

# 이미지 데이터 분석 (1일차)

# 강사 소개



- 유재명
- 서울대학교 산업공학과 / 동 대학원 인지과학 박사
- (주)퀀트랩 대표
- 국민대학교 겸임교수

# 강의 목표

- 머신러닝과 딥러닝의 기본 개념 이해
- 이미지 처리, 컴퓨터 비전의 방법론 습득
- Python 프로그래밍 언어를 통한 이미지 처리 실시
- 실습을 통한 개념 체득에 중점을 두고 진행

# 인공지능 Artificial Intelligence

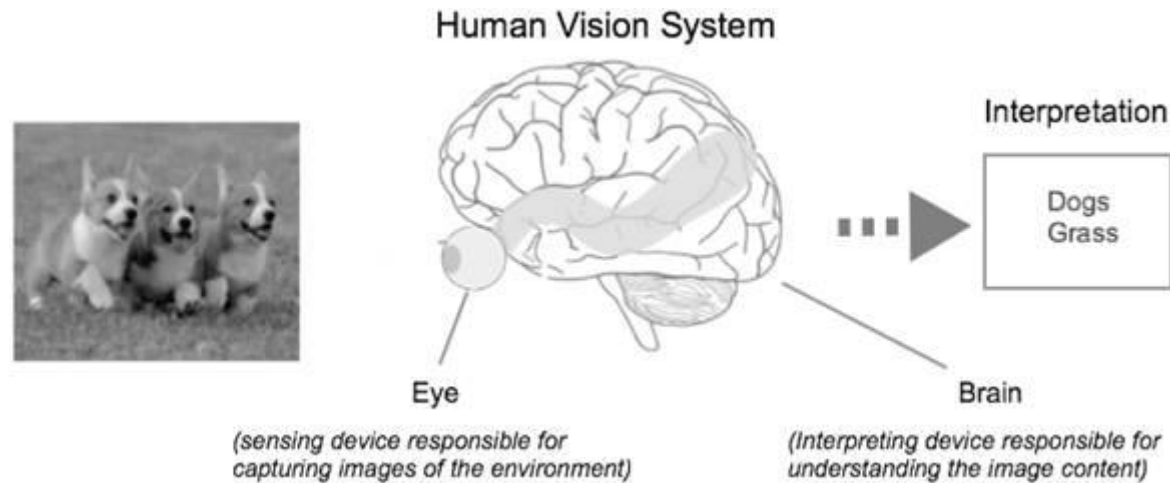
- 인간이나 동물의 지능을 컴퓨터나 시스템 등으로 만든 것
- 지능: 다양한 범위의 환경에서 목적을 달성할 수 있는 행위자의 능력
- 1956년 미국 다트머스大에서 열린 워크숍에서 John McCarthy가 명명

# AI 시스템의 핵심 개념

- 주변 환경에 대한 지각
- 그에 기반한 행동의 수행
- 컴퓨터 비전은 이미지를 이해하는 시지각(visual perception)에 관련

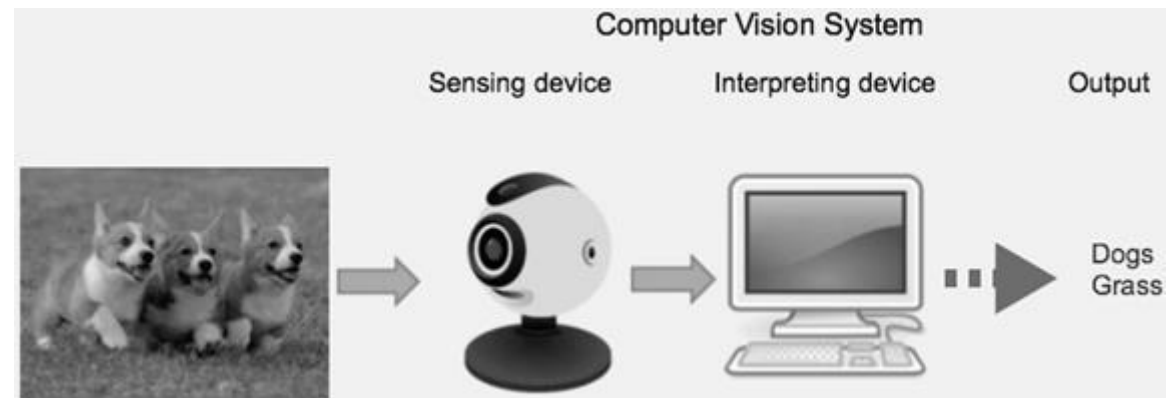
# 인간의 시각 시스템

- 이미지를 받아들이는 눈
- 이미지를 처리하고 해석하는 뇌

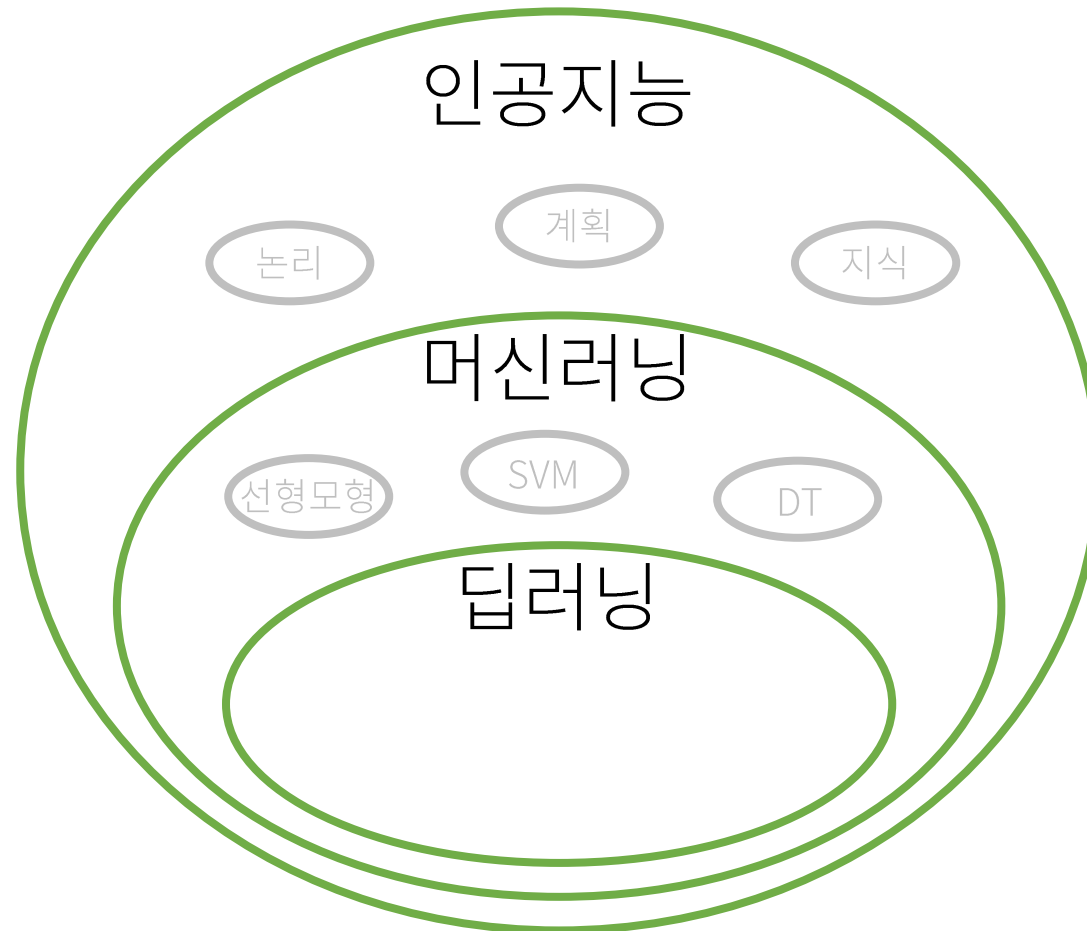


# AI 시각 시스템

- 카메라, 레이더, 엑스선, CT, 라이다 등으로 이미지 입력
- 인공신경망을 이용해 이미지 처리



# 인공지능, 머신러닝, 딥러닝의 관계





# 머신 러닝 Machine Learning

- 통계적/수학적 기법으로 모델을 설계
- 컴퓨터가 모델을 바탕으로 데이터로부터 학습

## 학습의 형태

- 지도학습
- 비지도학습
- 강화학습

## 모델

- 선형 모델
- 의사결정나무
- 딥러닝
- ...

# 머신러닝에서 학습의 형태

- 지도 학습:  $x$ 와  $y$ 가 짝지어진 형태의 데이터에서 관계를 학습
- 비지도 학습: 데이터의 내재적 구조를 학습
- 강화 학습: 환경과 상호작용하여 가장 많은 보상을 받는 행동을 학습

# 지도 학습 Supervised Learning

- 입력( $x$ )과 출력( $y$ )이 짝지어진 데이터셋이 있을 때
- $x$ 로부터  $y$ 를 예측하는 함수를 학습
- 아직  $y$ 가 관찰되지 않은  $x$ 에 적용하여  $y$ 를 예측
- 머신 러닝의 90% 이상을 차지하는 학습 형태

# 지도 학습의 예시

- 주택의 정보( $x$ )  $\rightarrow$  해당 주택의 시장가격( $y$ )
- 개인의 금융 생활 정보( $x$ )  $\rightarrow$  대출 상환 가능성( $y$ )
- 지원자의 자기소개서( $x$ )  $\rightarrow$  입사 후 인사평가 점수( $y$ )

# 지도학습으로 어려운 것

- 지도학습이 되려면 하나의 사례마다  $x$ 와  $y$ 가 있어야 함
- 예시) 수많은 인터넷 뉴스 → 금융 산업 트렌드
- $y$ 가 데이터로 관찰 가능해야 함
- 예시) 블로그 맛집 리뷰 → 진위여부

# 인간과 기계의 학습 방법의 차이

- 스스로 생각하는 존재, 인격화된 존재라는 통념
- 인간은 소량의 데이터나 심지어 데이터 없이도 학습이 가능하나 인공지능은 대량의 데이터가 필요 (cf. Zero-Shot Learning)
- 인공지능은 좁은 범위의 문제에만 학습이 가능, 전이(transfer)가 어려움
- 문제가 달라지면 처음부터 새롭게 접근해야

# 컴퓨터 비전의 응용

- 이미지 분류
- 대상 탐지
- 자동 이미지 캡셔닝
- 이미지 생성

# 이미지 분류 Image classification

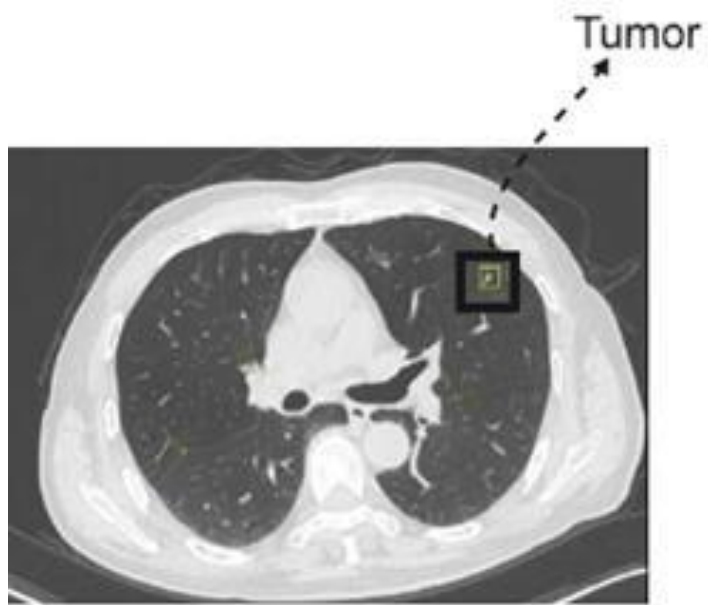


Score ▾	Label ▾
<div><div></div>51.8%</div>	Labrador retriever
<div><div></div>7.3%</div>	Chesapeake Bay retriever
<div><div></div>5.2%</div>	golden retriever
<div><div></div>3.6%</div>	dingo, warrigal, warragal, Canis dingo
<div><div></div>2.9%</div>	bloodhound, sleuthhound
<div><div></div>2.7%</div>	tennis ball

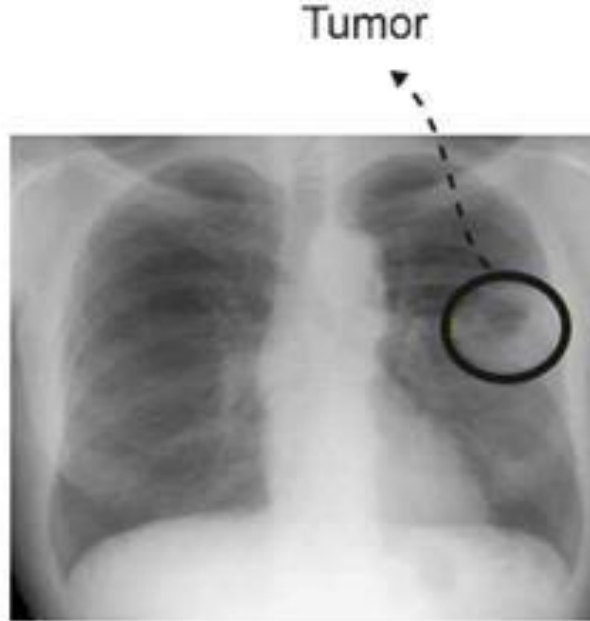


# 암 진단

- 초기의 종양의 경우 식별하기 어려운 경우가 있음
- 딥러닝을 이용해 조기에 종양을 식별



CT Scan



X-Ray

# Object detection and localization



# Semantic Segmentation



# Pose Estimation

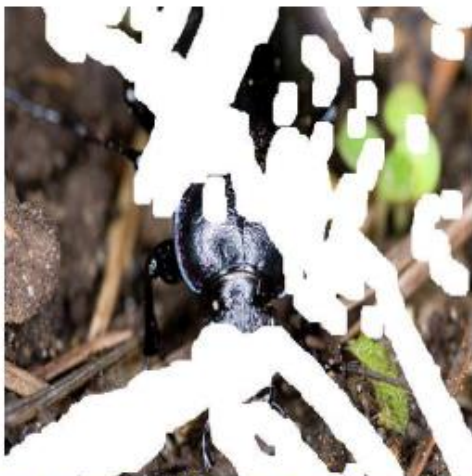




# Image Colorization



# Image Reconstruction





# Image Super-Resolution

bicubic  
(21.59dB/0.6423)



SRResNet  
(23.53dB/0.7832)



SRGAN  
(21.15dB/0.6868)



original



# Automatic image captioning



**A person riding a motorcycle on a dirt road.**



**A group of young people playing a game of frisbee.**



# Style Transfer





# GAN으로 생성한 이미지



<https://generated.photos/faces>

# 활용 사례

- 보험에서 사고차량의 사진으로 자동차 종류, 손상도 등을 판독. 예상수리비 산출, 보험 가액 판단
- 스마트 카메라를 활용한 무인형 매장. 고객들이 계산하고 싶은 품목의 이미지를 인식하고 분류
- 물류에서 패키징 검수 자동화. 제품의 개수나 종류가 완벽하게 패키지에 포함되었는지 확인
- 출입게이트나 위험구역 등에서 카메라로 출입자나 위험요소를 인지
- 비대면 상태에서 상황을 파악: 기차길 근처에 사람, 동물, 사물 등의 접근을 감시

# 인공지능 vs. 인간

- 130가지 견종을 분류하기
- 충분한 시간을 가지고 학습하면 인간이 우월
- AI는 이미지 몇 만장을 몇 분만에 학습하여 95% 정확도로 분류 가능

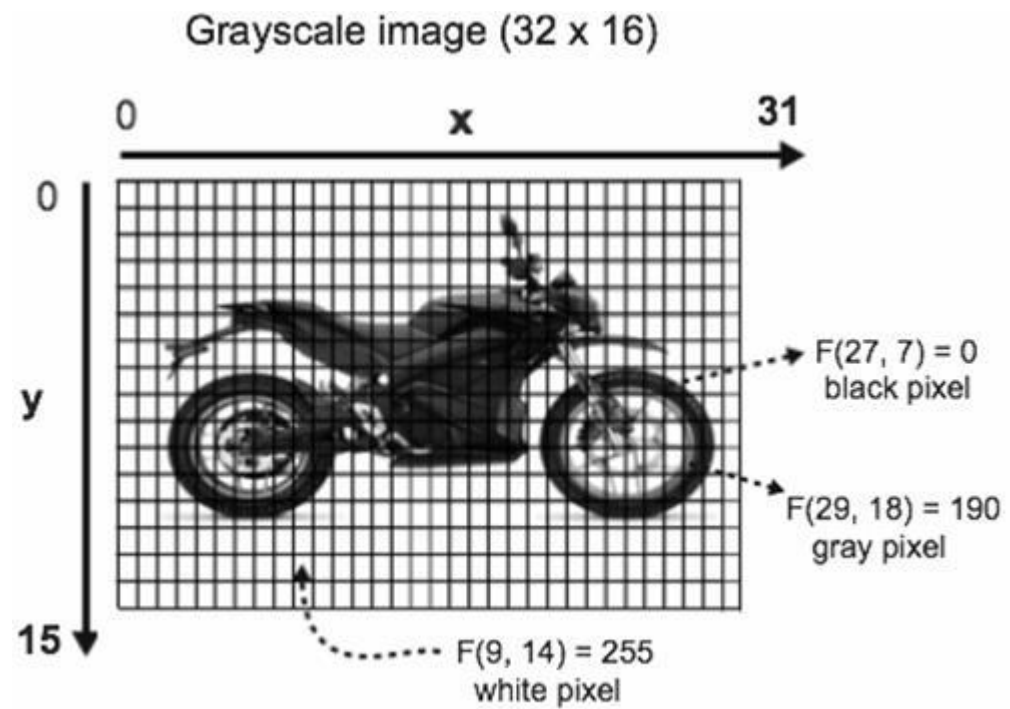
# 컴퓨터 비전 파이프라인

- 데이터 입력
- 전처리
- 특징 추출
- 기계학습 모형화

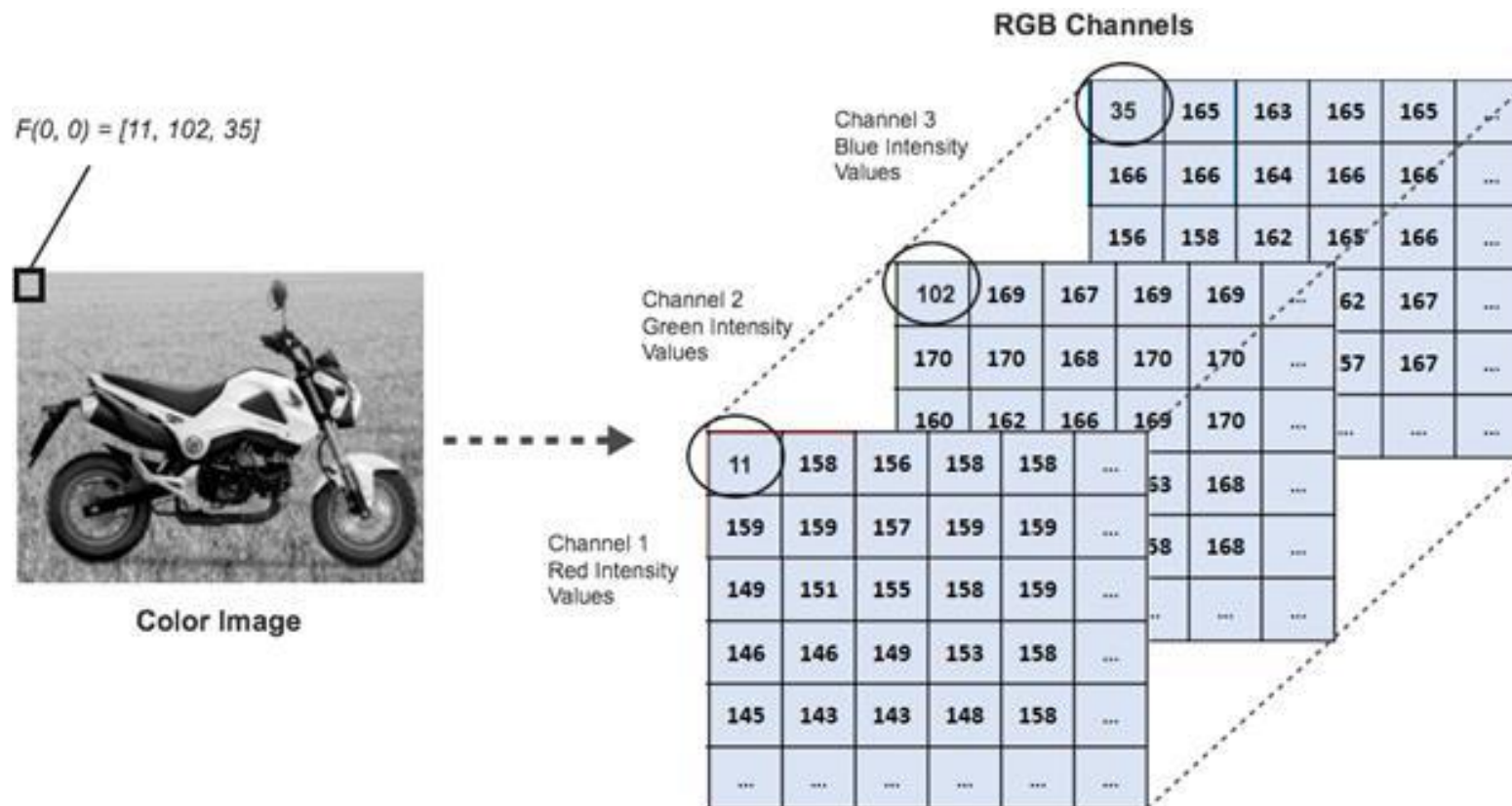
# 컴퓨터 그래픽의 2가지 방식

- 비트맵 방식: 이미지를 점으로 표현. 하나의 점이 픽셀(pixel)
- 벡터 방식: 이미지를 선으로 표현
- 일반적으로 컴퓨터 비전에서는 비트맵 방식을 사용

# 비트맵



# 컬러 이미지

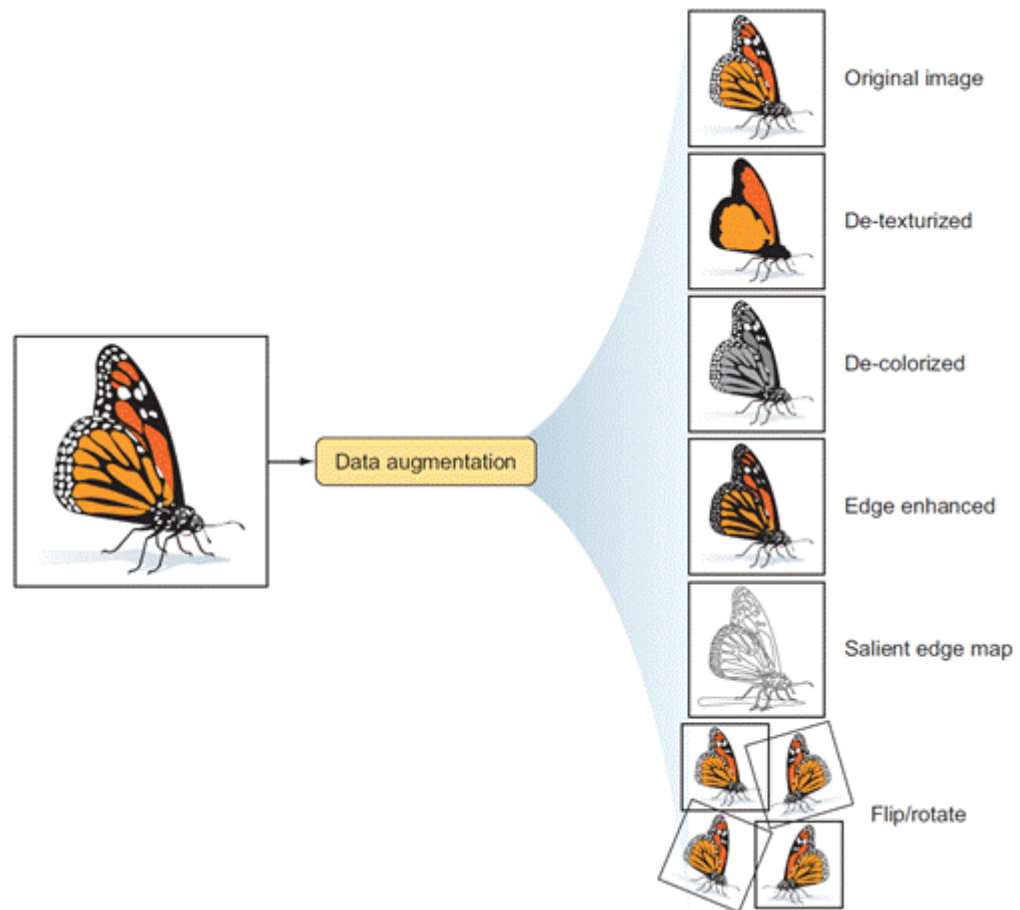




# 흑백 vs. 컬러

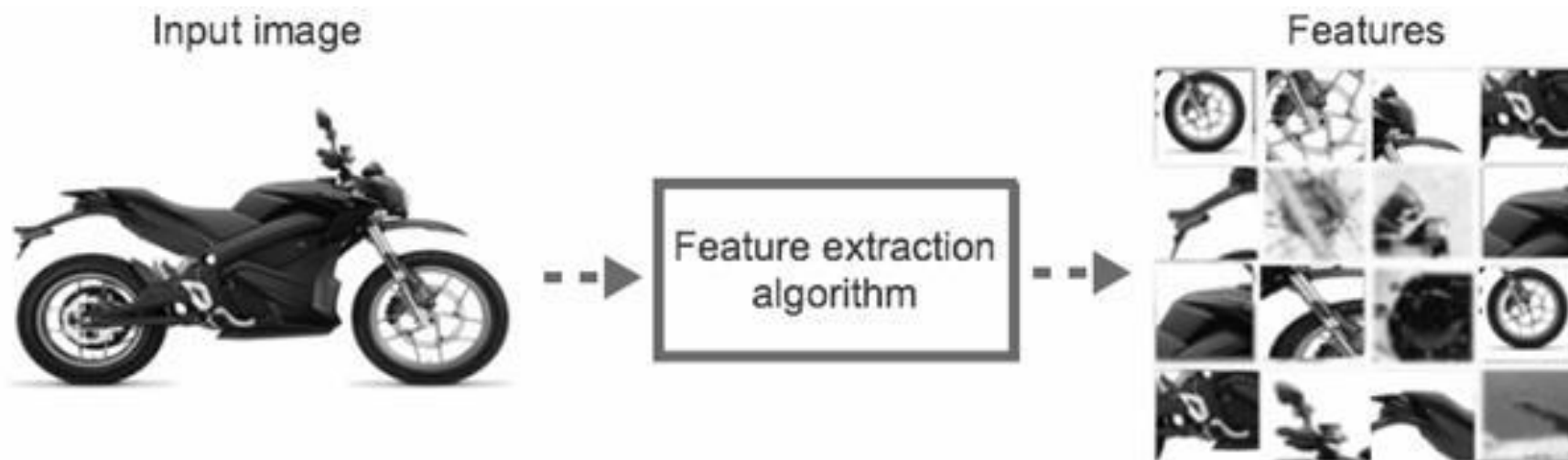
- 컬러 이미지는 RGB 3가지 색상의 이미지로 구성
- 흑백 이미지는 데이터의 크기가 작아지므로 용량이 줄어듦고 계산 속도가 빨라짐
- 색상이 중요하지 않은 과제의 경우 컬러 이미지보다 흑백 이미지를 사용하는 것이 계산 속도와 성능을 높일 수 있음

# 데이터 증강

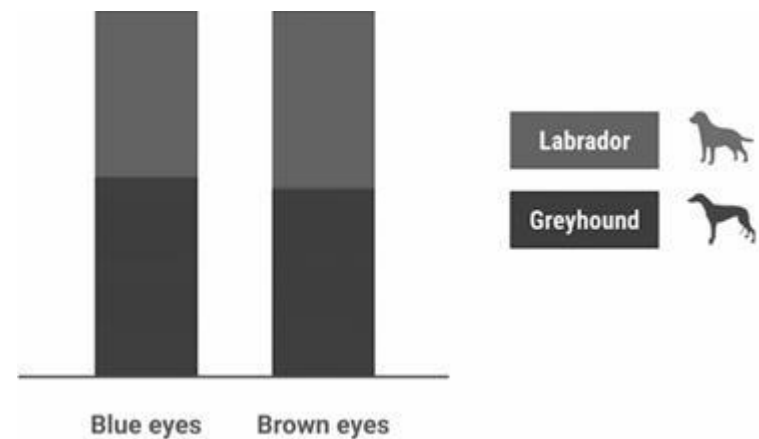
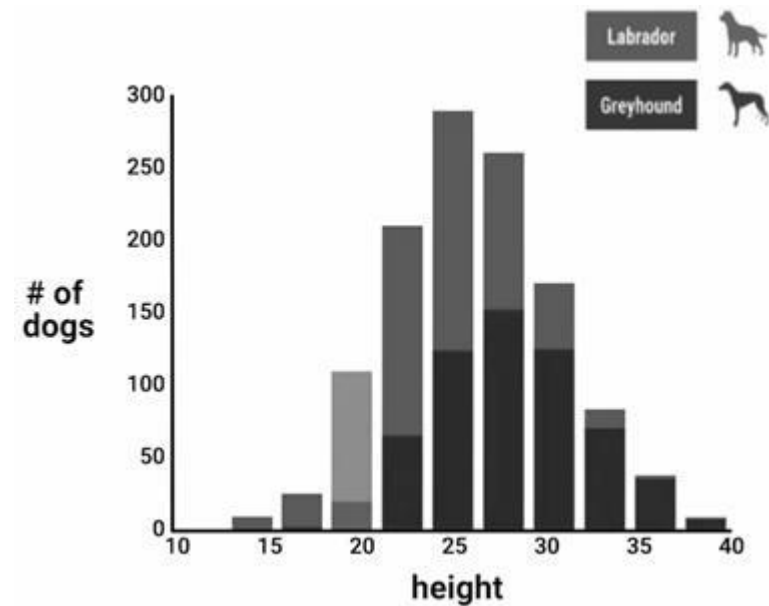


# 특징 feature

- 이미지의 한 부분 또는 패턴
- 이미지에 포함된 사물이 가진 고유함을 반영



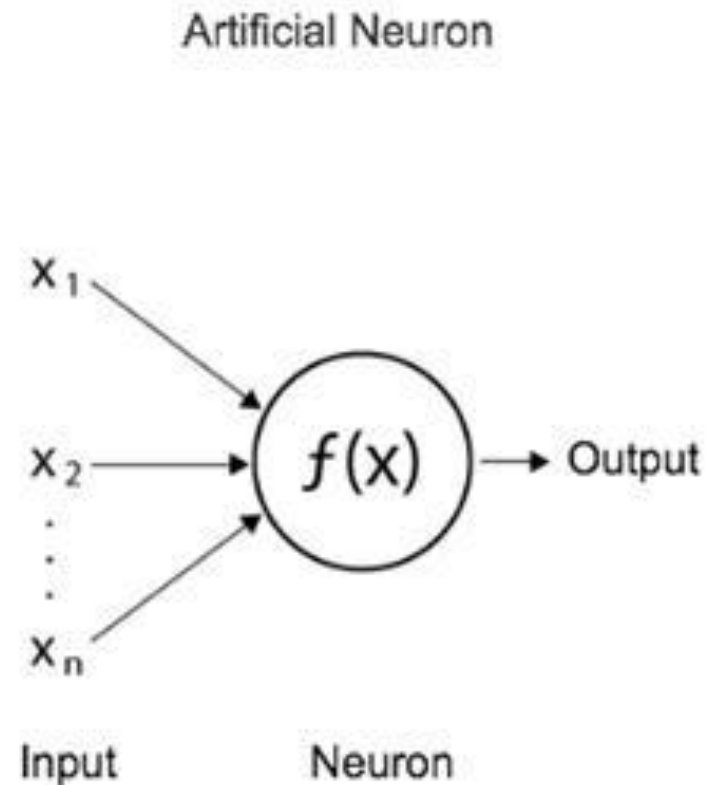
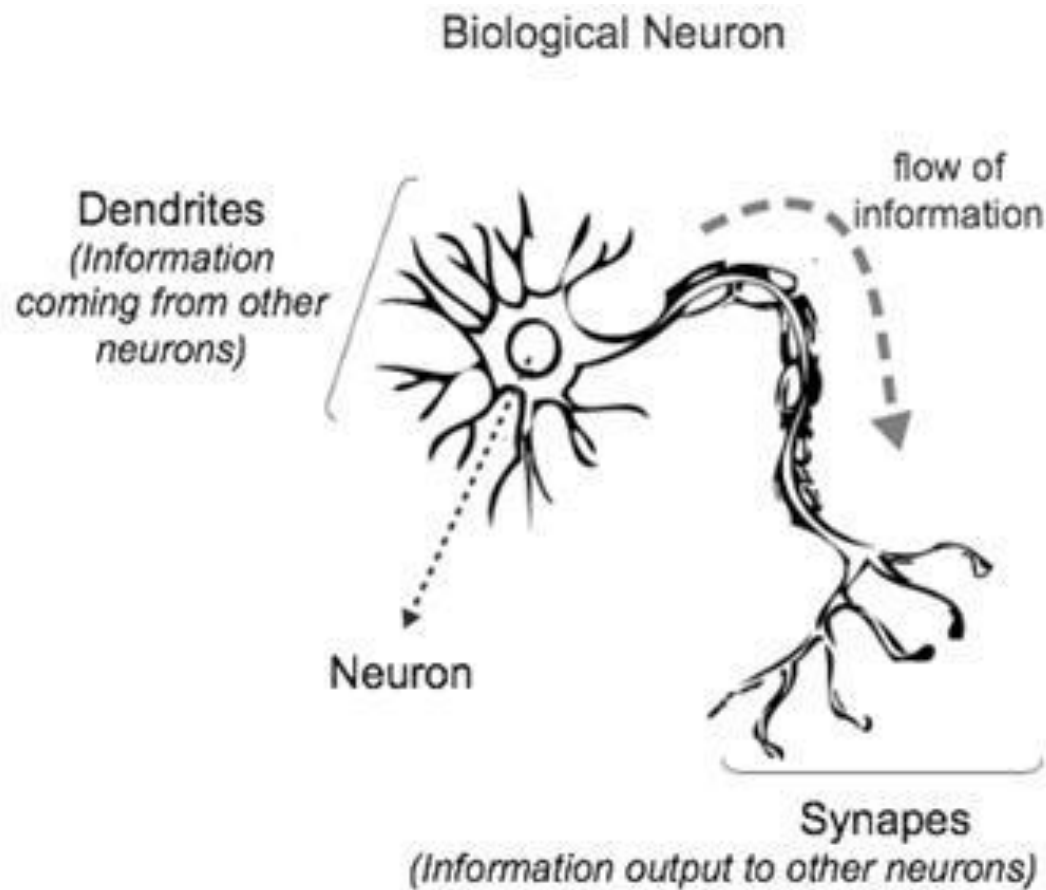
# 유용한 특징



# 유용한 특징의 성질들

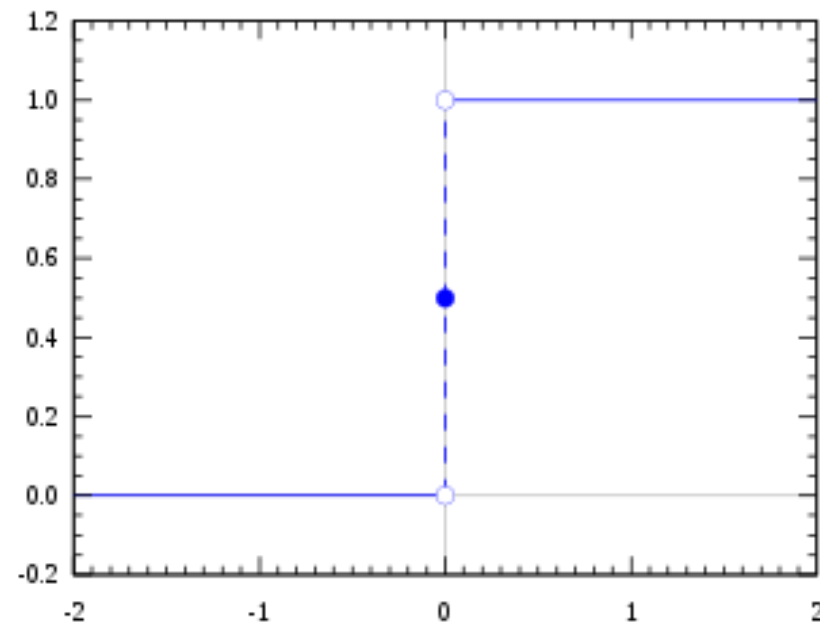
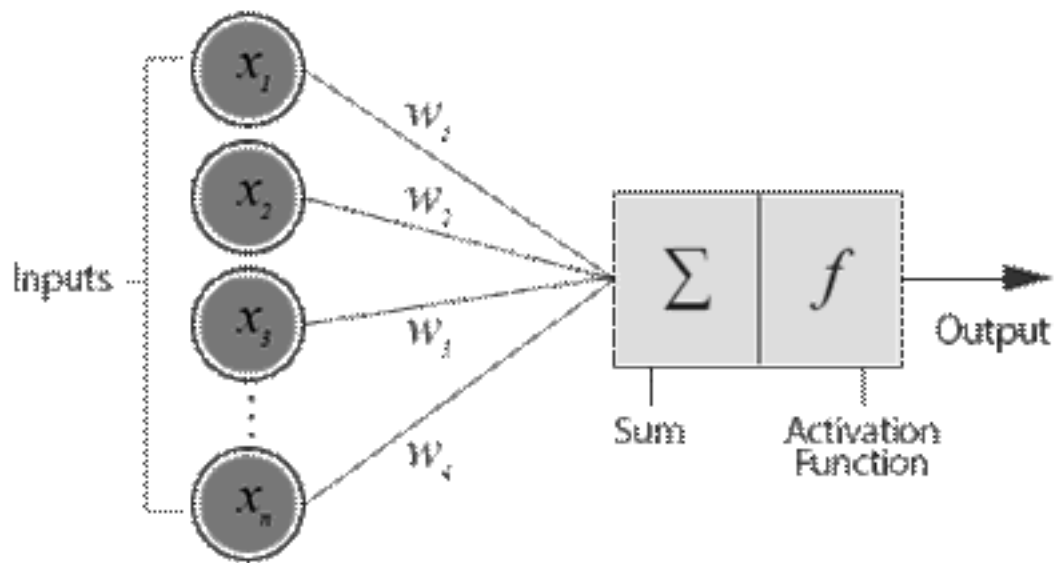
- 식별가능
- 비교가 쉬움
- 다양한 조건에서 일관적
- 잡음이 많은 이미지에서도 보임
- 사물의 특정한 부분에서도 보임

# 생물학적 뉴론과 인공 뉴론

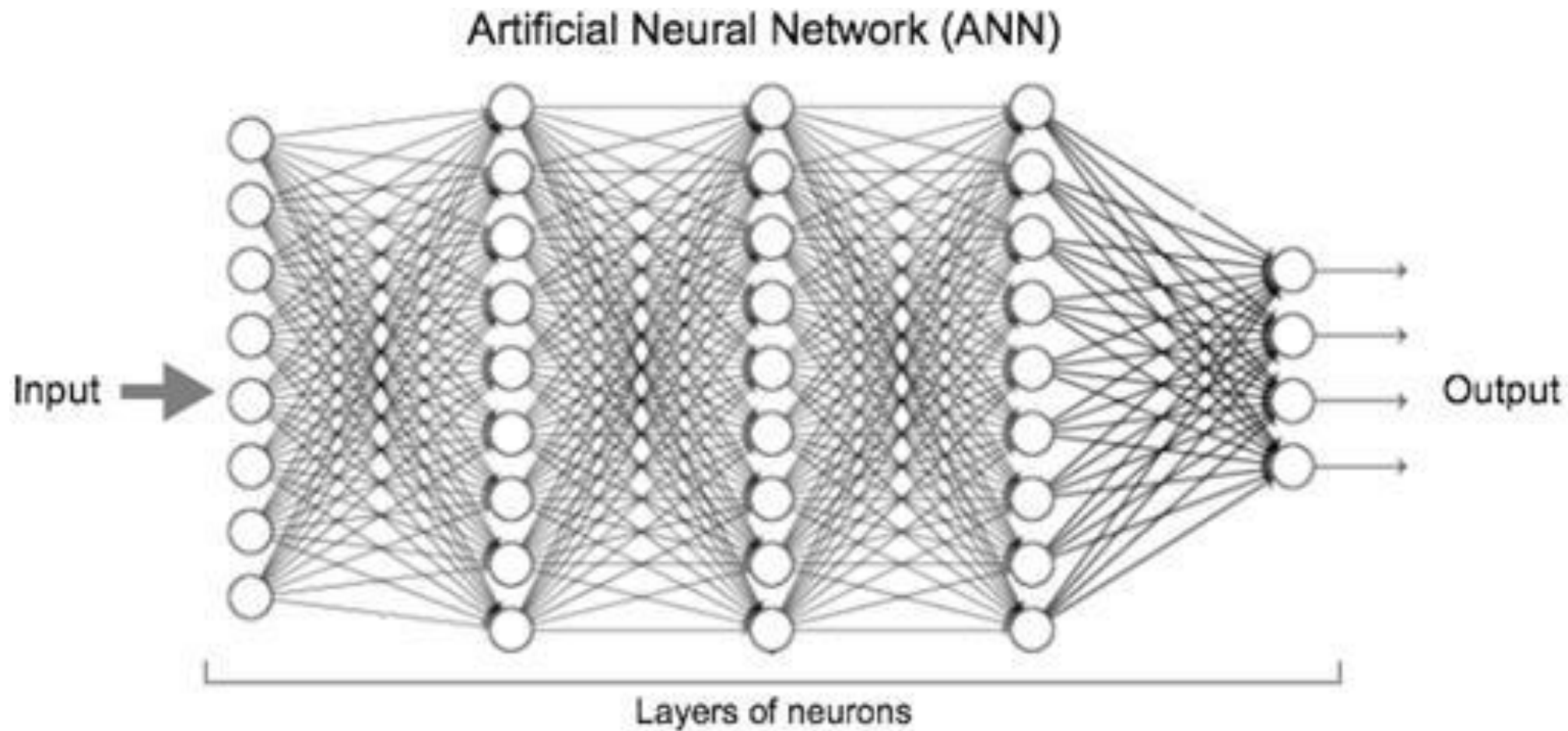


# 퍼셉트론의 작동 방식

- 각각의 입력을 가중합 weighted sum
- 역치 threshold 를 넘기면 1, 그렇지 않으면 0인 계단 함수 step function을 적용

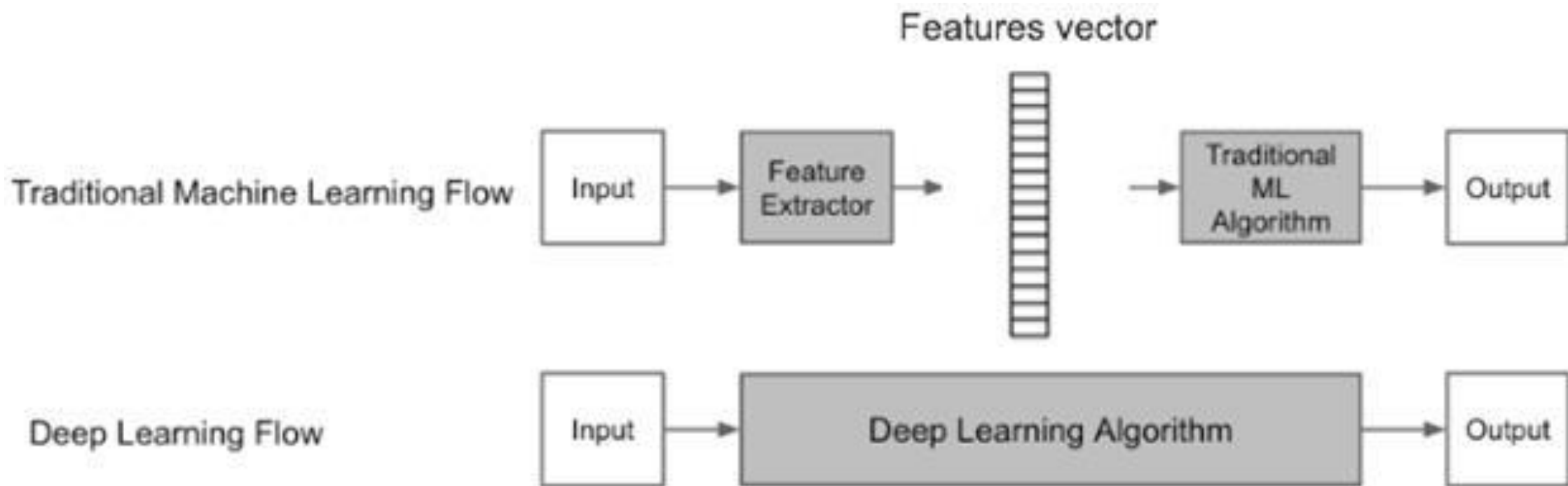


# 인공신경망





# 전통적 머신러닝 vs. 딥러닝



# 딥러닝 이전과 이후의 특징 추출

- 딥러닝 이전에는 과제 특성에 맞게 다양한 특징 추출 알고리즘들이 개발
- 딥러닝 이후에는 신경망에 이미지를 집어넣고 학습시키면, 신경망이 과제를 수행하는데 적합한 이미지의 특징들을 추출

# 딥러닝에서 특징 추출의 원리

- 인공신경망은 입력들의 가중합으로 구성
- 예측에 도움이 되는 입력에는 높은 가중치, 그렇지 않은 입력에는 낮은 가중치
- 이러한 처리 과정이 여러 레이어로 이뤄짐
- 각 레이어가 과제 수행에 필요한 특징들을 추출하게 됨

# 독립변수와 종속변수

종속변수 dependent variable

- 예측의 대상
- 인과 관계에서 결과
- 출력
- $y$

독립변수 independent variable

- 예측에 바탕이 되는 정보
- 인과 관계에서 원인
- 입력
- $x$

# 변수의 종류

- 연속변수
- 범주형 변수
- 변수의 종류에 따라 가능한 연산이 달라짐

# 연속변수 continuous variable

연속

- 연속변수: 값이 연속적으로 변하는 변수
- 예: 크기, 무게, 길이 등
- 연속적이지 않더라도 등간격으로 변하는 경우도 포함(예: 점수, 개수 등)
- 서로 더하거나 일정한 수로 나누는 등의 연산이 가능 (예: 평균)

# 범주형 변수 categorical variable

범주형

- 서로 다른 이름이나 유형을 값으로 가지는 변수 (예: 성별, 여부, 지역)
- 수나 번호로 표시해도 구별을 위한 이름일 뿐 크기나 순서를 뜻하지는 않음
- 예: 남자는 1, 여자는 2로 표기한다고 해서 남자 + 여자 = 3은 불가
- 더하거나 나누는 연산을 할 수 없고 세는(counting) 것만 가능

# 회귀분석 regression

- 여러 가지 의미로 사용
- 가장 넓은 의미:  $X \rightarrow Y$  예측
- 중간 의미: Y가 연속인 경우(Y가 범주형인 경우는 분류)
- 좁은 의미: 선형 회귀 분석(선형 모델을 이용한 회귀분석)



# 선형 모형 linear models

- 직선 형태의 모형

$$\hat{y} = wx + b$$

- $\hat{y}$  : 종속변수에 대한 예측치
- $x$  : 독립변수

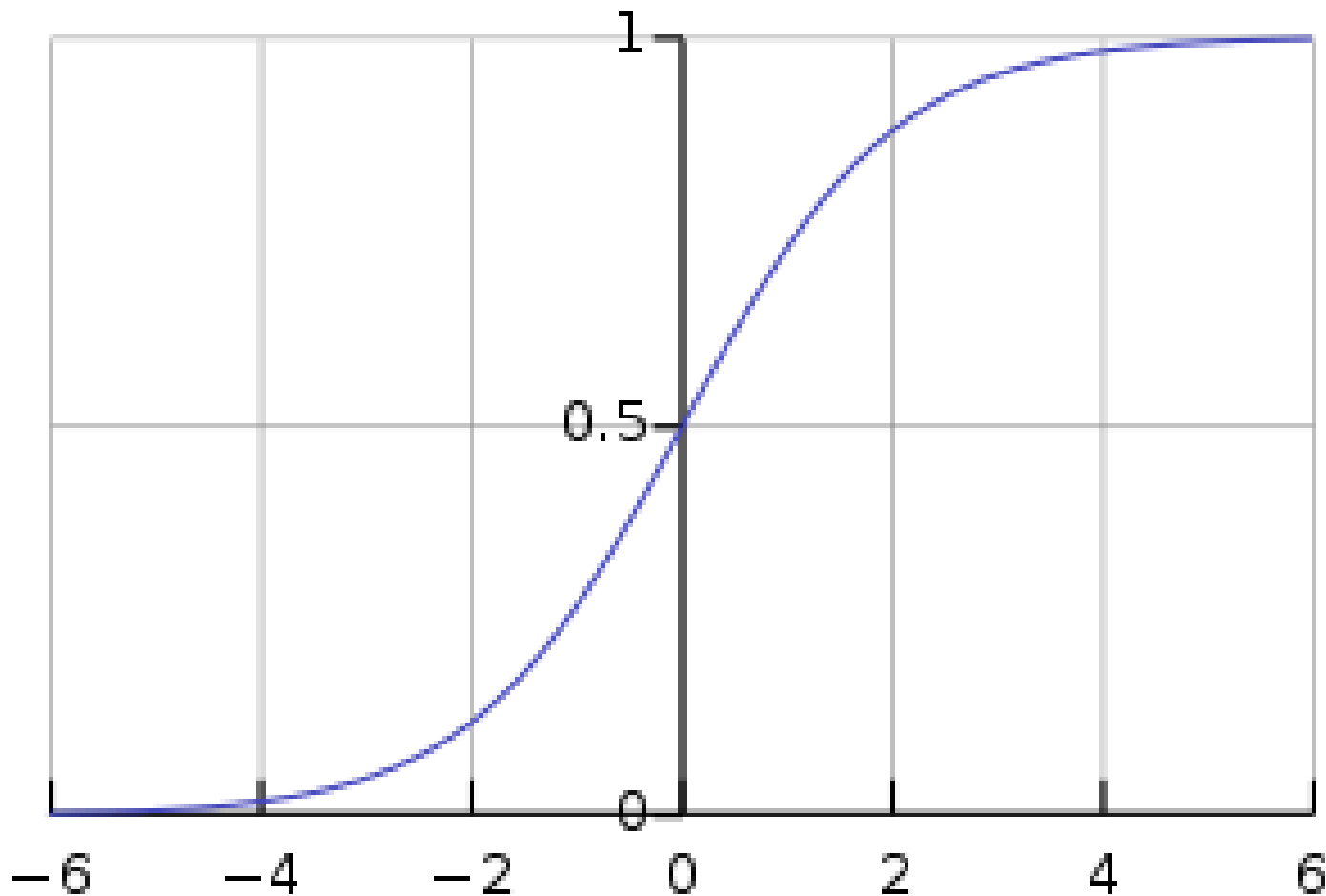
파라미터 parameter

- $w$  : 기울기 slope 독립변수가 1 증가할 때 종속변수의 변화
- $b$  : 절편 intercept 독립변수가 모두 0일 때 종속변수의 값

# 분류분석 classification

- Y가 범주형인 경우
- 로지스틱 회귀분석: 선형 모형 + 로지스틱 함수

# 로지스틱 함수 logistic function



$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$

0~1 사이의 출력을 갖는 함수

# 로지스틱 회귀분석 logistic regression

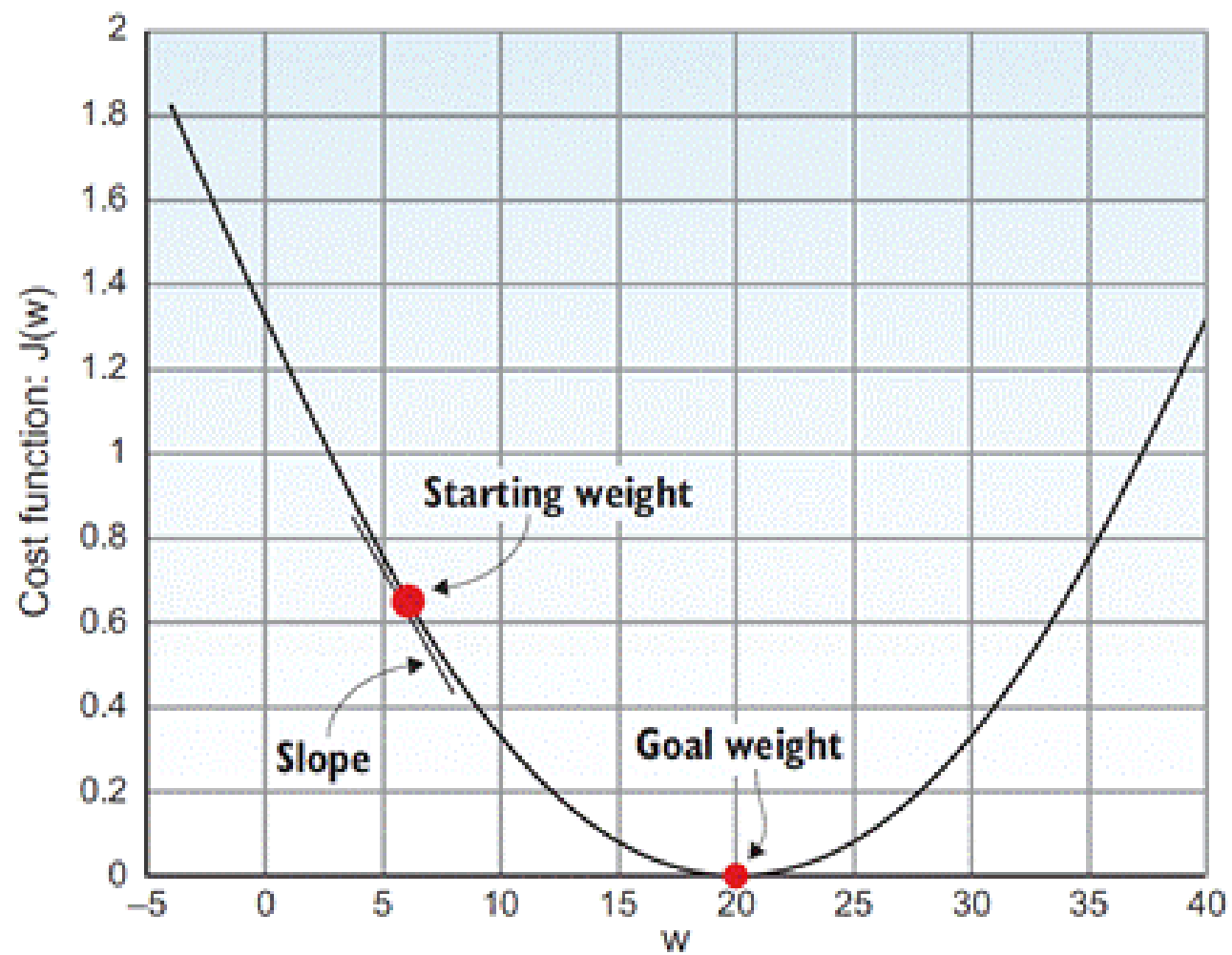
$$\hat{y} = \sigma(wx + b)$$

- $\sigma$ 에 로지스틱 함수를 적용
- 확률은 0~1 사이의 값이므로 확률을 출력하는 함수를 만들 수 있다

# 경사하강법 gradient descent

- 기존의 파라미터  $w$ 를 바탕으로 예측  $\hat{y}$ 을 출력
- 실제  $y$ 와 비교
- 손실 loss을 계산  $L(y, \hat{y})$
- 손실을 줄이는 방향으로  $w$ 를 수정 update

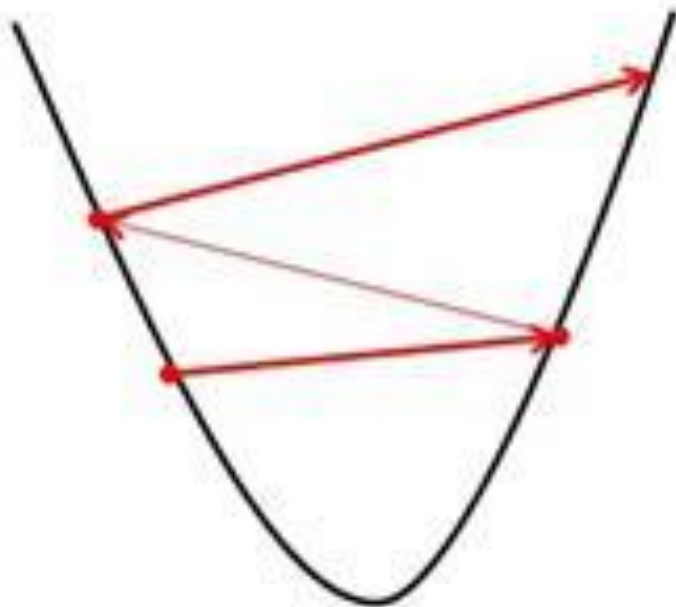
# 경사하강법



# 학습률 Learning rate

## Gradient Descent

Big learning rate



Small learning rate





# 학습률 learning rate

- 파라미터는 한 번에 경사  $\times$  학습률만큼 수정
- 경사가 0에 가까울 수록 오차는 최소에 가까워짐
- 경사에 비례하여 수정할 크기를 조정
- 학습률이 크면 학습이 빠르지만 최소점 근처에서 수렴하지 않을 수 있음

# 경사하강법 예시

- 데이터

y	x
3	1
6	2
9	3
12	4

- 모형

$$y = wx$$

# 경사하강법 (step 1)

- $w = 0$

	$y$	$x$	$\hat{y}$	$y - \hat{y}$	$L$	$\partial L / \partial w$
	3	1	0	3	9	-6
	6	2	0	6	36	-24
	9	3	0	9	81	-54
	12	4	0	12	144	-96
평균					67.5	-45

- 경사가 -45이므로  $w$  를 높여야 오차를 줄일 수 있음

# 파라미터 수정

- 현재  $w = 0$
- 경사 -30
- 학습률 0.1

$$w \leftarrow 0 - 0.1 \times (-45) = 4.5$$

## 경사하강법 (step 2)

- $w = 4.5$

	$y$	$x$	$\hat{y}$	$y - \hat{y}$	$L$	$\partial L / \partial w$
	3	1	4.5	-1.5	2.25	3
	6	2	9	-3	9	12
	9	3	13.5	-4.5	20.25	27
	12	4	18	-6	36	48
평균					16.88	22.5

- 경사가 +22.5이므로  $w$ 를 낮춰야 오차를 줄일 수 있음  
$$w \leftarrow 4.5 - 0.1 \times (-22.5) = 2.25$$

## 경사하강법 (step 3)

- $w = 2.25$

	$y$	$x$	$\hat{y}$	$y - \hat{y}$	$L$	$\partial L / \partial w$
	3	1	2.25	0.75	0.56	-1.5
	6	2	4.5	1.5	2.25	-6
	9	3	6.75	2.25	5.06	-13.5
	12	4	9	3	9	-24
평균					4.22	-11.25

- 경사가 -11.25이므로  $w$ 를 높여야 오차를 줄일 수 있음  
$$w \leftarrow 2.25 - 0.1 \times (-11.25) = 3.375$$

# 경사하강법

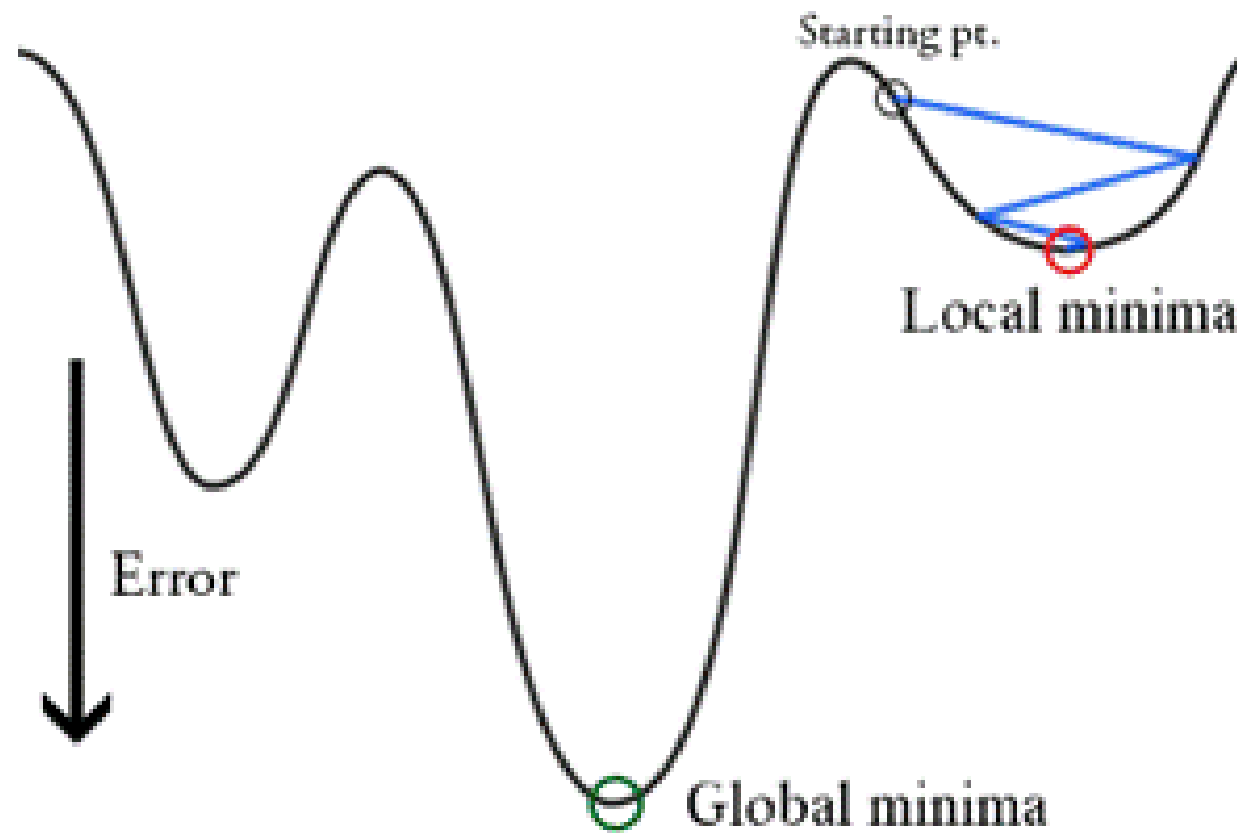
- 경사하강법을 반복하면 오차가 점점 감소
- 오차의 감소폭이 일정 수준 아래가 되면 정지
- 경사하강법의 학습속도를 높이기 위한 다양한 알고리즘이 있음(ADAM 등)



# 경사하강법의 문제점

- 국소최적화
- 한 단계의 가중치 수정을 위해 전체 데이터셋에 대해 계산

# 국소 최적화



# 확률적 경사하강법 Stochastic Gradient Descent

- 하나의 사례를 바탕으로 경사를 계산
- 극소점을 향해 바로 가지 않고 지그재그로 이동하게 됨
- 계산량이 적음
- 국소최적화를 피할 가능성이 있음

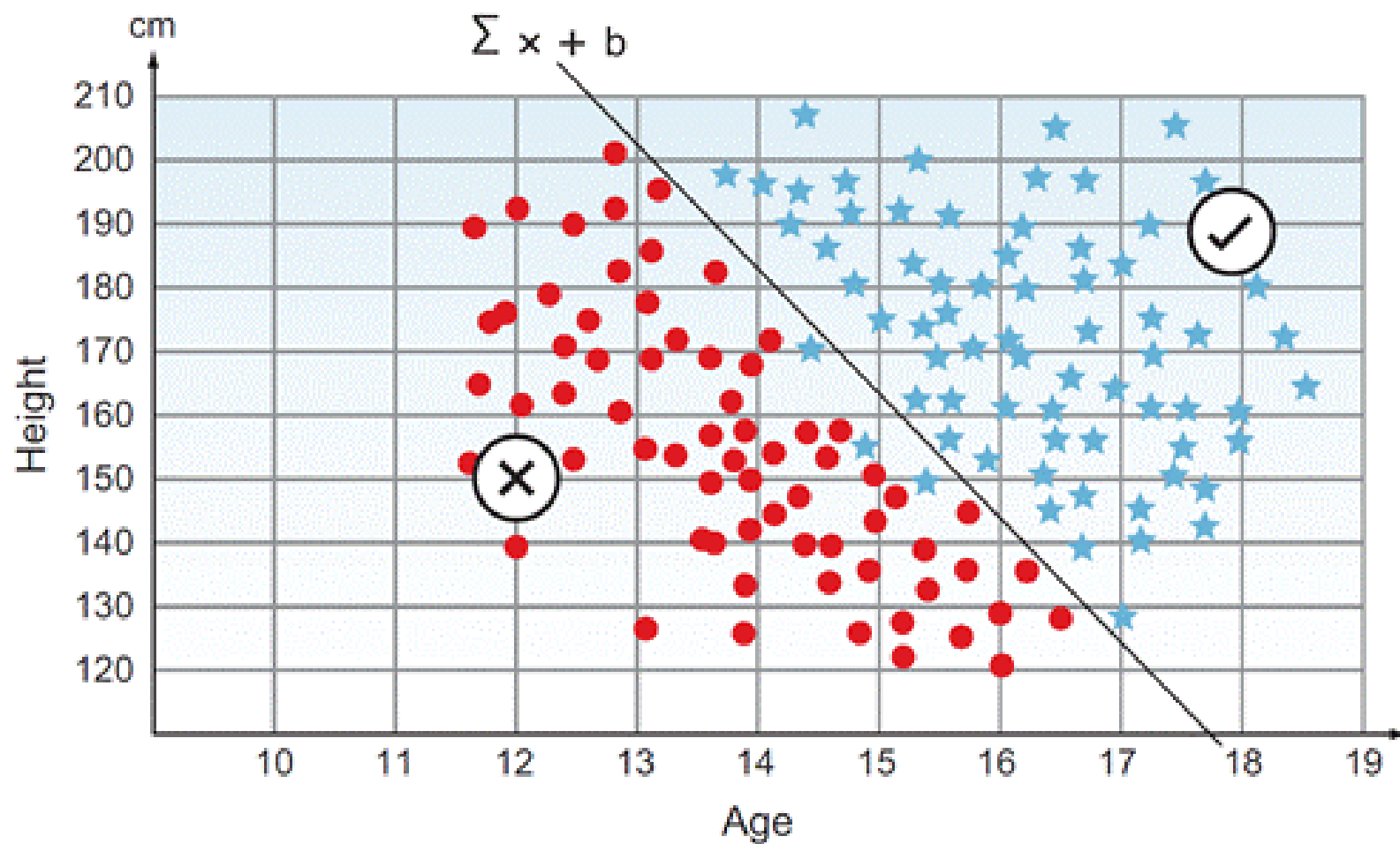
# 미니배치 경사하강법 Mini-batch Gradient Descent

- 전체 데이터의 일부(=미니배치)만을 사용하여 경사하강법
- 경사하강법과 확률적 경사하강법을 장점을 융합

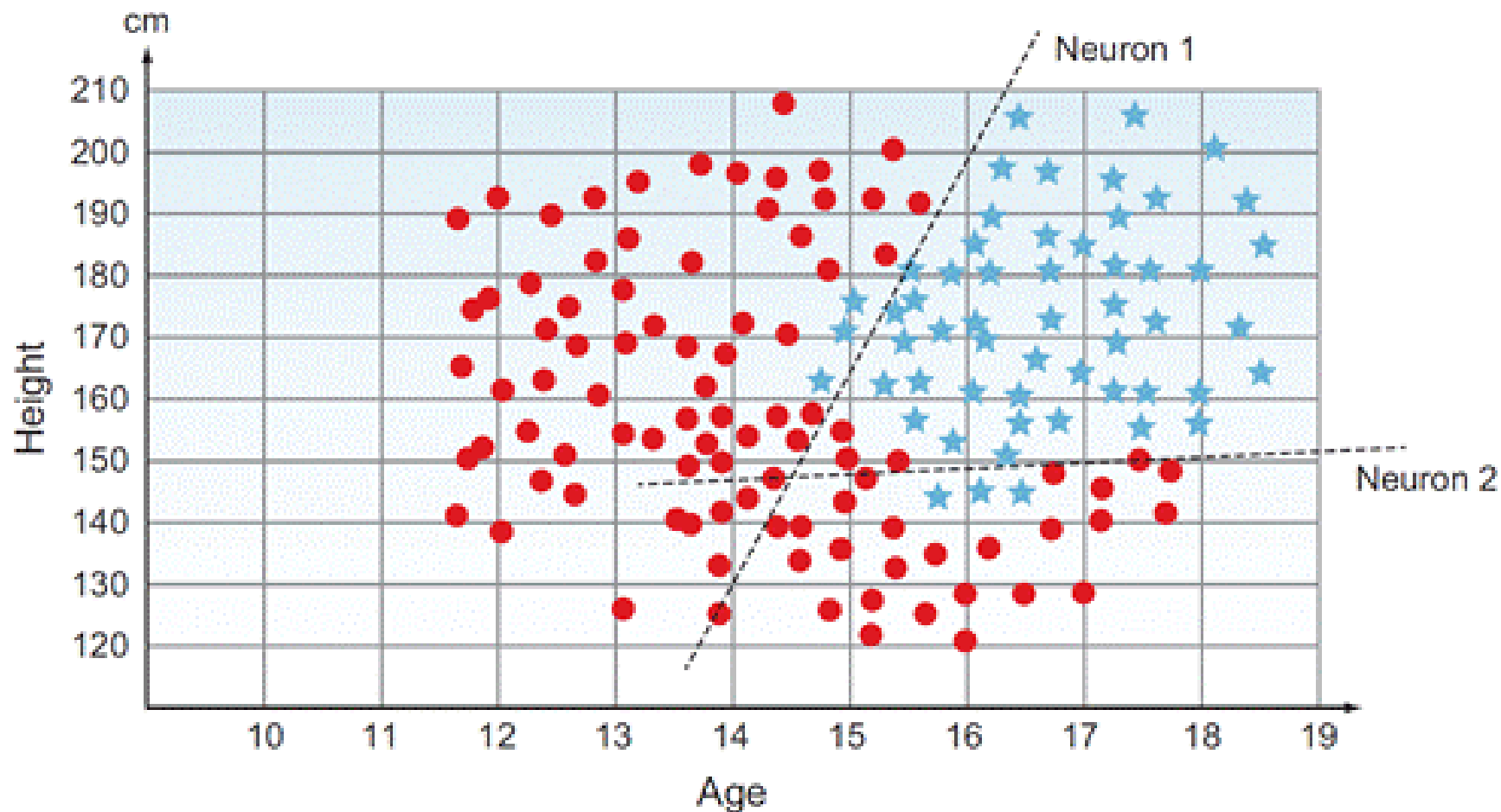
# 경사하강법의 발전

- Nesterov accelerated gradient
- RMSprop
- Adam
- Adagrad

# linearly separable



# linearly non-separable

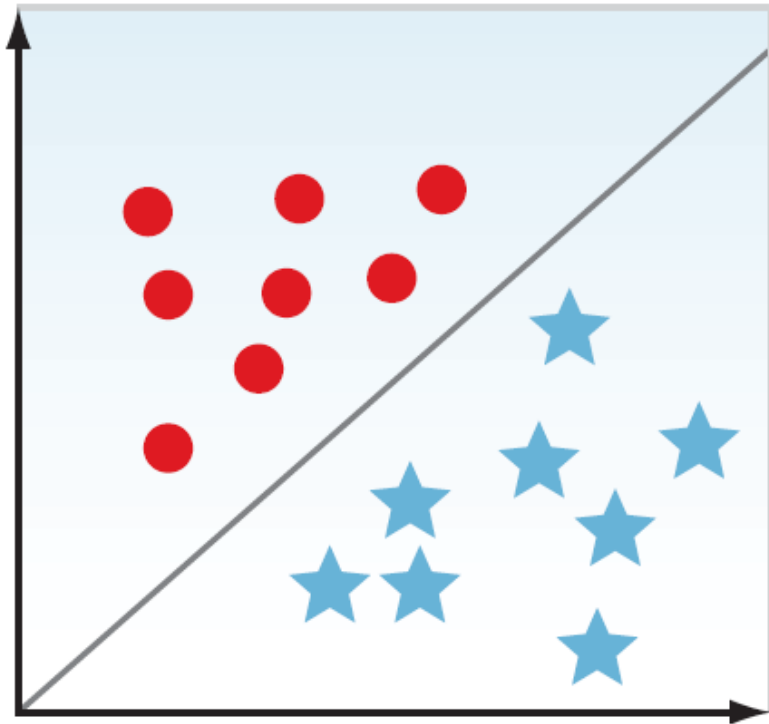




# linear vs. nonlinear

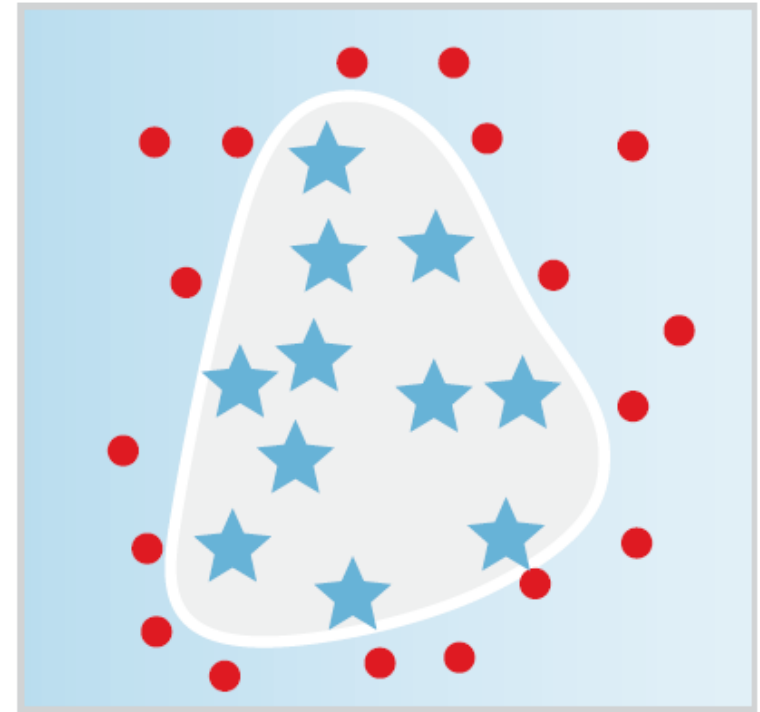
Linear

(can be split by one straight line)

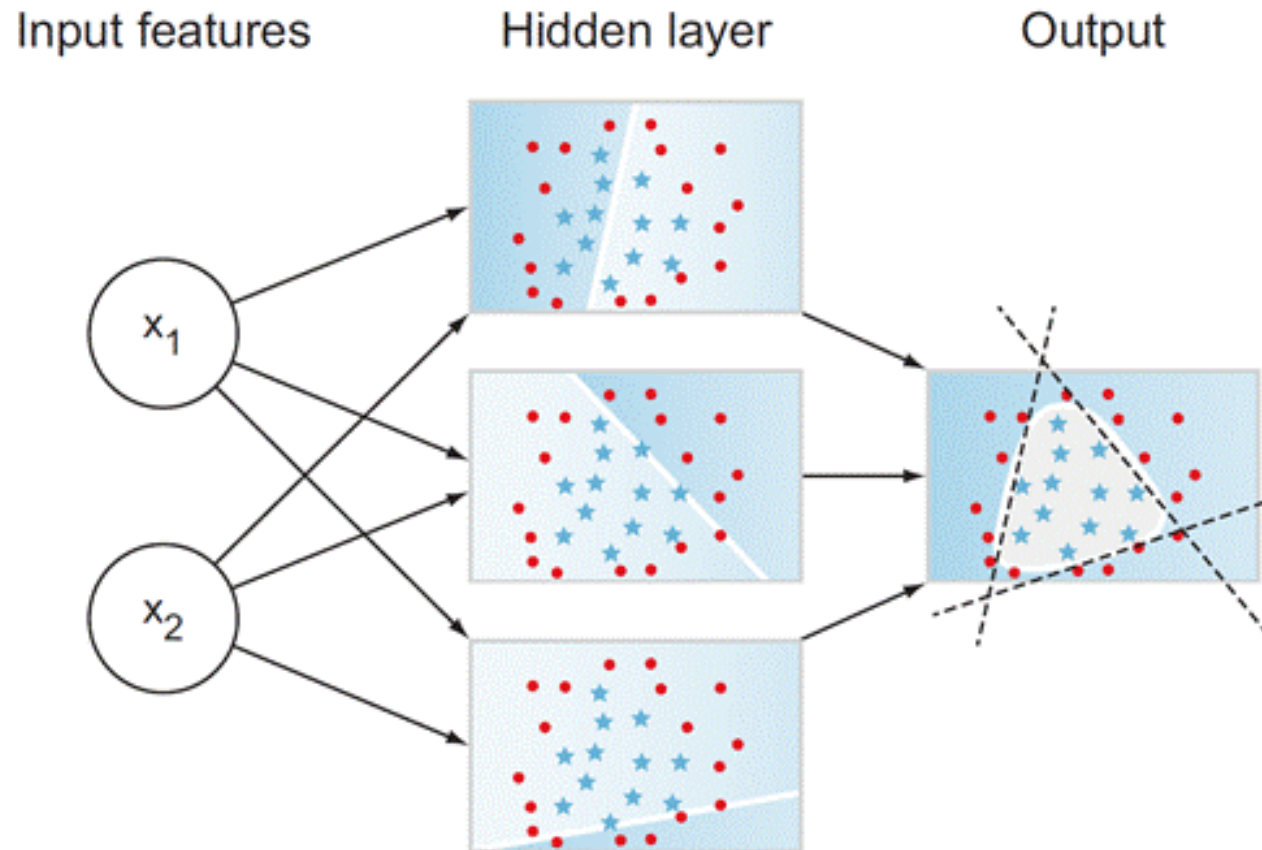


Nonlinear

(Need more than one line to split the data)



# 다층신경망 Multi-Layer Perceptron



# 다층 신경망의 구조

1. 입력층 Input layer

2. 은닉층 Hidden layers

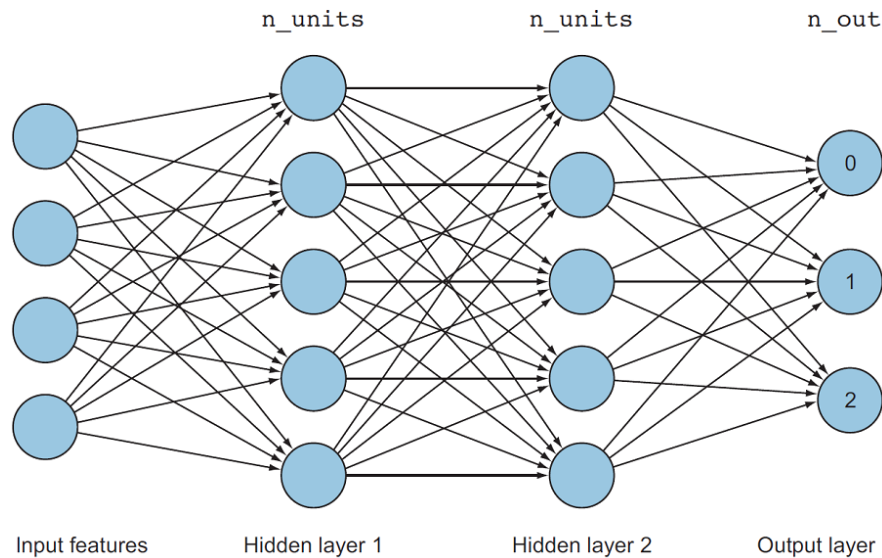
3. 출력층 Output layer

# 은닉층의 역할

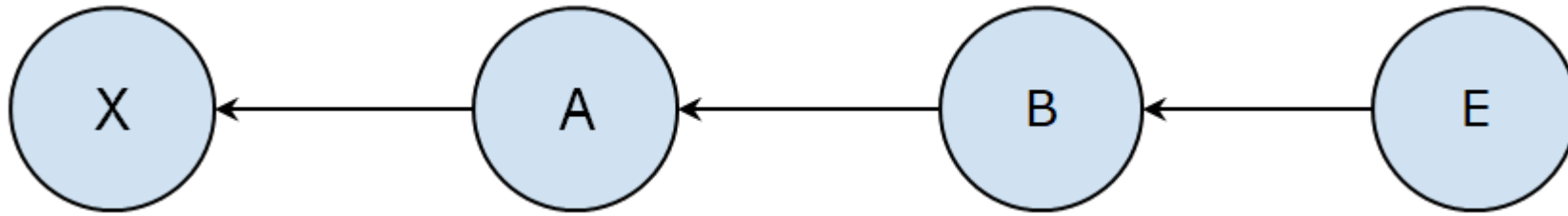
- 신경망의 학습 과정에서 과제 수행에 필요한 특징을 학습
- 입력층에 가까운 은닉층은 단순한 특징을, 출력층에 가까운 은닉층은 복잡한 특징을 학습
- 은닉층의 적절한 수와 크기는 문제에 따라 달라짐 (다양한 시도를 통해 가장 성능이 좋은 것으로 결정)

# Fully Connected or Dense layer

- 레이어의 모든 입력이 레이어의 모든 출력에 영향을 주는 형태
- 입력이  $n$ 개이고 출력이  $m$ 개이면 가중치의 개수가  $n \times (m+1)$ 개

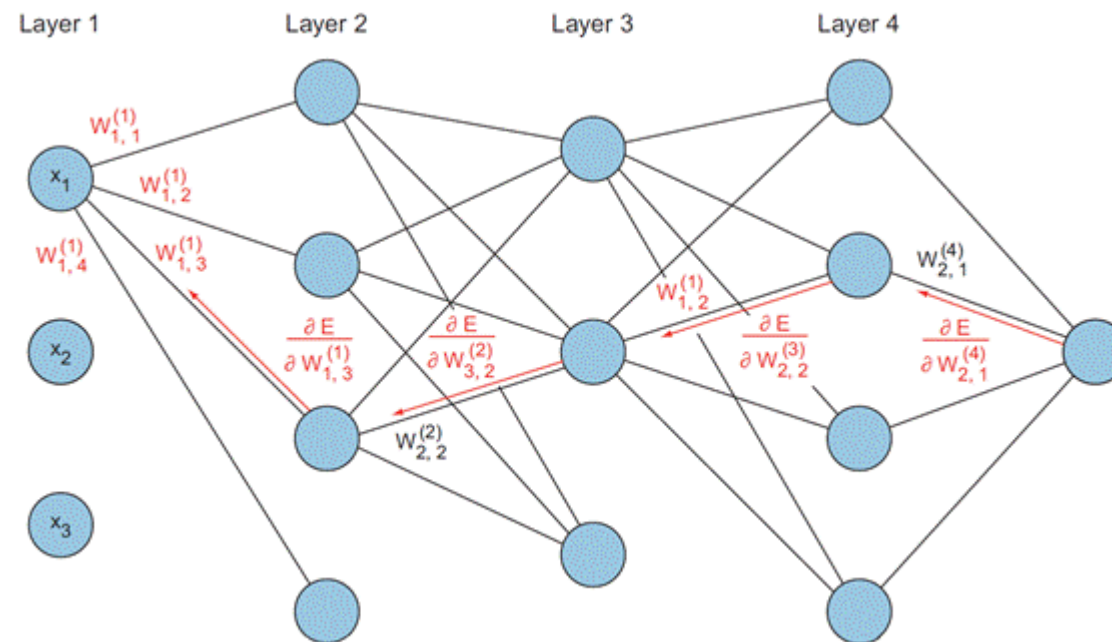


# Chain rule



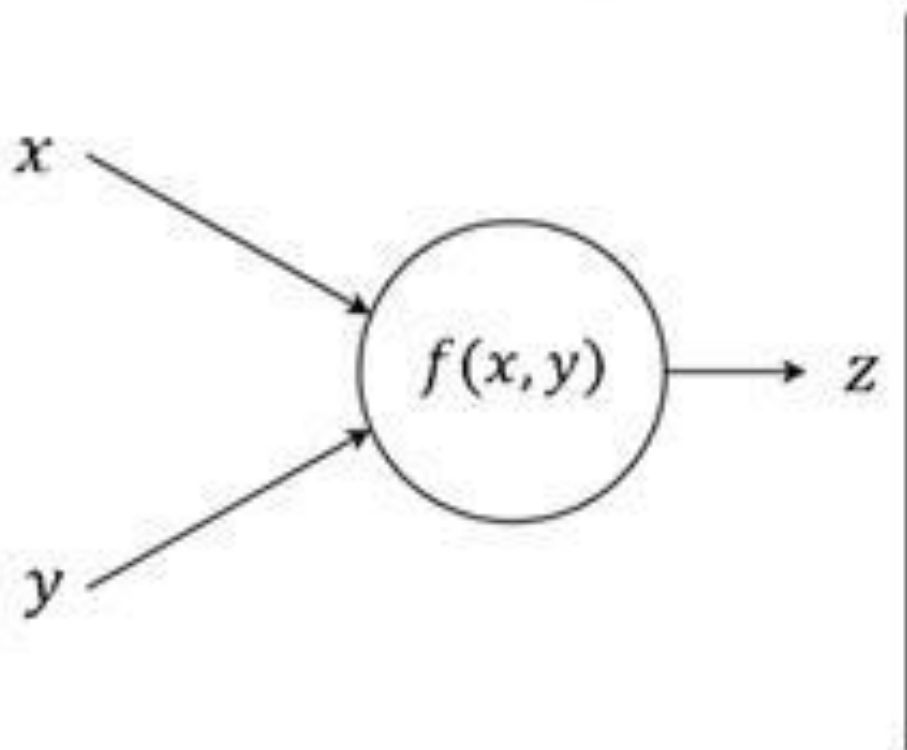
$$\frac{dE}{dx} = \frac{dE}{dB} * \frac{dB}{dA} * \frac{dA}{dx}$$

# 역전파 Backpropagation

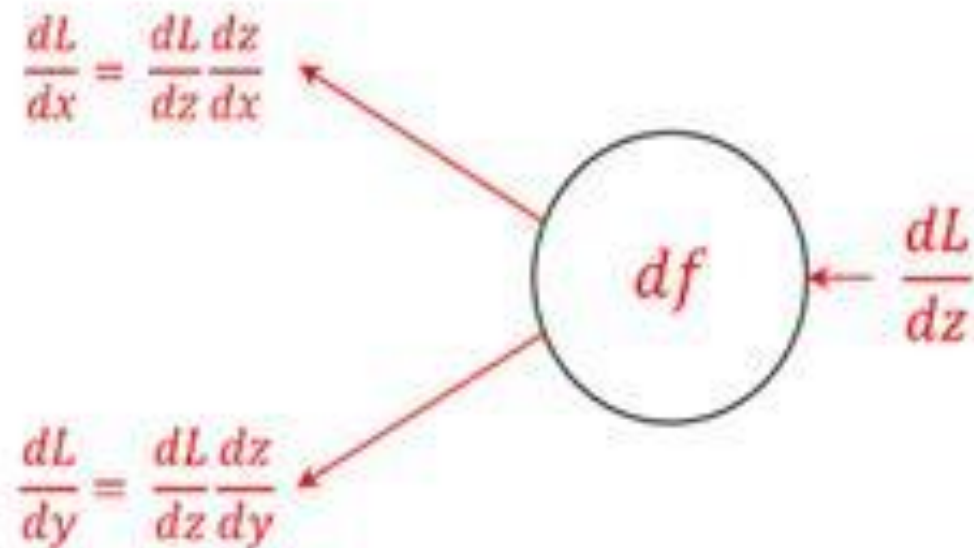


# 앞먹임과 역전파

Forwardpass



Backwardpass



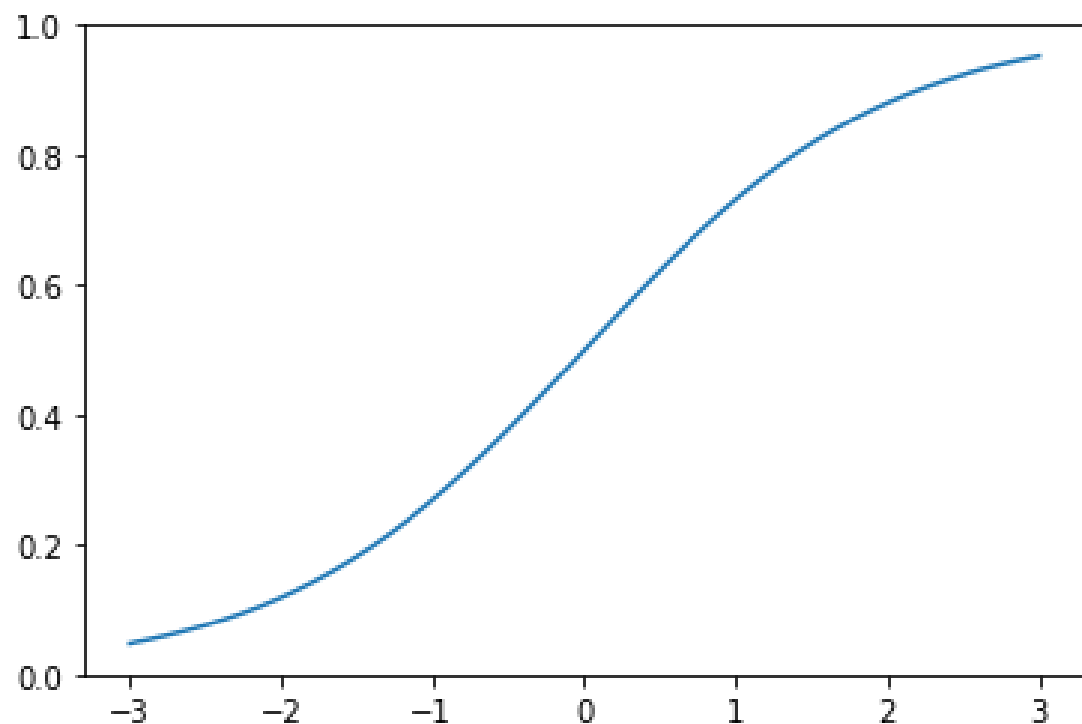


# 활성화 함수 Activation functions

- linear or identity function
- step function
- sigmoid(logistic) function
- softmax
- tanh
- ReLU
- Leaky ReLU

# 로지스틱 함수 logistic function

- 출력층에서 0~1 사이의 출력을 표현하기 위해 사용
- 은닉층에 사용할 때는 문제가 있음



# Not Zero Centered

- 로지스틱 함수는 출력이 0~1 범위로 항상 양수
- 경사의 부호가 항상 같음
- 한 번에 일부 파라미터는 높이고, 일부 파라미터는 낮추는 것이 불가능

# 포화 saturation

- 함수의 양쪽 끝에서 경사가 거의 0
- 입력값이 변하더라도 출력값에 거의 차이가 없음
- 신경망 앞쪽 레이어에서 변화가 뒤쪽 레이어에 영향을 주지 못함
- 앞쪽 레이어로 오차 역전파가 잘 이뤄지지 않음

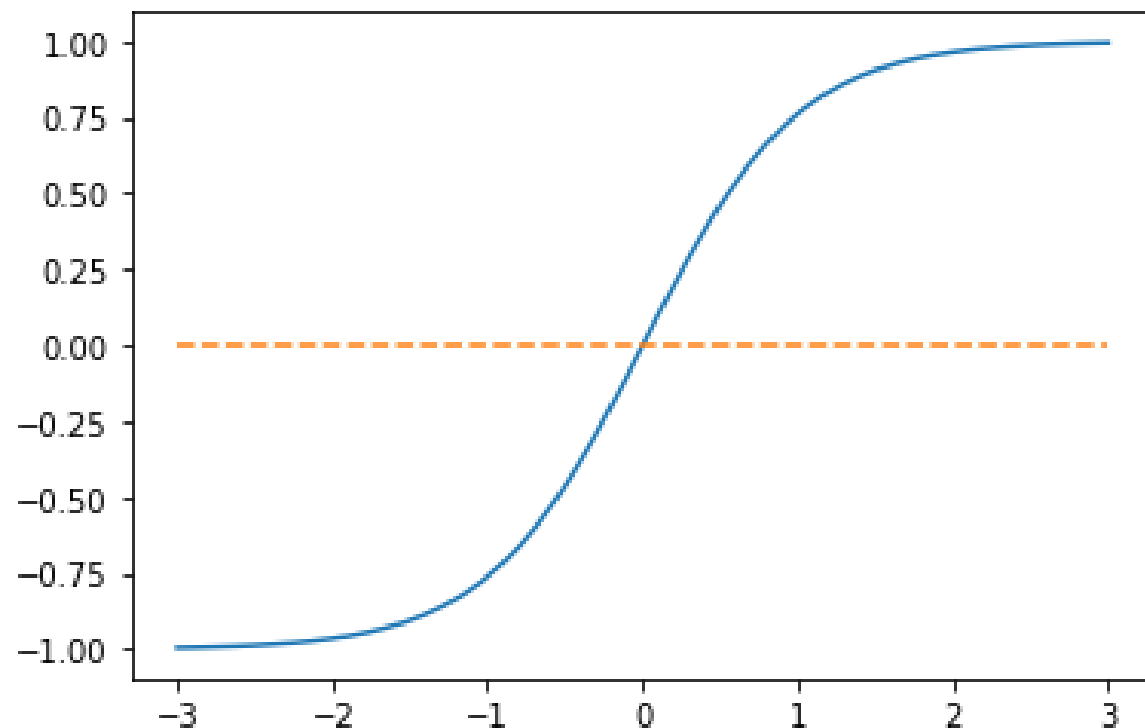
# 소프트맥스 함수 softmax

- 이항분류에서는 로지스틱 함수로  $-\infty \sim +\infty$ 의 값을 0~1 범위로 변환
- 다항 분류에서는 소프트맥스 함수를 사용
- 여러 개의 입력을 받아, 같은 개수를 출력
- 모든 출력의 합은 1, 각 출력의 범위는 0~1
- `softmax([-1., 0.5, 2.]) == [0.039, 0.175, 0.796]`

$$y_j = \frac{\exp(x_j)}{\sum_i \exp(x_i)}$$

# 쌍곡탄젠트 Hyperbolic Tangent

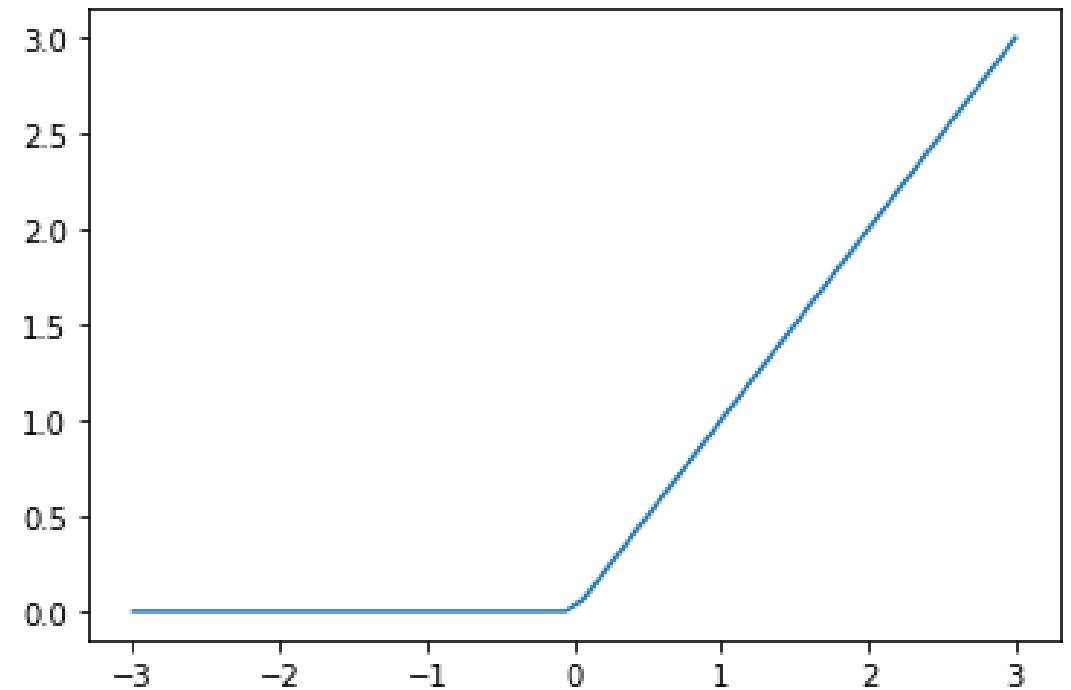
- 로지스틱 함수와 비슷 하게 생겼지만 출력 범위가  $-1 \sim 1$
- zero-centered
- 포화 문제는 동일



# Rectified Linear Unit

$$\begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases}$$

- not zero-centered
- 포화 문제 완화
- 계산이 간단
- 은닉층에 흔히 사용



# 손실 함수 loss function

- 비용 함수 cost function 또는 오차 함수 cost function라고도 함
- 예측과 실제의 차이를 계산
- 손실  $\geq 0$
- 손실을 최소화할 수록 성능이 개선



# MSE

- 오차 = 실제 - 예측
- 오차 제곱의 평균
- 연속변수의 예측에 사용
- 이상치 outlier에 민감

# MAE

- 오차의 절대값의 평균
- 연속변수의 예측에 사용
- 이상치에 둔감

# 교차 엔트로피 cross entropy

- 두 확률 분포의 차이를 계산
- 범주형 변수의 예측에 사용

$$-\sum y \log f(x)$$

- 높은 확률로 예측했을 때 맞고, 낮은 확률로 예측했을 때 틀려야 교차 엔트로피가 감소