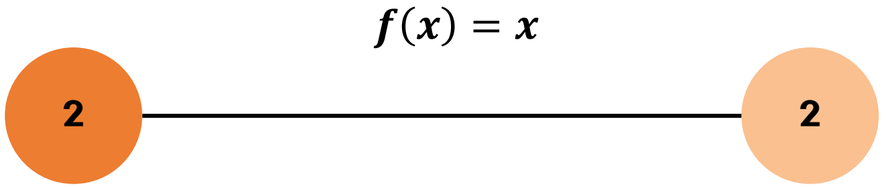


활성화함수(Activation function)

중간층에 활성화 함수로 비선형 함수를 사용하는 이유

- 중간층에 활성화 함수로 선형 함수(f(x) = x)를 사용하게 된다면





keras 맛보기 : 유방암 데이터 이진 분류

활성화함수(Activation function)

다중분류

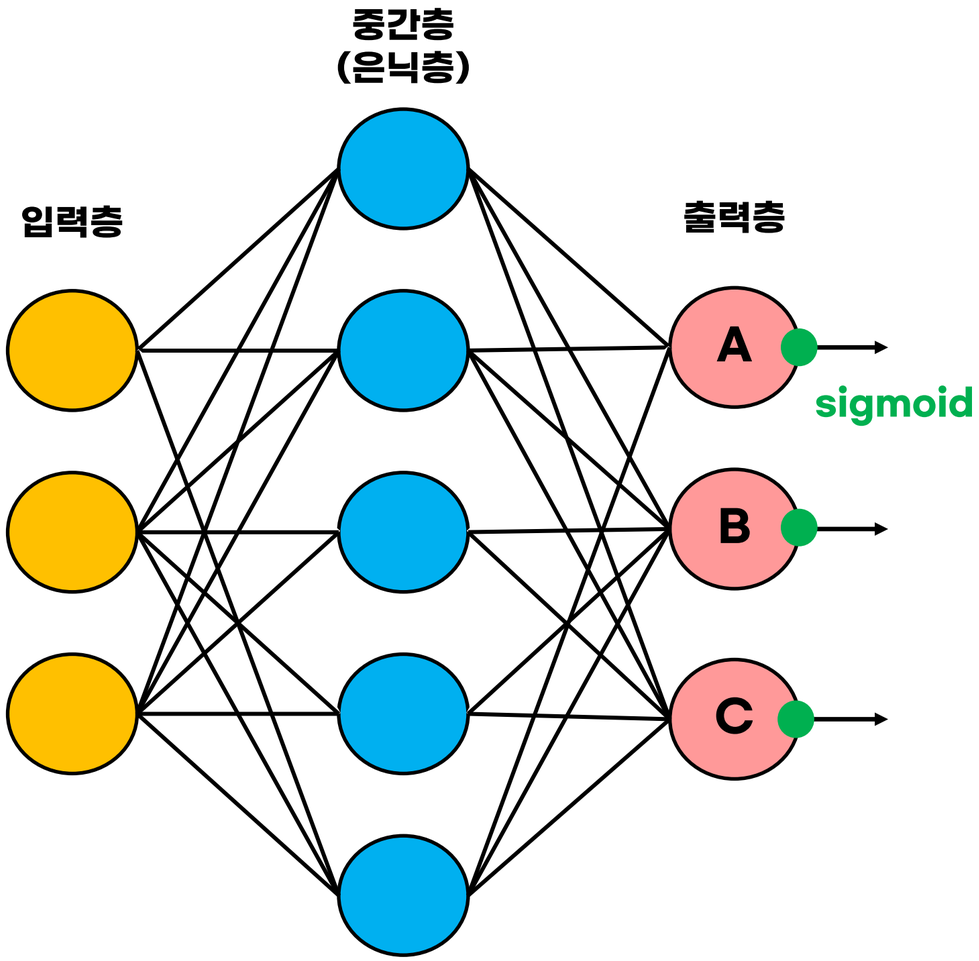
- 딥러닝 신경망에서 다중분류 문제를 해결하는 프로세스는 각 클래스에 대한 확률 값을 토대로 가장 높은 확률 값을 가지는 클래스로 최종 분류를 진행함

- 각 레이블의 확률들을 알기 위해 출력층 퍼셉트론 개수를 클래스 개수와 같게 맞춰야 함(하나의 퍼셉트론이 하나의 클래스에 대한 확률 값을 출력)

- 또한, 다중 분류 문제를 풀 경우 정답 데이터를 원 핫 인코딩 해야 함

- 신경망 학습을 위해서는 원 핫 인코딩 된 정보(0,1)와 출력층의 각 퍼셉트론이 예측한 확률(0~1)과의 오차를 바탕으로 신경망이 학습하게 됨

활성화함수(Activation function)



	A	B	C
원핫	0	0	1
레이블	1	2	3

정답이 C라고 가정

예측 확률값	원핫인코딩 오차	레이블 인코딩 오차
0.4	0.4	0.6
0.7	0.7	1.3
0.9	0.1	2.1

오차를 제대로 파악하기 위해서는 예측확률과
실제 정답의 범위(0~1)가 같아야 비교가능

활성화함수(Activation function)

Softmax 함수 → 다중분류

- 딥러닝 다중분류에서 레이블 값에 대한 각 퍼셉트론의 예측 확률의 합을 1로 설정
- sigmoid에 비해 예측 오차의 평균을 줄여주는 효과

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

입력 값들의
지수함수의 합

활성화함수(Activation function)

Softmax 함수 코드 구현

```
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
```

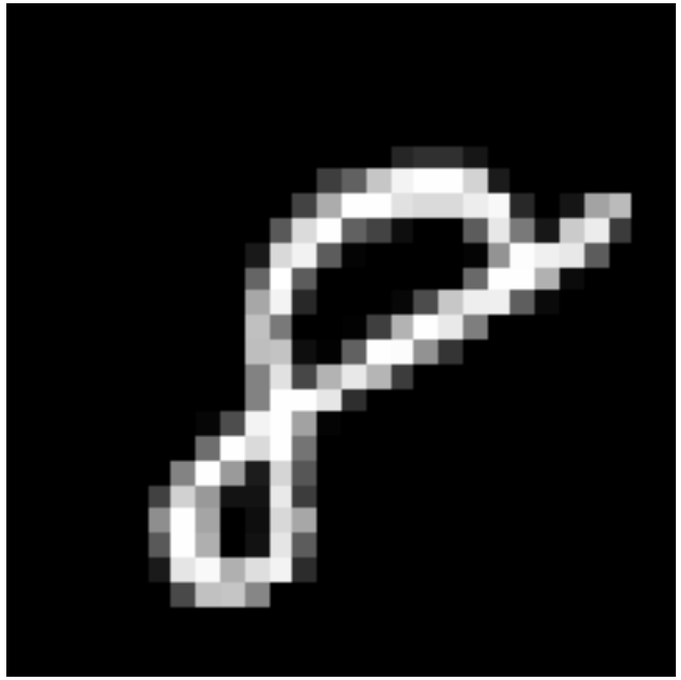
문제 유형에 따른 활성화함수, 손실함수

유 형	출력층 활성화 함수(activation)	손실함수(=비용함수) (loss func.)
회귀	linear(항등 함수)	MSE
2진 분류	sigmoid(로지스틱 함수)	binary_crossentropy
다중 분류	softmax(소프트맥스 함수)	categorical_crossentropy

keras 맛보기 : iris 데이터 다중 분류

MNIST 손글씨 이미지 데이터 분류 모델 만들기

코드 실습



```
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   38  48  48  22  0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   62  97  198 243 254 254 212 27  0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   67  172 254 254 225 218 218 237 248 40  0   21  164 187 0   0
0   0   0   0   0   0   0   0   0   0   0   89  219 254 97  67  14  0   0   92  231 122 23  203 236 59  0   0
0   0   0   0   0   0   0   0   0   0   25  217 242 92  4   0   0   0   0   4   147 253 240 232 92  0   0   0
0   0   0   0   0   0   0   0   0   0   101 255 92  0   0   0   0   0   0   105 254 254 177 11  0   0   0   0
0   0   0   0   0   0   0   0   0   0   167 244 41  0   0   0   7   76  199 238 239 94  10  0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   192 121 0   0   2   63  180 254 233 126 0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   190 196 14  2   97  254 252 146 52  0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   130 225 71  180 232 181 60  0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   130 254 254 230 46  0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   6   77  244 254 162 4   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   110 254 218 254 116 0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   131 254 154 28  213 86  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   66  209 153 19  19  233 60  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   142 254 165 0   14  216 167 0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   90  254 175 0   18  229 92  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   26  229 249 176 222 244 44  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   73  193 197 134 0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
```

패션이미지 데이터 분류 모델 만들기

스마트인재개발원
Smart Human Resources Development

오차역전파

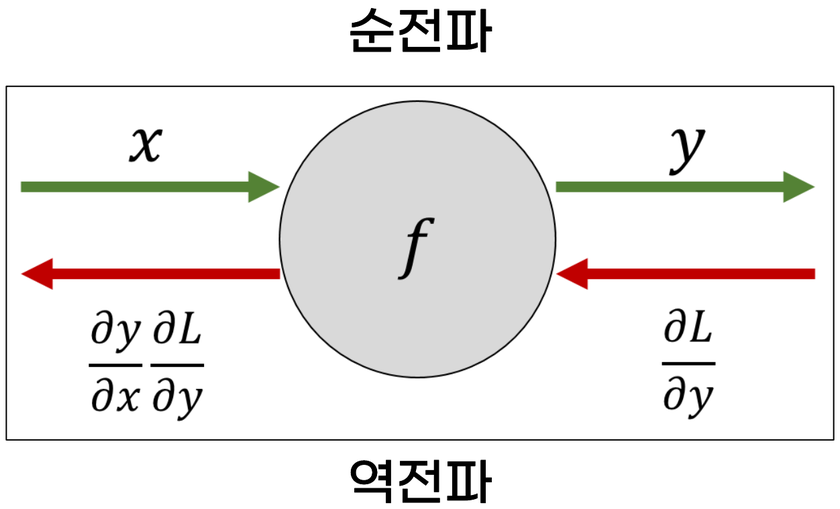
신경망이 학습되는 원리는 무엇일까?

오차역전파(Back propagation)

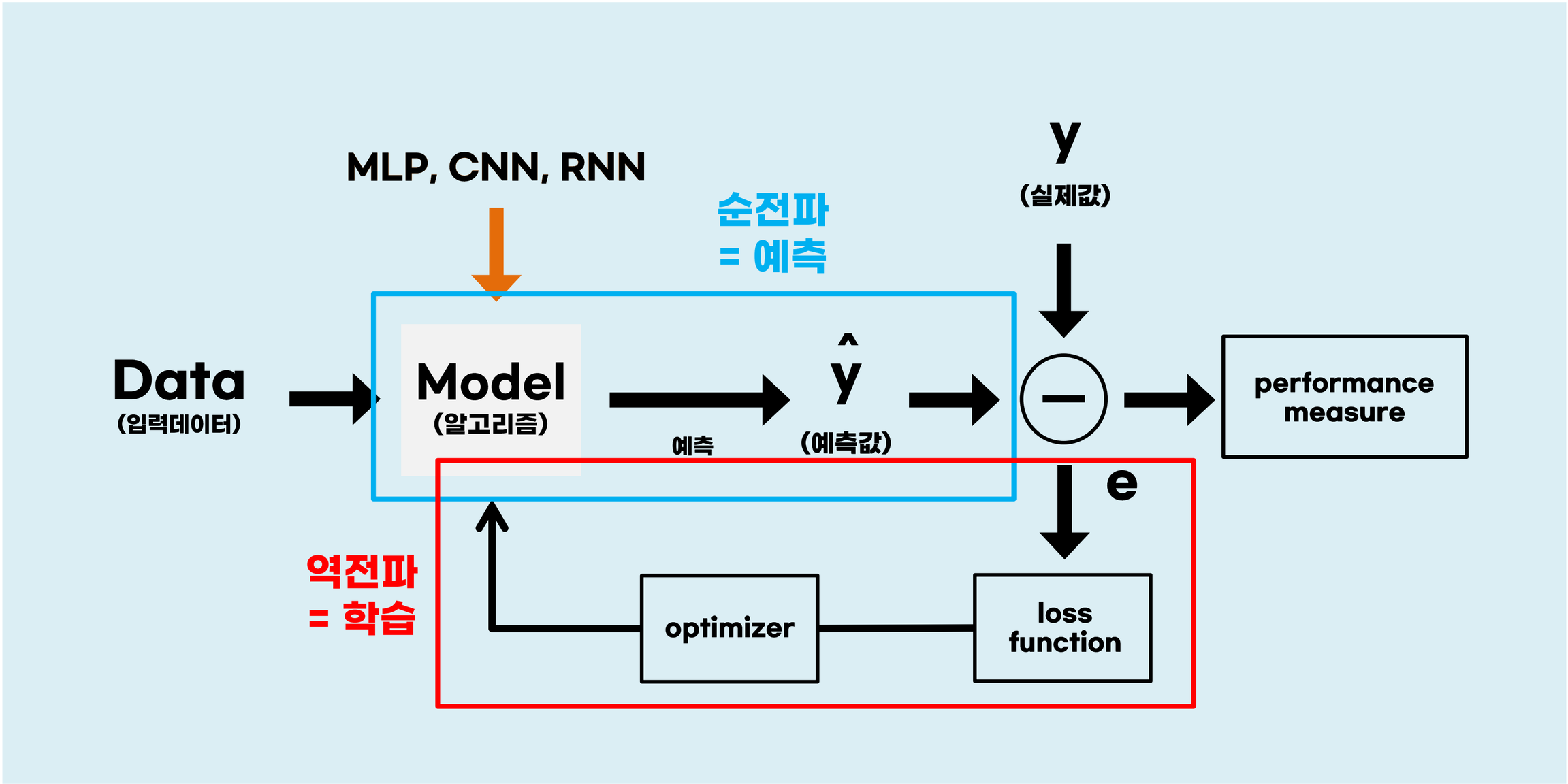
오차역전파(역전파)

- 순전파 : 입력 데이터를 입력층에서부터 출력층까지 정방향으로 이동시키며 출력 값을 예측해 나가는 과정

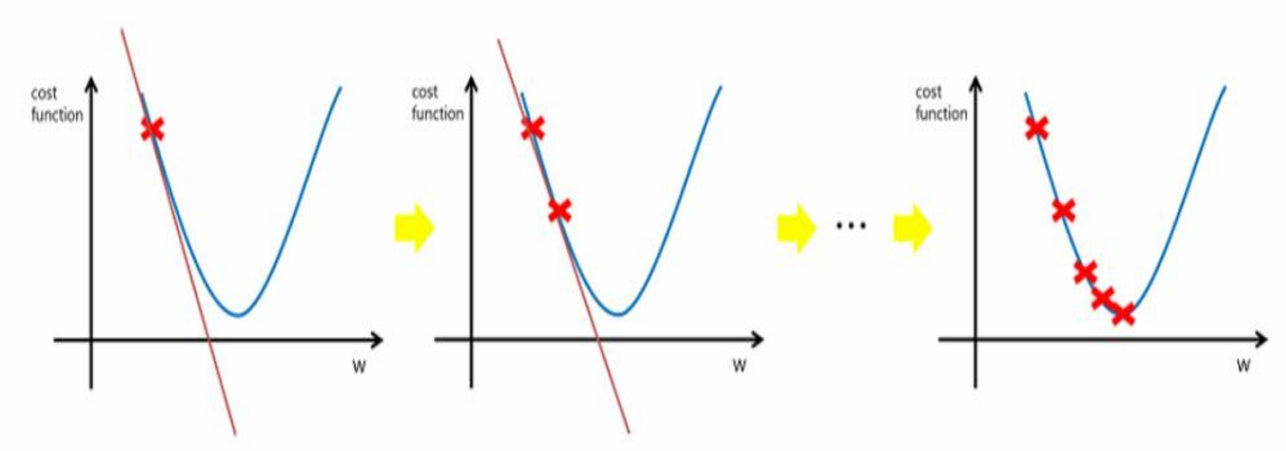
- 역전파 : 출력층에서 발생한 에러를 입력층 쪽으로 전파시키면서 최적의 결과를 위해 신경망의 가중치(w)를 학습해 나가는 과정



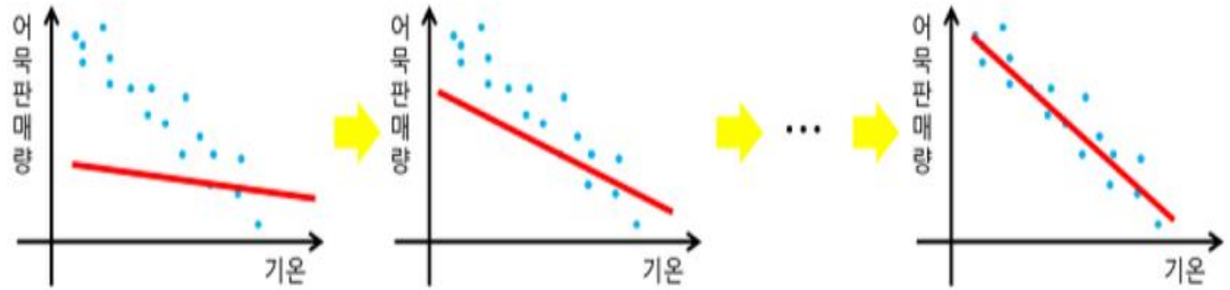
오차역전파(Back propagation)



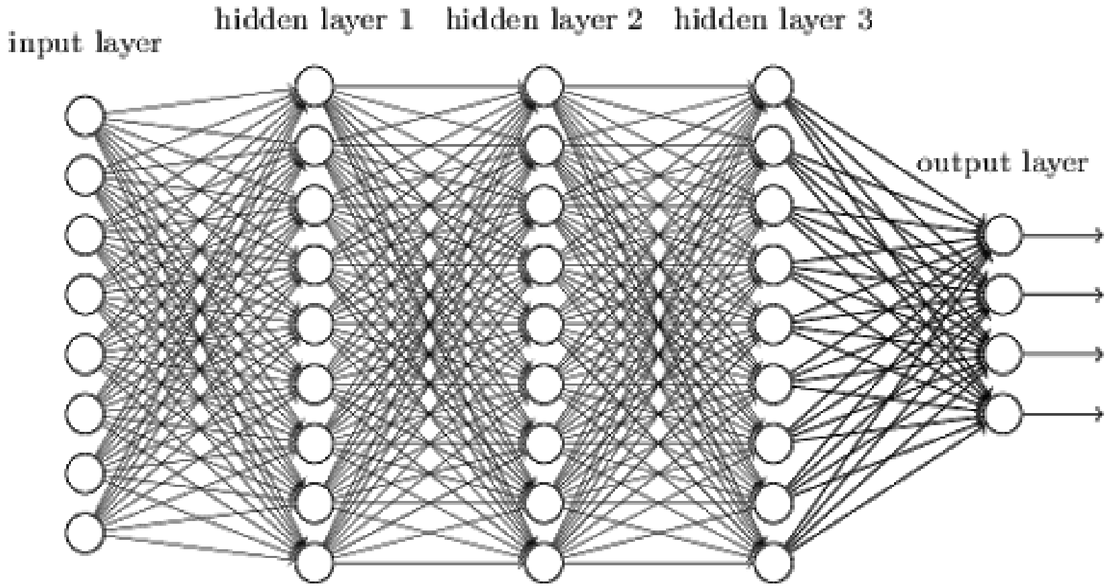
오차역전파(Back propagation)



최적의 w값을 찾는 과정



최적의 직선을 생성하는 과정



신경망 구조

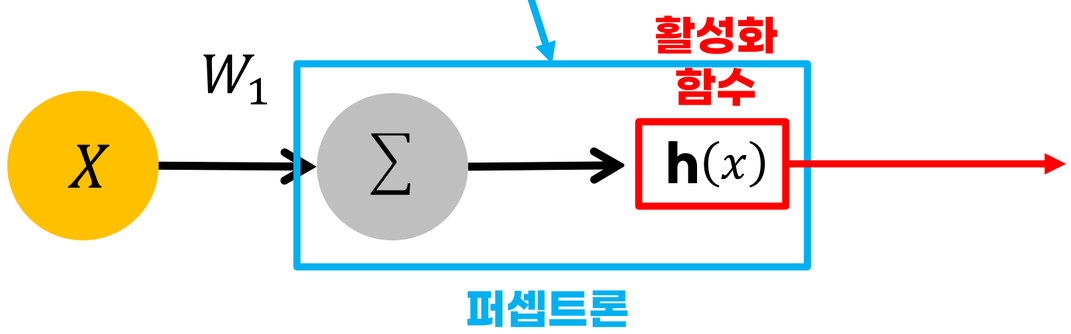
오차역전파(Back propagation)

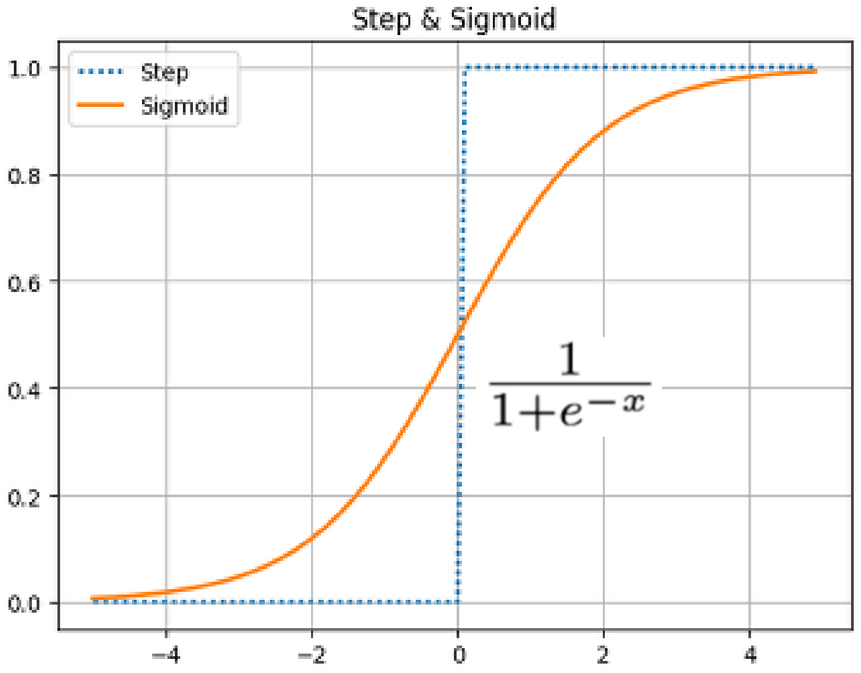
손실함수(loss) 미분

- 신경망 학습을 위해서는 경사하강법(loss함수의 미분)을 사용함

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

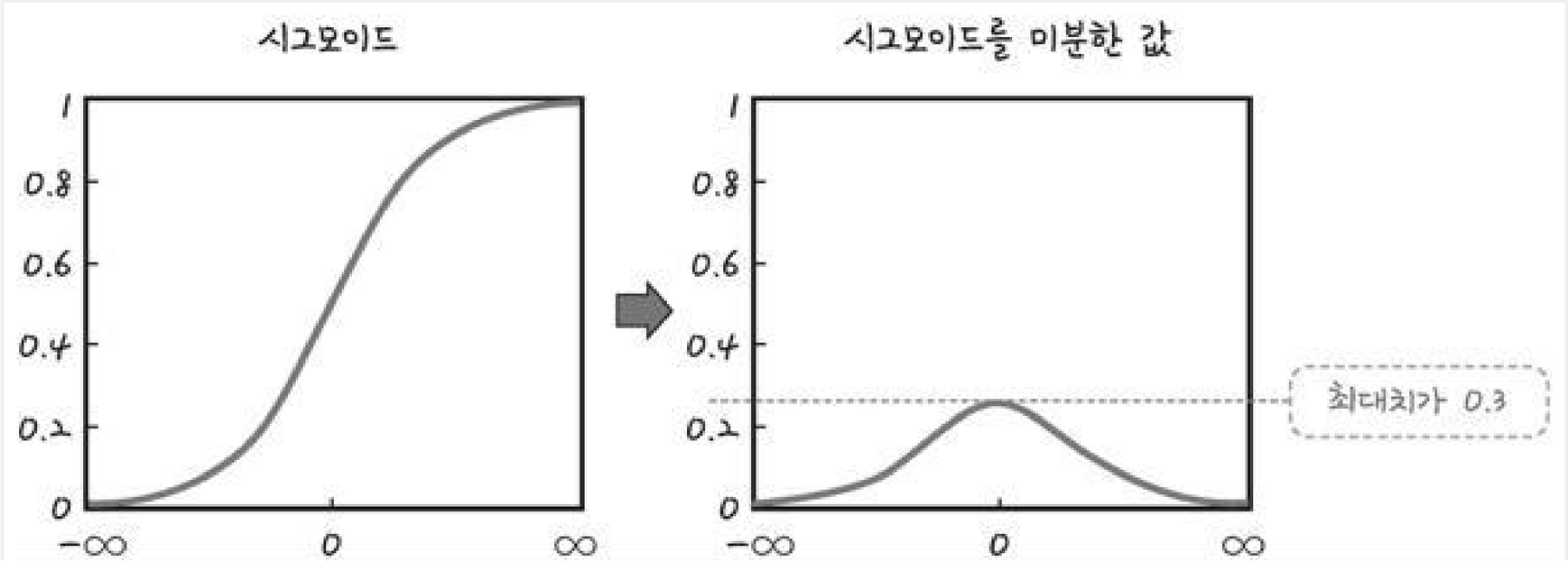
예측값





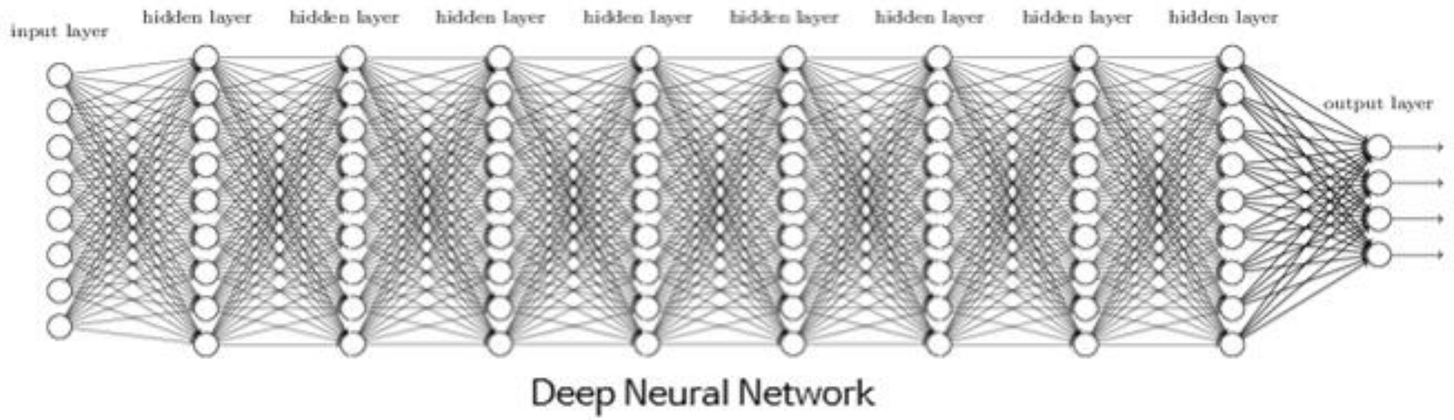
오차역전파(Back propagation)

Sigmoid 함수의 문제점

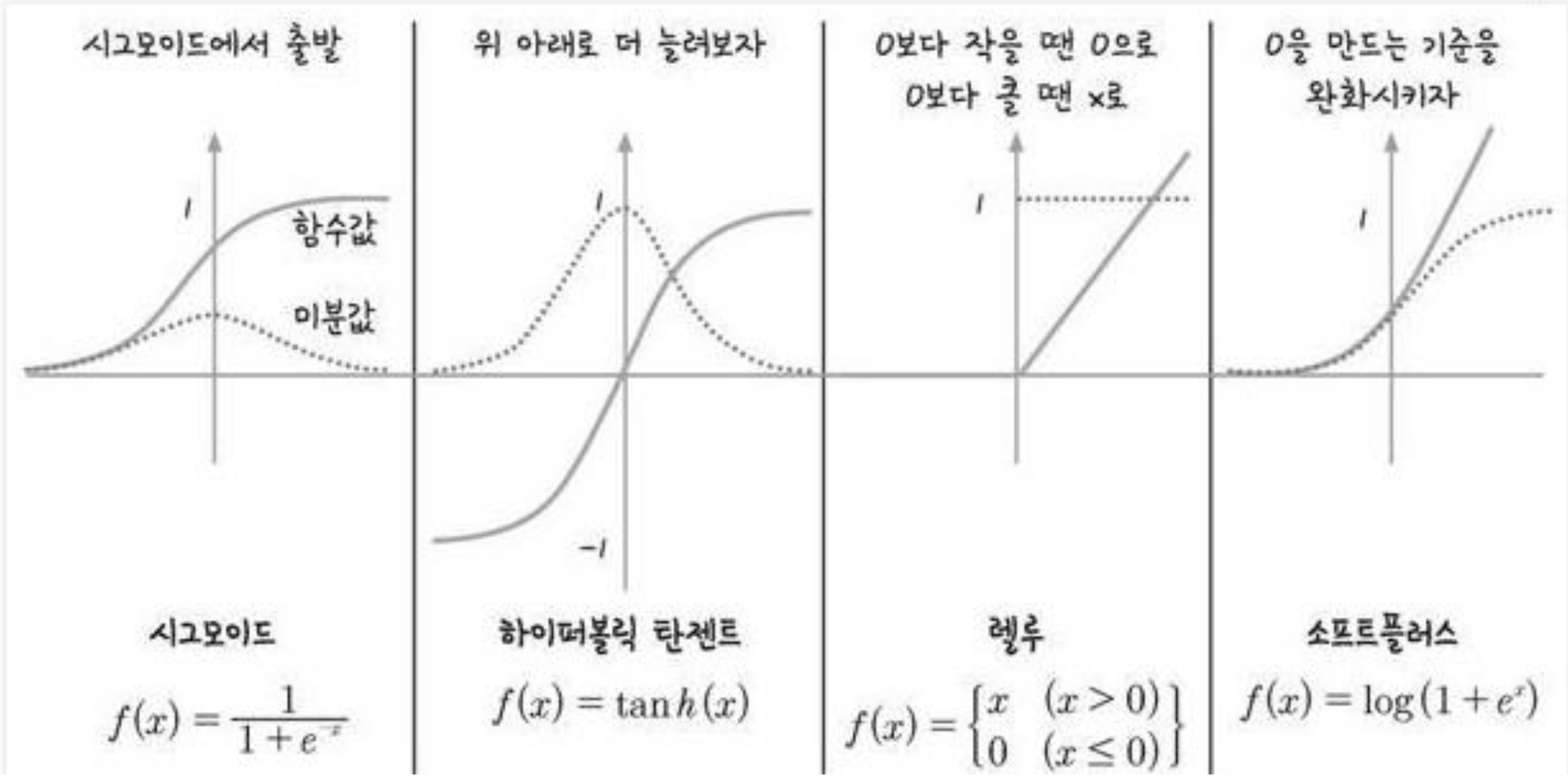


오차역전파(Back propagation)

Sigmoid 함수의 문제점
- Vanishing Gradient(기울기 소실 문제)



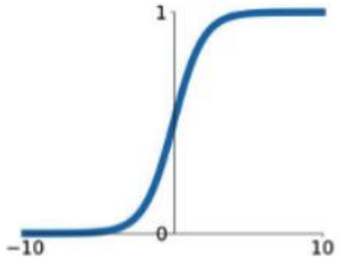
활성화함수(Activation function)



활성화함수(Activation function)

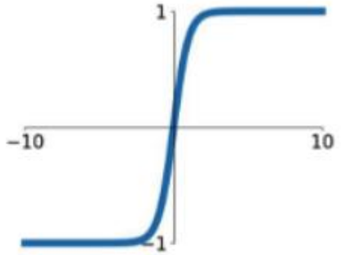
Sigmoid

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



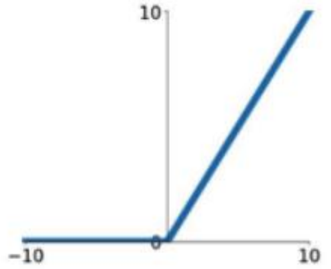
tanh

$$\tanh(x)$$



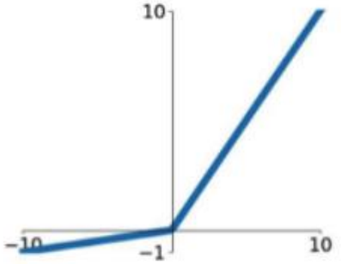
ReLU

$$\max(0, x)$$



Leaky ReLU

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

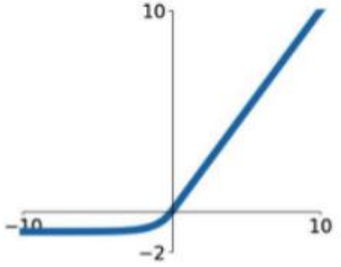


Maxout

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

ELU

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



스마트인재개발원
Smart Human Resources Development

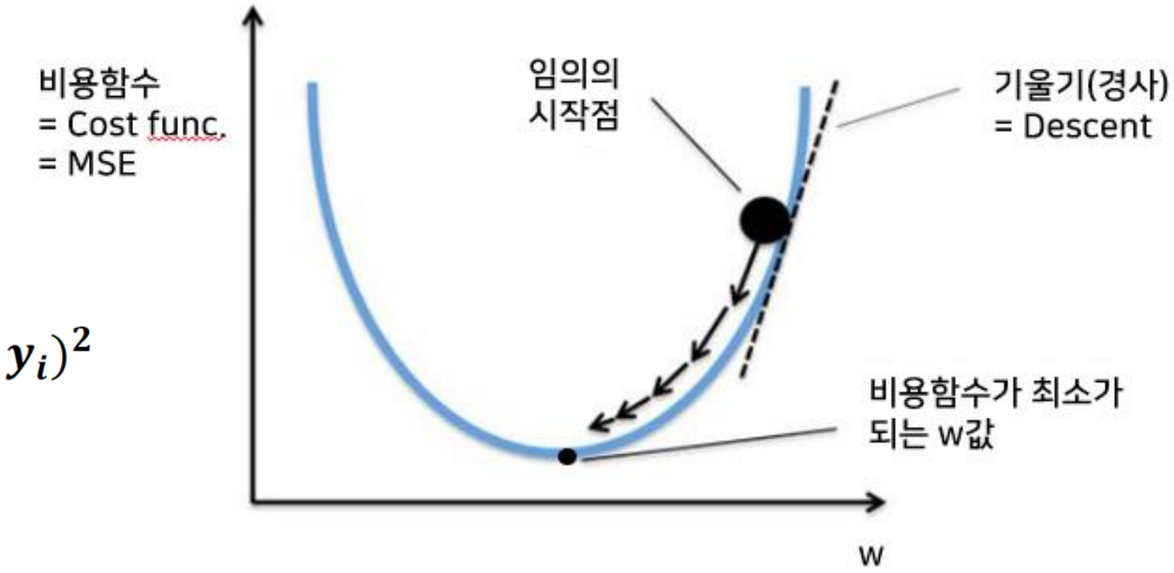
최적화함수

신경망을 좀 더 효과적으로 학습시키는 경사하강법을 찾자!

최적화함수(Optimizer)

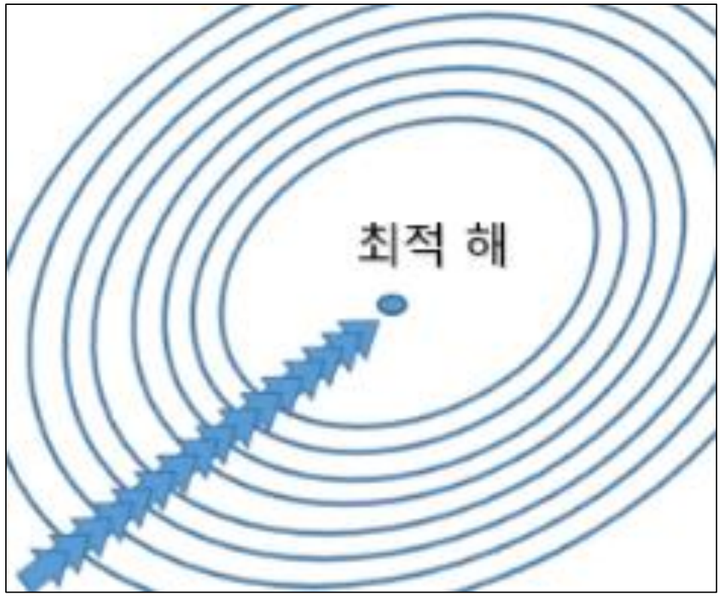
경사하강법(Gradient Descent Algorithm)

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



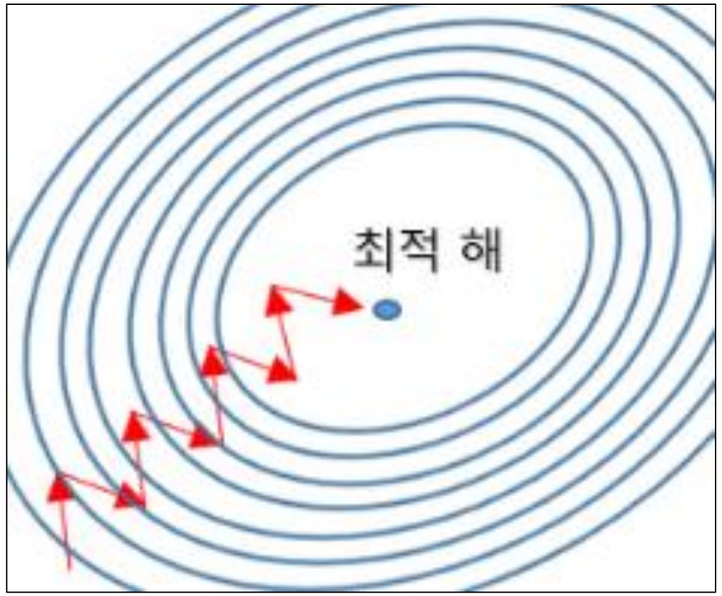
비용함수(=손실함수)의 기울기를 구하여 낮아지는 방향으로
계속 이동하여 파라미터 값을 최적화 시키는 방법

최적화함수(Optimizer)



경사하강법
(Gradient Descent)

모든 데이터를 이용해 업데이트



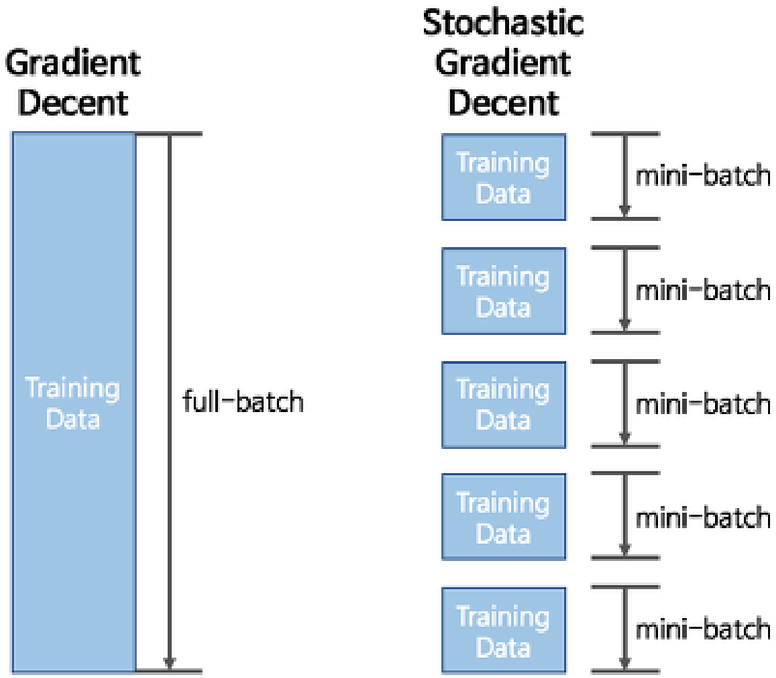
확률적 경사하강법
(Stochastic Gradient Descent)

확률적으로 선택된 하나의 데이터를
이용해 업데이트

최적화함수(Optimizer)

Batch_size

일반적으로 PC 메모리의 한계 및 속도 저하 때문에 대부분의 경우에는 한번에 epoch에 모든 데이터를 한꺼번에 집어 넣기가 힘듦



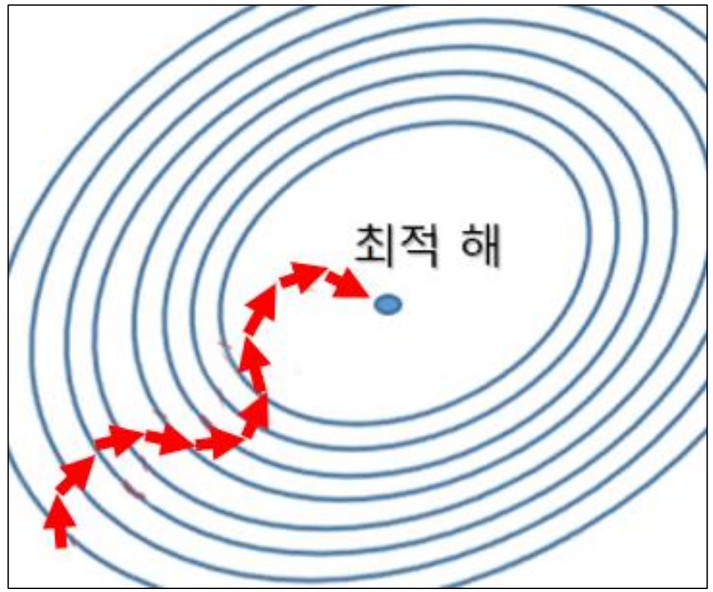
batch_size를 **줄임**
- 메모리 소모가 적음(저 사양)
- 학습 속도가 느림
- 속도가 느린 만큼 좀 더 정확하게 학습

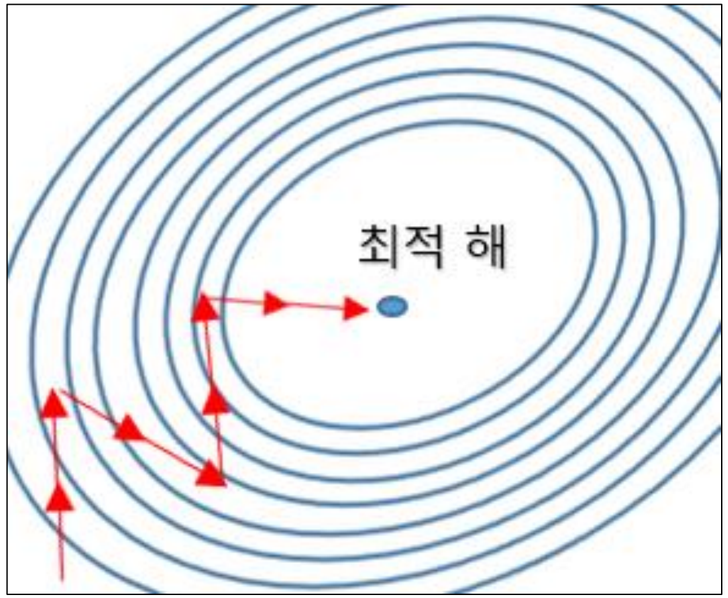
batch_size를 **높임**
- 메모리 소모가 큼, 학습속도 빠름
- 정확도가 비교적 낮음

→ batch_size의 디폴트 값은 32

최적화함수(Optimizer)

모멘텀(Momentum)





경사하강법에 관성의 법칙을 적용해 현재 batch뿐만
아니라 이전 batch의 데이터까지 업데이트에 반영

최적화함수(Optimizer)

모멘텀(Momentum)

- 가중치(w)를 수정하기 전 이전 방향을 참고하여 업데이트
- 경사하강법이 진행되면서 가중치가 잘 업데이트 됐다면 해당 방향으로 더 많이 업데이트하고, 아니라면 방향을 틀어서 최적 값에 맞게 좀 더 유연하게 수정해 나가는 방식
- 지그재그로 이동하는 현상이 줄어듦
- 수식에서 α는 Learning Rate, m은 momentum 계수로 보통 0.9로 설정함

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

네스테로프 모멘텀(Nesterov Accelrated Gradient)

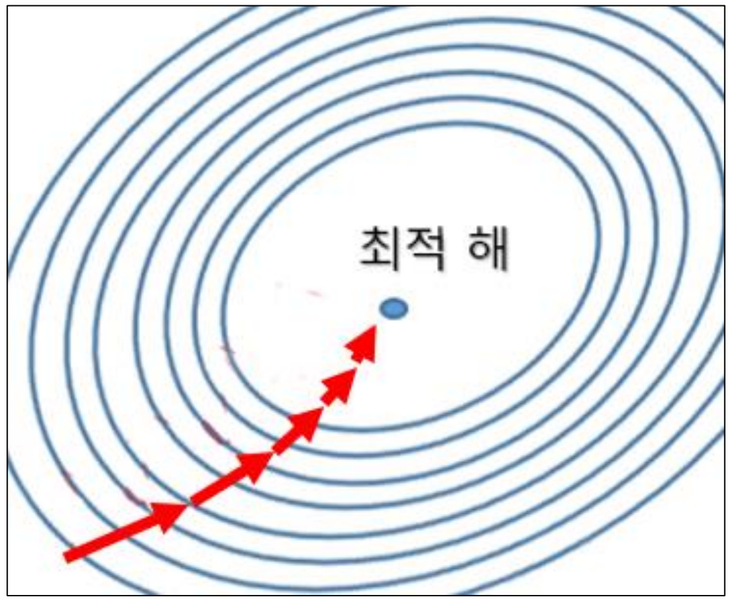
- 모멘텀을 개선한 방식으로 모멘텀 방식으로 먼저 더한 다음 계산
- 미리 해당 방향으로 이동한다고 가정하고 값을 계산해본 뒤 실제 업데이트에 반영
- 불필요한 이동을 좀 더 줄일 수 있음

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

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

최적화함수(Optimizer)

아다그래드(Adaptive Gradient)



학습률을 점점 감소시켜가며 업데이트

최적화함수(Optimizer)

아다그래드(Adaptive Gradient)

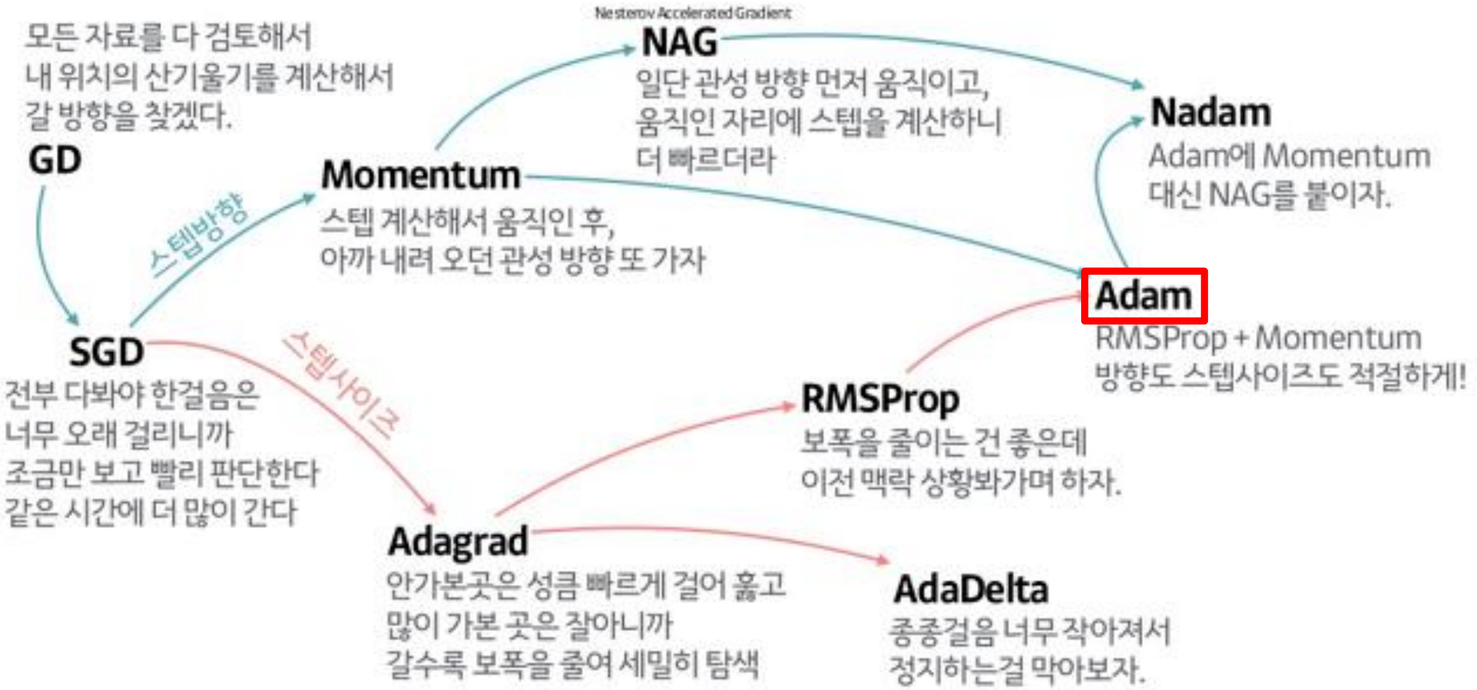
- 업데이트를 진행하면서 학습률을 점차 줄여가는 방식
- 처음에는 크게 학습하다가 조금씩 작게 학습함
- 학습을 빠르고 정확하게 할 수 있음

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

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

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

최적화함수(Optimizer)



출처 : https://www.slideshare.net/yongho/ss-79607172

최적화함수(Optimizer)


```
from tensorflow.keras import optimizers

opti = optimizers.SGD(learning_rate=0.01, momentum=0.9)

model.compile(loss='mse', optimizer=opti, metrics=['acc'])
```

Momentum

```
from tensorflow.keras import optimizers

opti = optimizers.SGD(learning_rate=0.01, momentum=0.9, nesterov=True)

model.compile(loss='mse', optimizer=opti, metrics=['acc'])
```

NAG

```
model.compile(loss ="mse", optimizer="Adam", metrics=["acc"])
```

Adam

Adagrad, RMSprop, Adam 등은 이름으로 지정 가능

패션이미지 데이터 분류 모델 만들기 (활성화함수, 최적화함수 변경)

다음시간에

복습 없이 전부를 이해하려는 것은 정신병 초기 증상이다