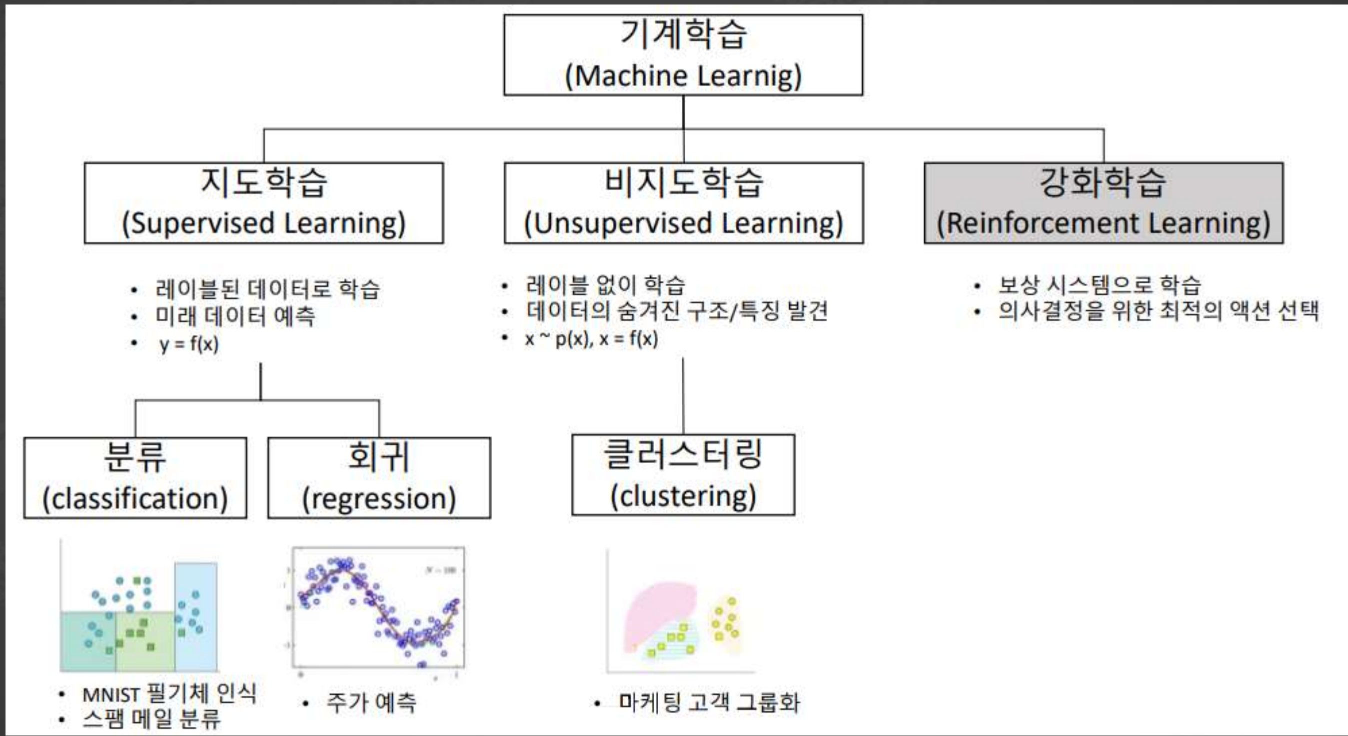
8/6

딥러닝



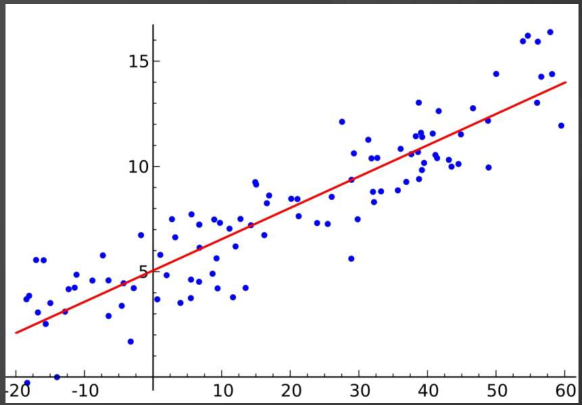
기계학습
(Machine Learnig)
지도학습
(Supervised Learning)
비지도학습
(Unsupervised Learning)
강화학습
(Reinforcement Learning)
• 레이블된 데이터로 학습
• 미래 데이터 예측
• y = f(x)
• 레이블 없이 학습
• 데이터의 숨겨진 구조/특징 발견
• x ~ p(x), x = f(x)
• 보상 시스템으로 학습
• 의사결정을 위한 최적의 액션 선택
분류
(classification)
회귀
(regression)
클러스터링
(clustering)
• MNIST 필기체 인식
• 스팸 메일 분류
• 주가 예측
• 마케팅 고객 그룹화

Regression (Regression toward the mean)

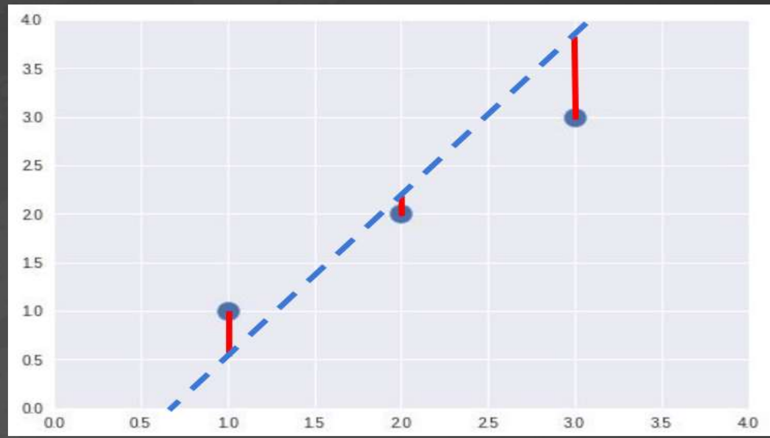
결과적으로, 전체적으로 봤을 때 이 데이터들은 전체 평균으로 되돌아 가려는 즉, 회귀하려는 속성이 있다는 통계적 원리를 설명하는 말

Linear Regression (선형 회귀)

- 요약하자면 데이터를 가장 잘 대변하는 직선의 방정식을 찾는 것이라고 요약이 가능하다.
- $y = ax + b$
- $H(x) = Wx + b$ (hypothesis: 가설, weight: 가중치, bias: 편향)



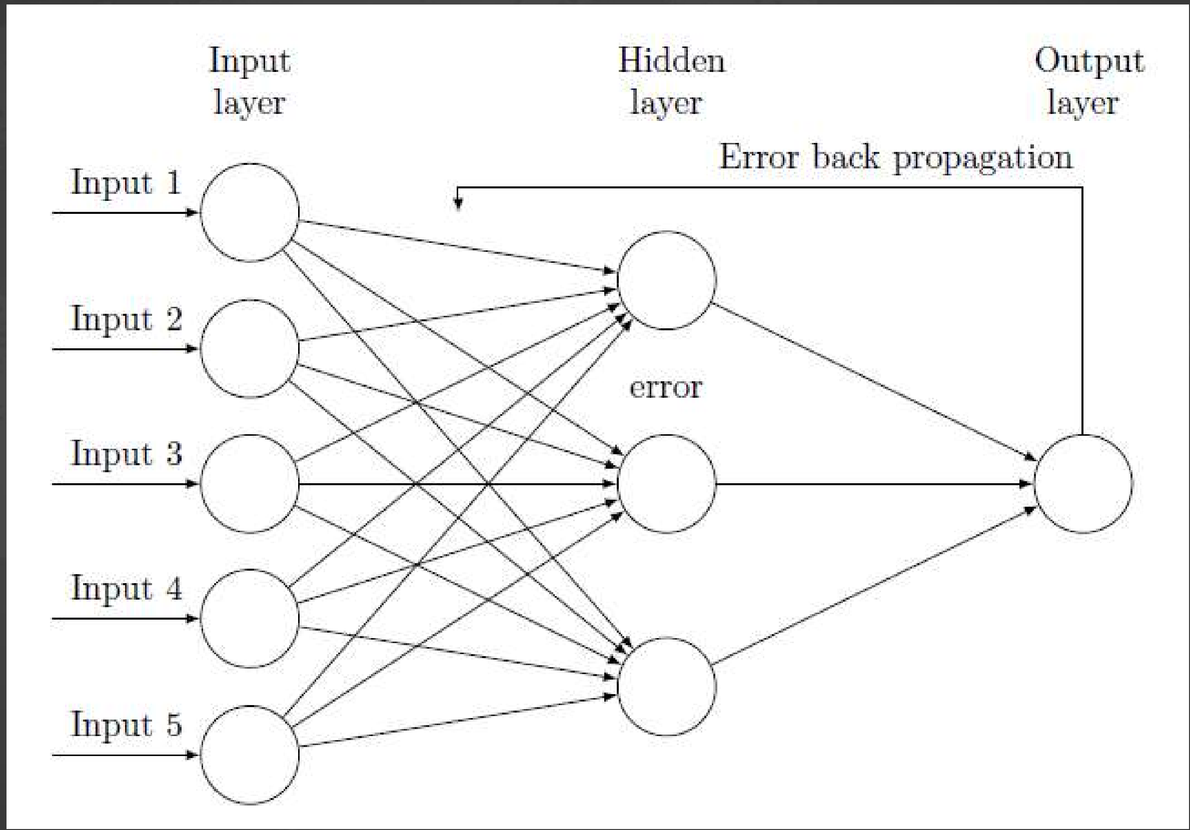
x	y
1	1
2	2
3	3



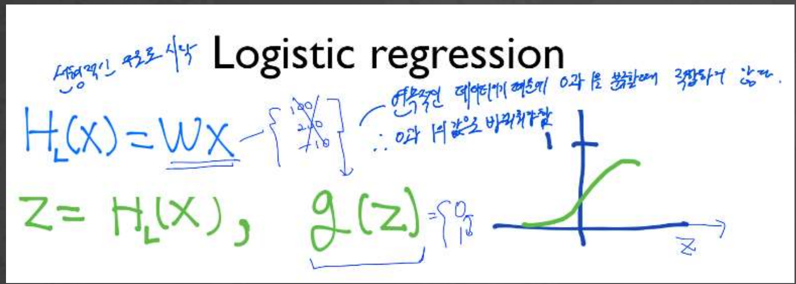
$$H(x) = Wx + b$$

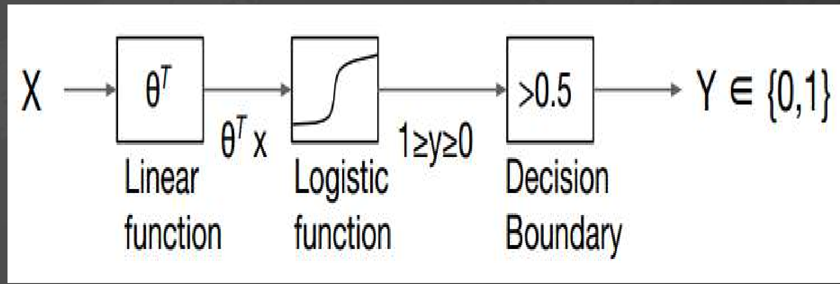
$$H(x) - y$$

오차에 제곱을 하는 이유는 h(x)의 값에서 y값을 뺀 값의 부호를 신경쓰지 않고 오차율만 구해야하기 때문에 제곱을 한다.


Input layer
Hidden layer
Output layer
Error back propagation
Input 1
Input 2
Input 3
Input 4
Input 5
error

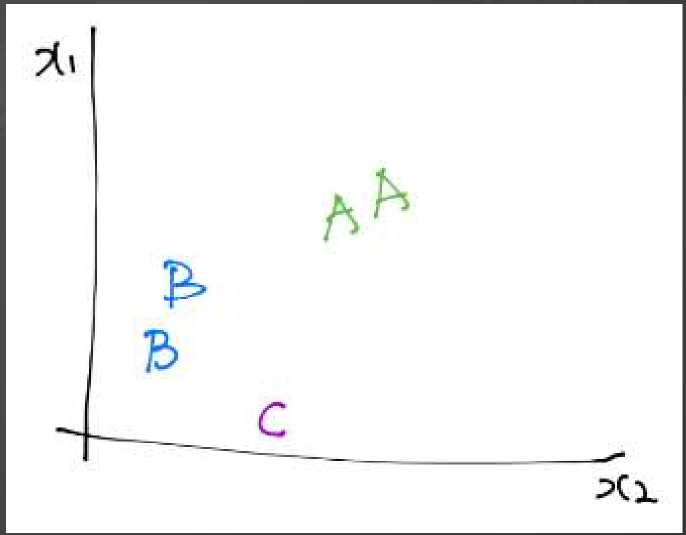
Logistic regression



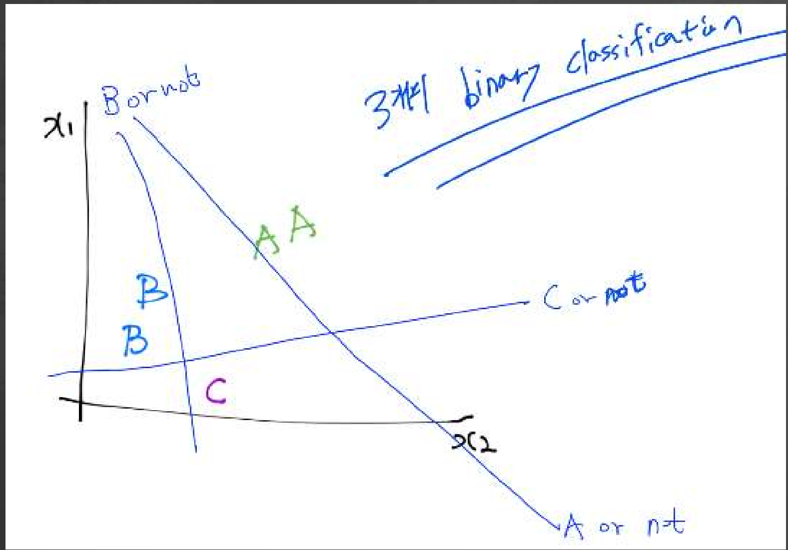


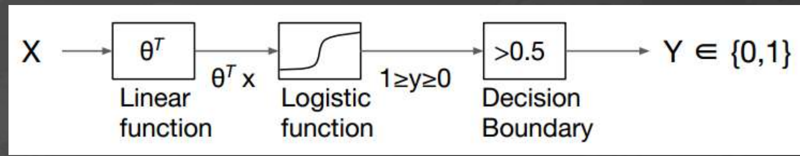
Multinomial classification

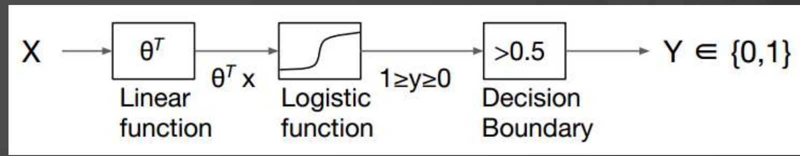
x1 (hours)	x2 (attendance)	y (grade)
10	5	A
9	5	A
3	2	B
2	4	B
11	1	C

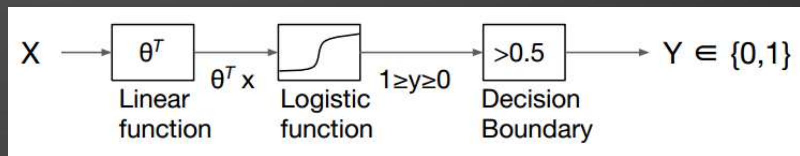


Multinomial classification









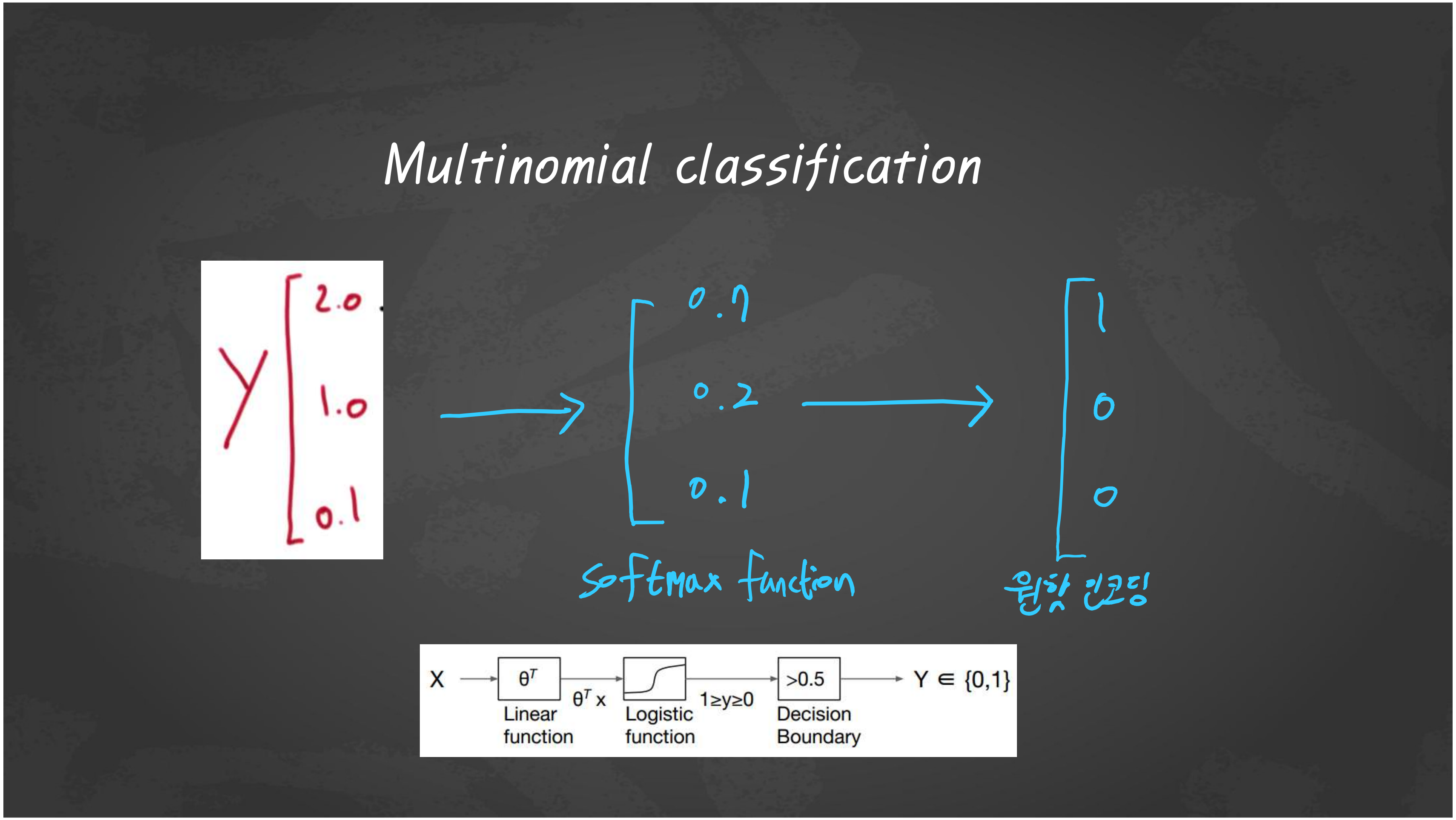
Multinomial classification

$$\begin{bmatrix} w_1 & w_2 & w_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} w_1x_1 + w_2x_2 + w_3x_3 \end{bmatrix}$$

$$\begin{bmatrix} w_1 & w_2 & w_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} w_1x_1 + w_2x_2 + w_3x_3 \end{bmatrix}$$

$$\begin{bmatrix} w_1 & w_2 & w_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} w_1x_1 + w_2x_2 + w_3x_3 \end{bmatrix}$$

$$\begin{bmatrix} w_{A1} & w_{A2} & w_{A3} \\ w_{B1} & w_{B2} & w_{B3} \\ w_{C1} & w_{C2} & w_{C3} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} w_{A1}x_1 + w_{A2}x_2 + w_{A3}x_3 \\ w_{B1}x_1 + w_{B2}x_2 + w_{B3}x_3 \\ w_{C1}x_1 + w_{C2}x_2 + w_{C3}x_3 \end{bmatrix}$$


Multinomial classification
Y
2.0
1.0
0.1
0.7
0.2
0.1
softmax function
1
0
0
원핫 인코딩
X
θ^T
θ^T x
Linear function
Logistic function
1≥y≥0
>0.5
Decision Boundary
Y ∈ {0,1}

Multinomial classification

$$-\sum_i L_i \log(S_i) = -\sum_i L_i \log(\bar{y}_i) = \sum_i (L_i) * (-\log(\bar{y}_i))$$

$0 \sim 1$

$$L = \begin{array}{c} A \\ B \end{array} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = A$$

$$\bar{Y} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = A \ (0), \quad \begin{bmatrix} 1 \\ 0 \end{bmatrix} \odot \begin{bmatrix} 0 \\ \infty \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Rightarrow 0$$

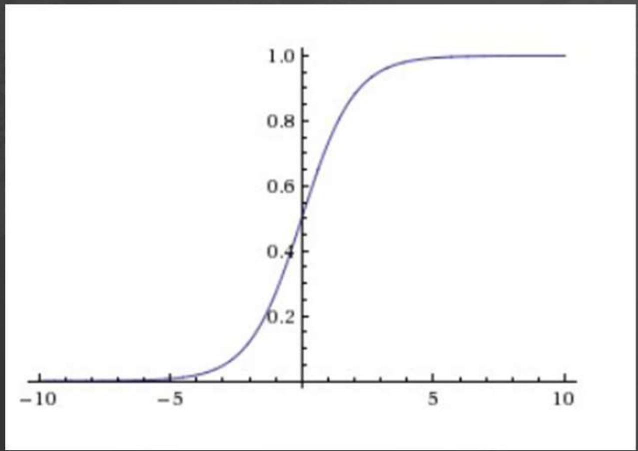
$$\bar{Y} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = B \ (\times), \quad \begin{bmatrix} 1 \\ 0 \end{bmatrix} \odot \begin{bmatrix} \infty \\ 0 \end{bmatrix} = \begin{bmatrix} \infty \\ 0 \end{bmatrix} \Rightarrow \infty$$

활성화 함수

시그모이드 함수 (Sigmoid Function)

Gradient Vanishing

sigmoid 함수는 0<n<1 사이의 값만 다루므로 결국 chain rule을 이용해 계속 값을 곱해나간다고 했을 때 결과 값이 0에 수렴할 수 밖에 없다는 한계를 가지고 있다.

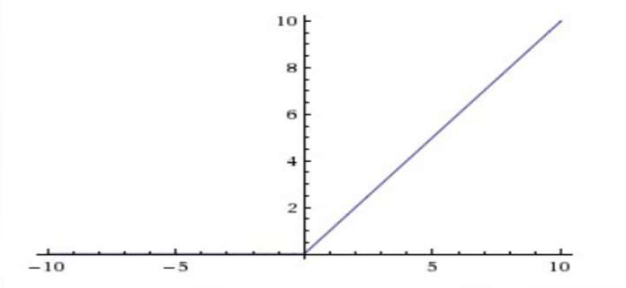


ReLU, Rectified Linear Unit

(1) 양 극단값이 포화되지 않는다. (양수 지역은 선형적)
(2) 계산이 매우 효율적이다 (최대값 연산 1개)
(3) 수렴속도가 시그모이드류 함수대비 6배 정도 빠르다.

입력값이 음수인 경우 항상 0을 출력함
-> 파라미터 업데이트가 안됨

$$f(x) = max(0, x).$$



비용 함수

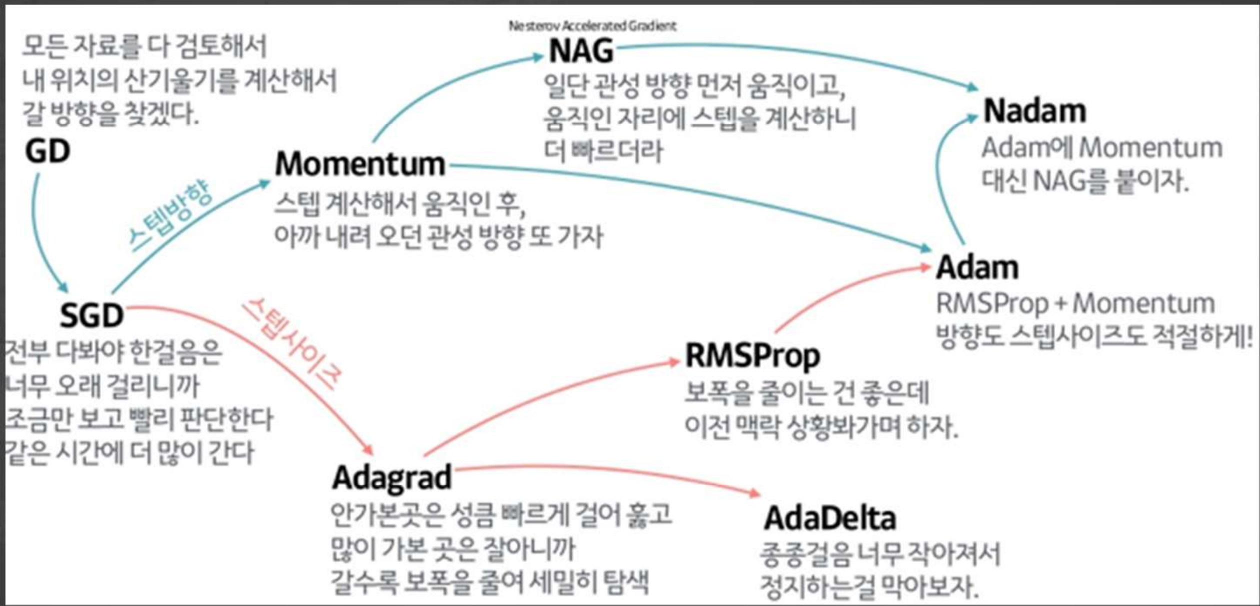
회귀

1. MSE
2. MAE
3. MSLE
4. MAPE
5. KLD
6. Poisson
7. Logcosh
8. Cosine Similarity
9. Huber

분류

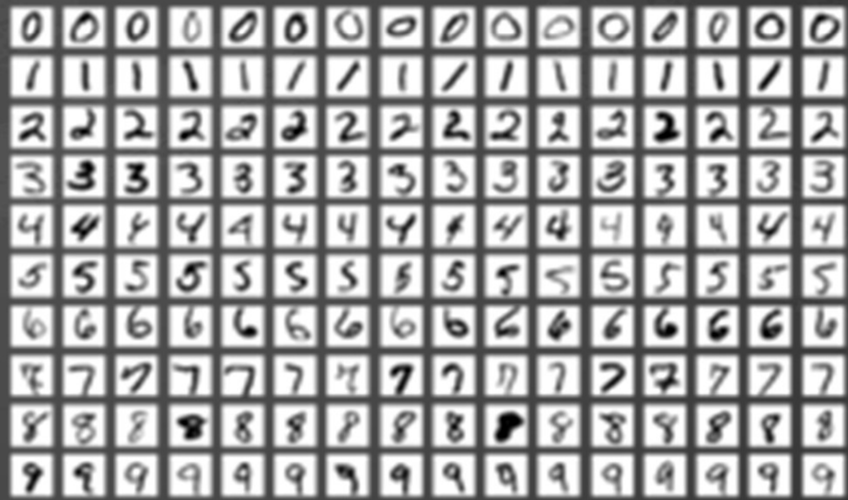
1. Binary cross-entropy
2. Categorical cross-entropy
3. Sparse categorical cross-entropy
4. Hinge
5. Squared Hinge
6. Categorical Hinge

옵티마이저

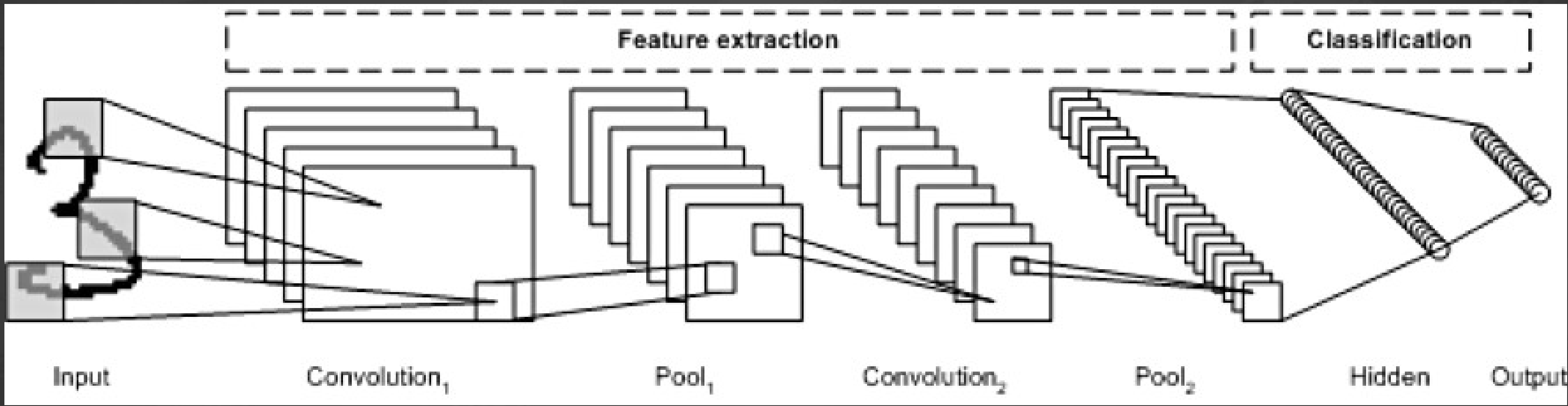

모든 자료를 다 검토해서
내 위치의 산기울기를 계산해서
갈 방향을 찾겠다.
GD
스텝방향
Momentum
스텝 계산해서 움직인 후,
아까 내려 오던 관성 방향 또 가자
Nesterov Accelerated Gradient
NAG
일단 관성 방향 먼저 움직이고,
움직인 자리에 스텝을 계산하니
더 빠르더라
Nadam
Adam에 Momentum
대신 NAG를 붙이자.
Adam
RMSProp + Momentum
방향도 스텝사이즈도 적절하게!
SGD
전부 다봐야 한걸음은
너무 오래 걸리니까
조금만 보고 빨리 판단한다
같은 시간에 더 많이 간다
스텝사이즈
RMSProp
보폭을 줄이는 건 좋은데
이전 맥락 상황봐가며 하자.
Adagrad
안가본곳은 성큼 빠르게 걸어 훑고
많이 가본 곳은 잘아니까
갈수록 보폭을 줄여 세밀히 탐색
AdaDelta
종종걸음 너무 작아져서
정지하는걸 막아보자.

MNIST

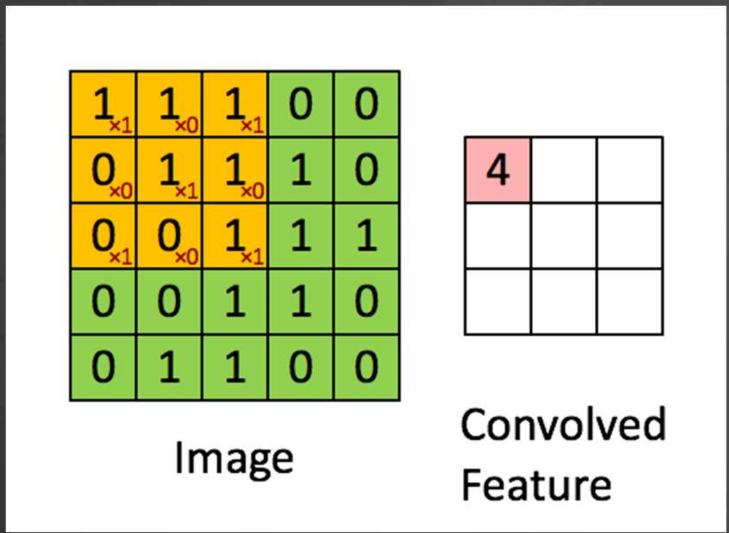
데이터베이스(Modified National Institute of Standards and Technology database)는 손으로 쓴 숫자들로 이루어진
대형 데이터베이스이며,
다양한 화상
처리 시스템을트레이닝하기 위해
일반적으로 사용된다.

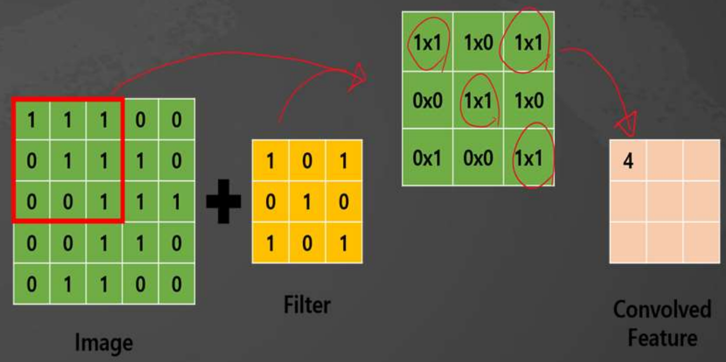


CNN(Convolutional Neural Network)

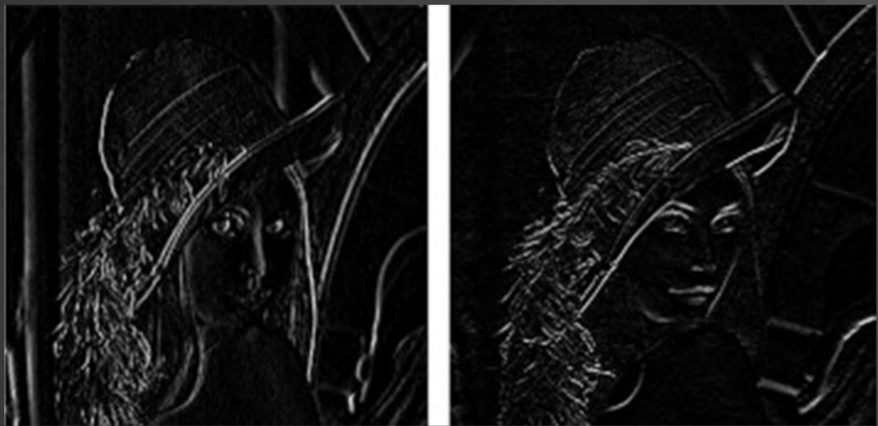


Convolution



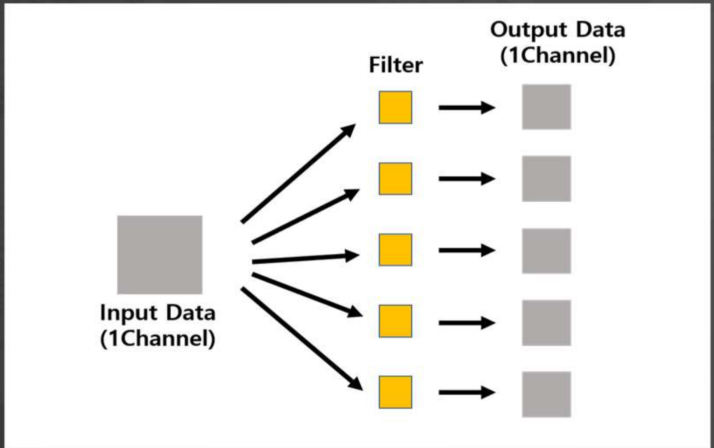


Filter(Kernel)

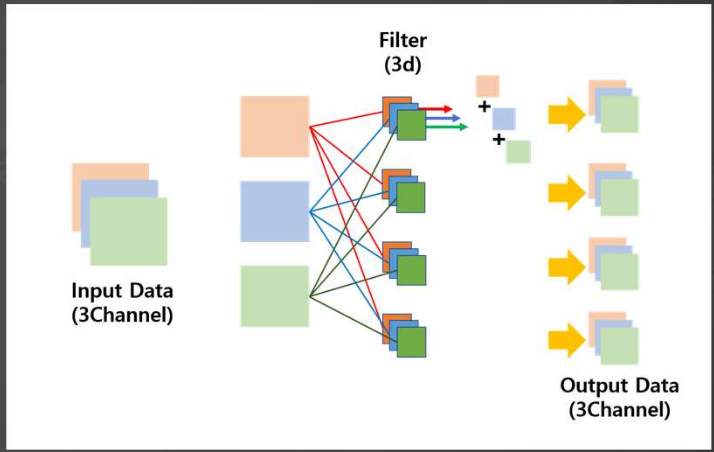


sobel filter

Channel



Filter
Output Data
(1Channel)
Input Data
(1Channel)



Filter
(3d)
Input Data
(3Channel)
Output Data
(3Channel)

Padding

0	0	0	0	0	0	0
0	1	1	1	0	0	0
0	0	1	1	1	0	0
0	0	0	1	1	1	0
0	0	0	1	1	0	0
0	0	1	1	0	0	0
0	0	0	0	0	0	0

Image
(+zero Padding)

- *valid*: padding 0을 뜻합니다. 즉 입력보다 출력의 크기가 작아집니다.
- *same*: padding이 존재하여 입력과 출력의 크기는 같습니다.

Pooling

13	20	30	0
8	12	3	0
34	70	33	5
111	80	10	23

Activation Map

20	30
111	33

Max Pooling

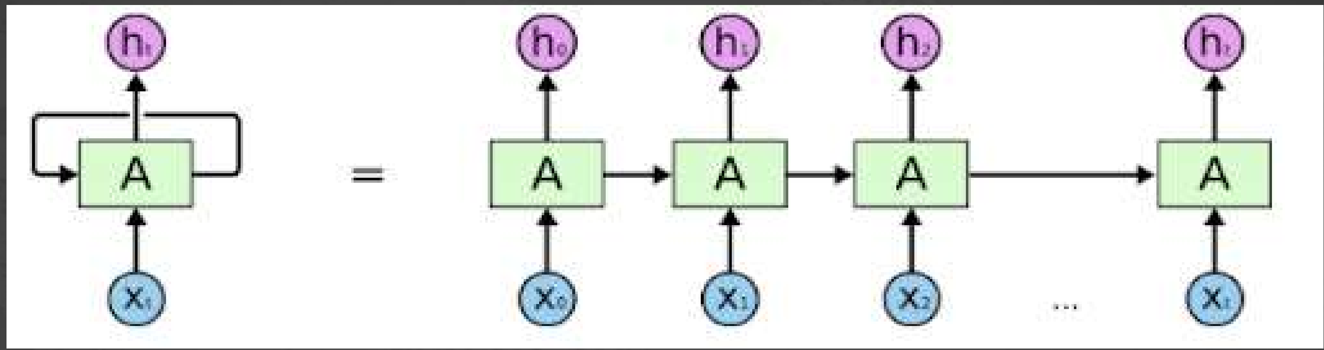
13	8
66	18

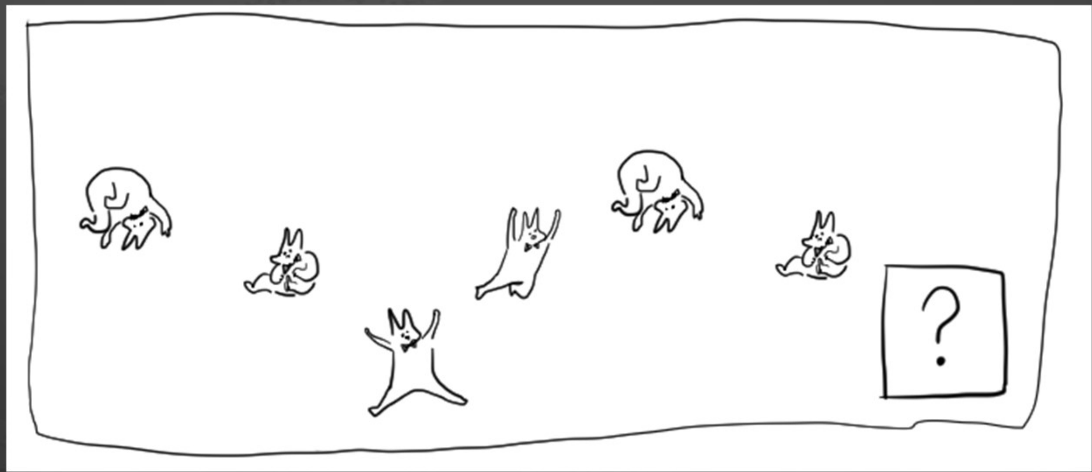
Average Pooling

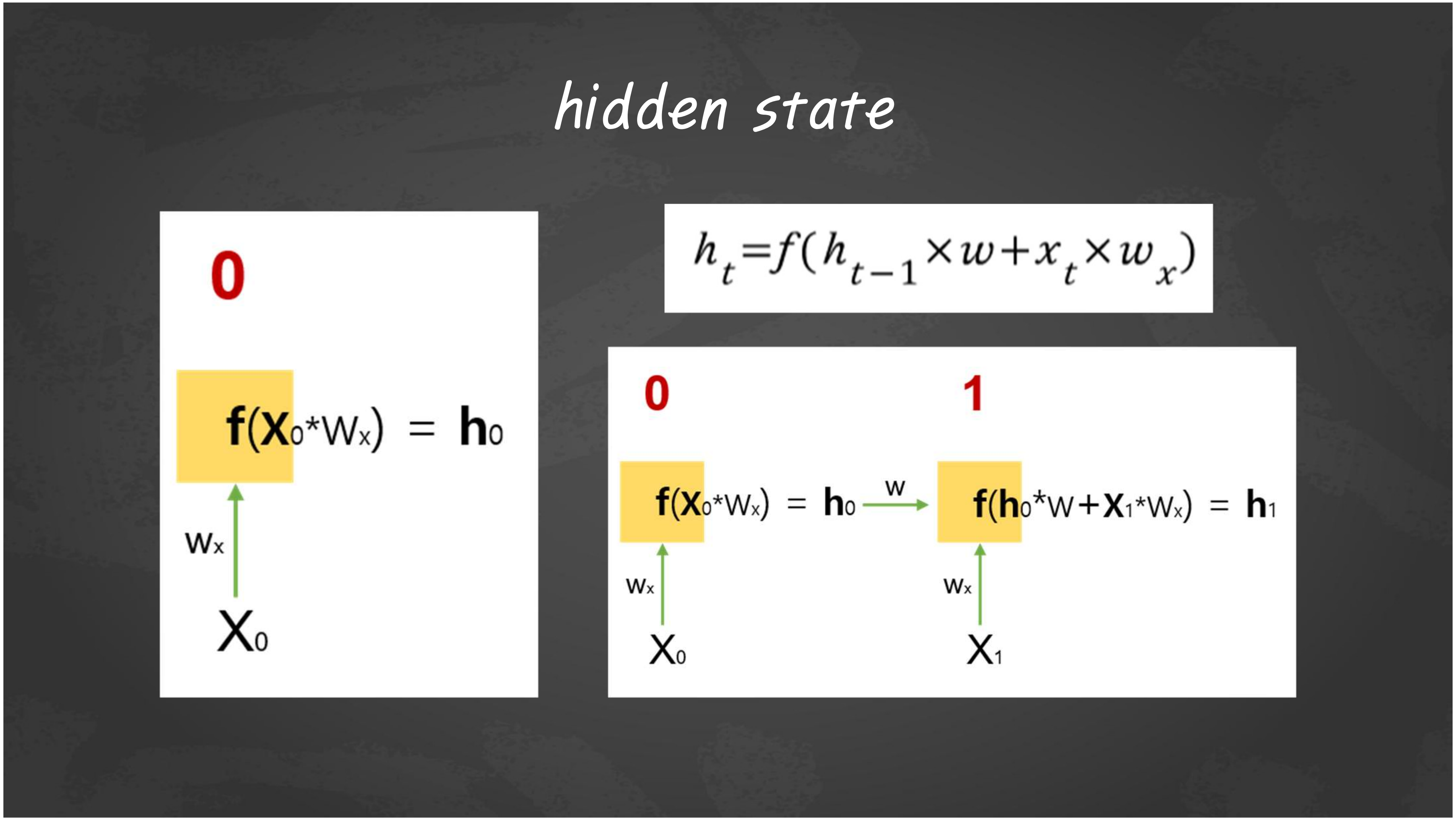
8	0
34	5

Min Pooling

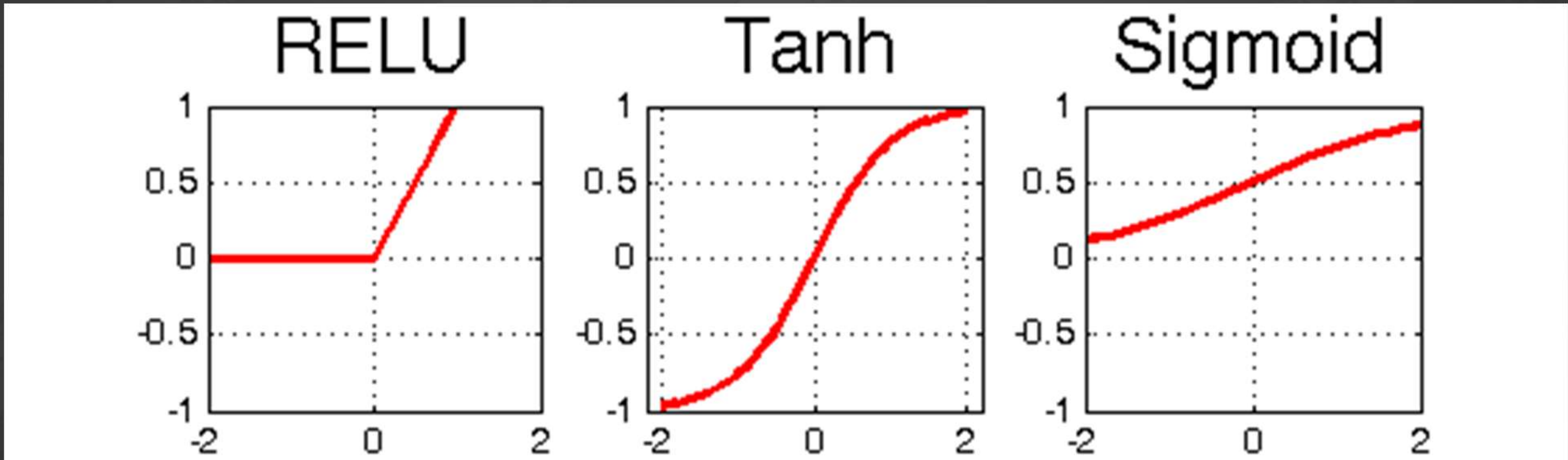
RNN(Recurrent Neural Network)






hidden state
$h_t = f(h_{t-1} \times w + x_t \times w_x)$
0
$f(X_0 * W_x) = h_0$
W_x
X_0
0
1
$f(X_0 * W_x) = h_0 \xrightarrow{W} f(h_0 * W + X_1 * W_x) = h_1$
W_x
W_x
X_0
X_1

활성화 함수



Output

$$y_t = f(h_t \times w_y)$$