Lecture note 5 : Machine Learning

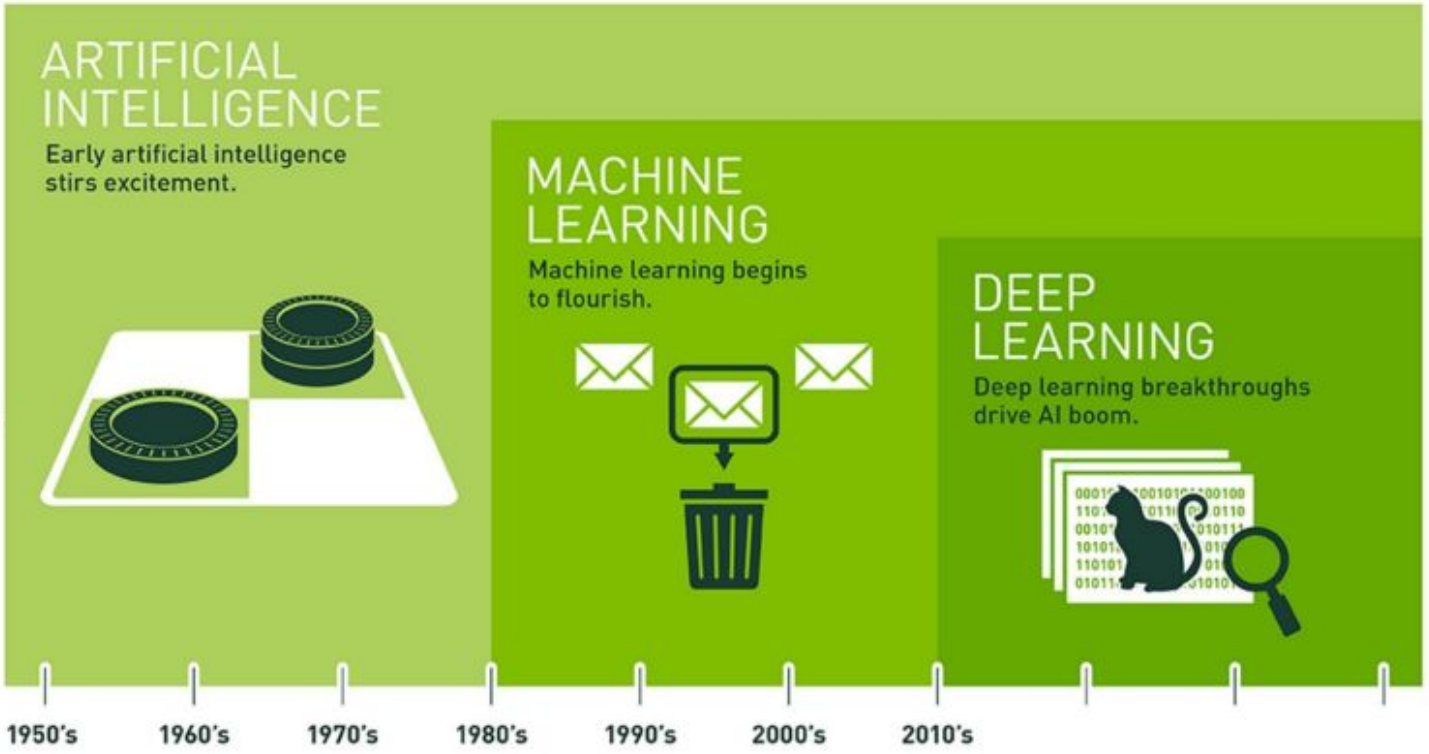
프로젝트 기반 딥러닝 이미지처리
한국인공지능아카데미 x Hub Academy

강사 : 김형욱 (hyounguk1112@gmail.com)

Outline

1. Introduction
2. Intro. To Machine Learning
3. 선형회귀 알고리즘
4. 선형분류 알고리즘

From AI to Deep learning

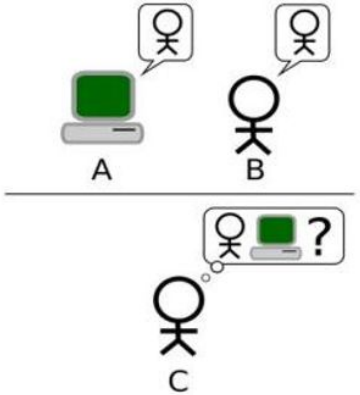


튜링 머신 : 인공지능 개념의 시작

Q. 기계가 생각할 수 있는가?

A. **"지성 있는 사람이 관찰하여 기계가 진짜 인간처럼 보이게 하는데 성공한다면 확실히 그것은 지능적이라고 간주해야 한다"**

- Alan Turing



But, It's not enough now...

인공지능의 정의

Q. 인공지능이란?

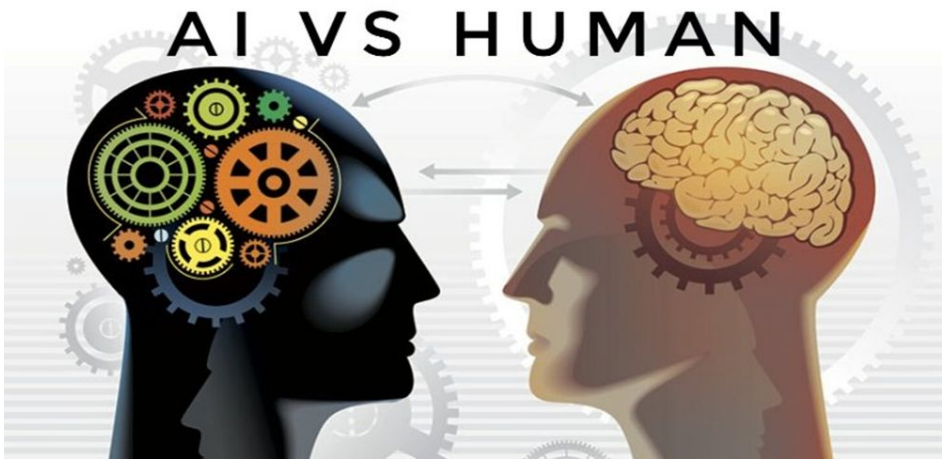
A. 인간처럼 임의의 문제를 해결할 수 있는 기계적 혹은 생체적 장치를 의미함. 그러나 명확하게 합의된 정의는 아직까지 없음.

Q. 그렇다면 자연지능과의 관계는?

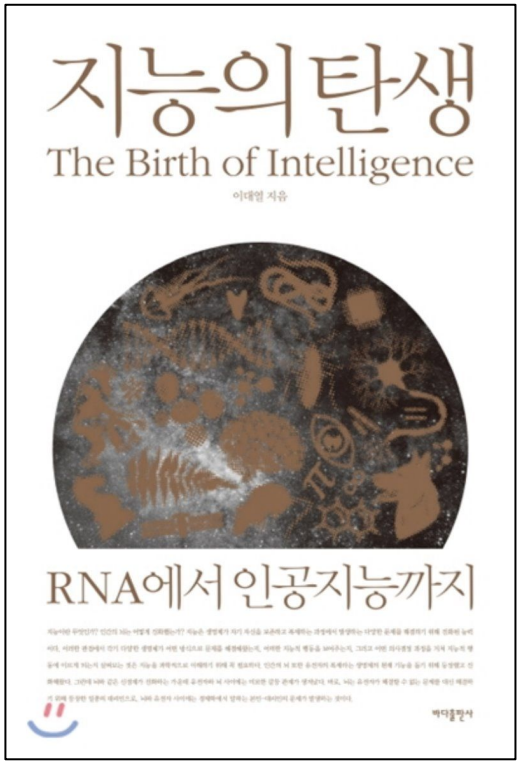
A. 자연지능은 생존과 번식이라는 자연에 의한 일관된 목표를 달성하기 위해 진화/발전해온 생물체의 기능임. 우리가 아는 범위에서 고도로 어려운 문제를 풀 수 있는 존재는 인간이기 때문에 인간을 모방하는 방향으로 발전해왔으나 기본적으로 '자연지능'의 목적과 달리 인간을 대신하여 복잡한 문제를 해결하는데 목적이 있기 때문에 차별성을 갖는다.

[1] Kaplan, Andreas; Haenlein, Michael (1 January 2019). "Siri, Siri, in my hand: Who's the fairest in the land? On the interpretations, illustrations, and implications of artificial intelligence". Business Horizons. 62 (1): 15–25. doi:10.1016/j.bushor.2018.08.004

인공지능과 인간지능



인공지능의 개념은 인간지능을 이해하고 모방하는 방향으로 발전해왔다. 그렇다면 인공지능도 지능이 될 수 있을까?



인공지능의 지엽적 정의

A system's ability to correctly interpret external data, to learn from such data, and to use those learnings to achieve specific goals and tasks through flexible adaptation."[1]

[1] Kaplan, Andreas; Haenlein, Michael (1 January 2019). "Siri, Siri, in my hand: Who's the fairest in the land? On the interpretations, illustrations, and implications of artificial intelligence". Business Horizons. 62 (1): 15–25. doi:10.1016/j.bushor.2018.08.004

Brief history of Artificial Intelligence

1950

Turing test

Alan Turing proposes a test for computer intelligence: A computer would have passed when its text-based conversation is indistinguishable from humans

1956

'Artificial intelligence' coined

Leading minds gather to discuss possibility of machines that think at Dartford University in New Hampshire. The term stuck

1968

2001: A Space Odyssey

Stanley Kubrick's space epic introduced HAL 9000, a paranoid computer that attempted to kill the spacemen on the spacecraft it controlled.

1970s

AI Winter

Amid disappointment about a lack of progress, organisations including US government arm DARPA reduce investment

1997

Deep Blue defeats Kasparov

IBM's computer beats the world chess champion over six games

1984

Terminator released

Arnold Schwarzenegger's blockbuster has been the doomsday scenario for artificial intelligence since it was released

2016

DeepMind beats champion Go player

Seen as a major breakthrough, the deep learning system used by DeepMind's AlphaGo breaks one of the holy grails of AI

2012

Driverless cars

Google's driverless cars, announced in 2009, make their way onto California's roads

2011

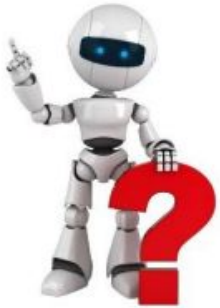
Watson wins Jeopardy

Another IBM victory: The artificially-intelligent system defeated two human players at the popular quiz show

2011

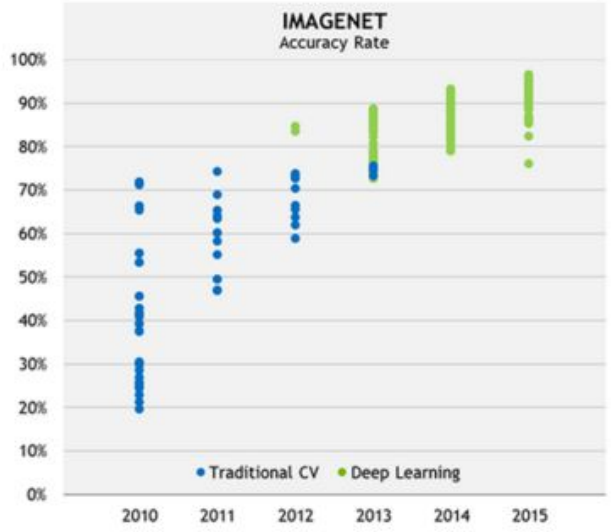
Siri

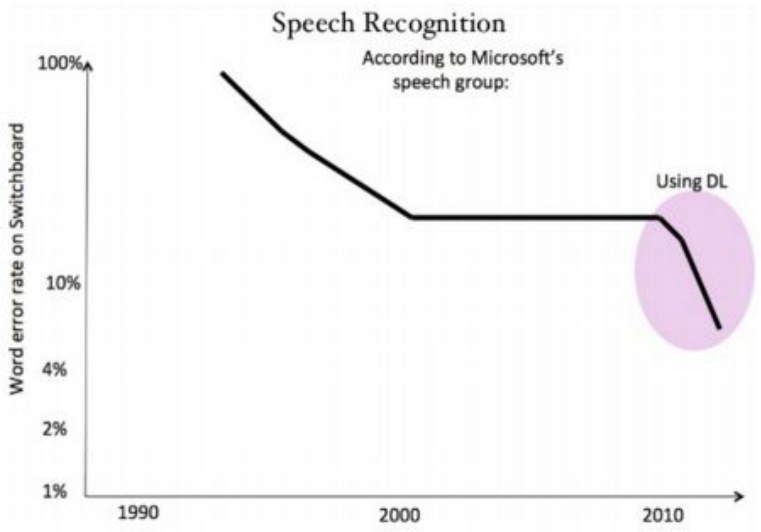
Apple's intelligent assistant debuts on the iPhone. It has improved dramatically since then



Deep Learning Make Machines to have ‘Perception’

Deep Learning is Driving Recent Major Breakthroughs in Visual and Speech Recognition Tasks





Now, Machines Learn Deep

Now, Machines Beat Human in Tasks Once Considered Impossible

VS

5:0
vs Fan Hui
(Oct. 2015)

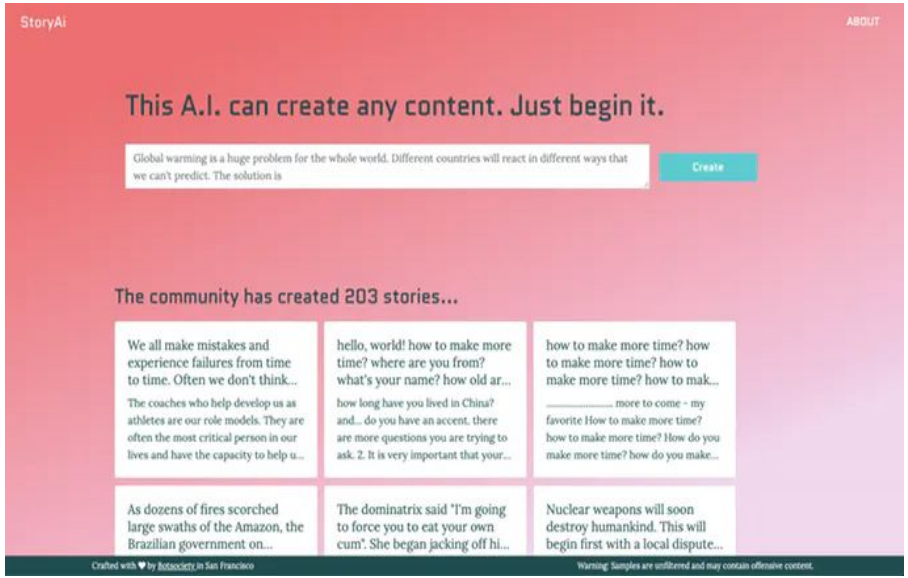
4:1
vs Sedol Lee
(Mar. 2016)

Now, Machines Learn Deep

Now, Machines do Tasks which we believed only humans can do

[1]

[2]



[1] https://www.smithsonianmag.com/smart-news/ai-system-produces-new-styles-art-180963912/
[2] https://www.producthunt.com/posts/story-ai

Explaining Deep Learning

You could think of Deep Learning as the building of learning machines, say pattern recognition systems or whatever, by assembling lots of modules or elements that all train the same way.

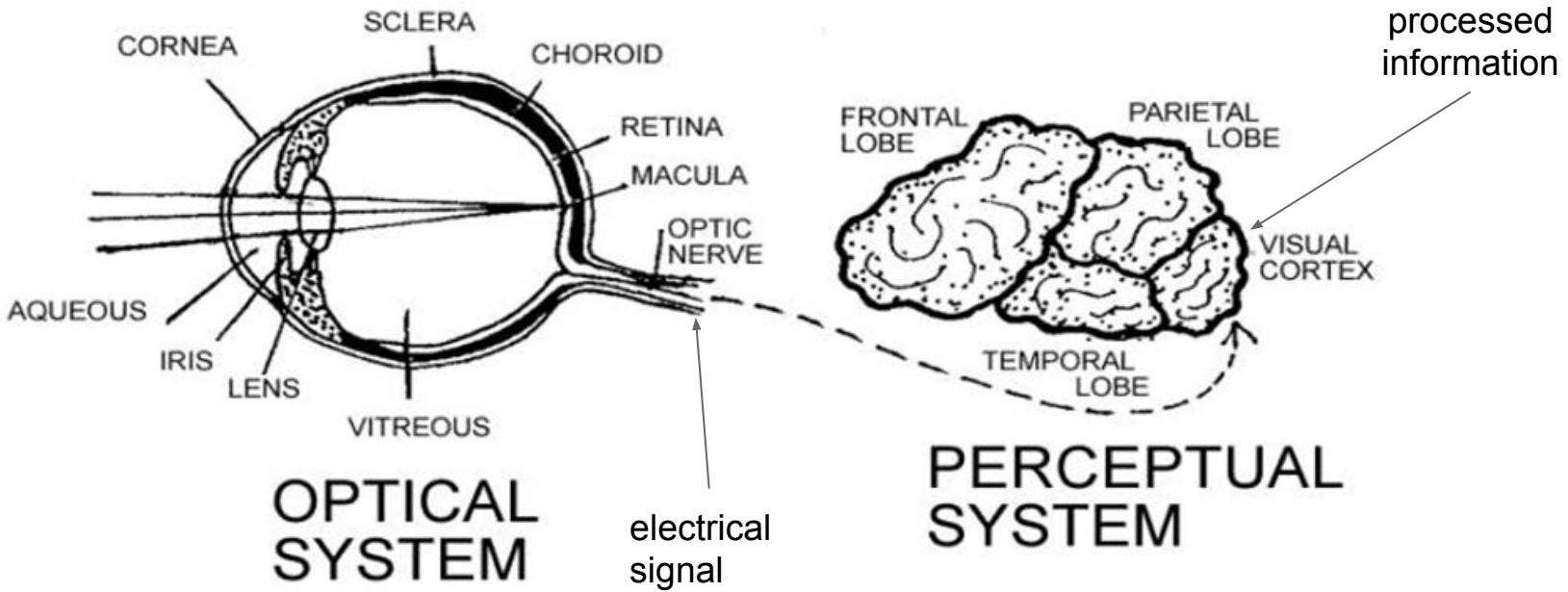
So there is a **Single Principle** to train everything.

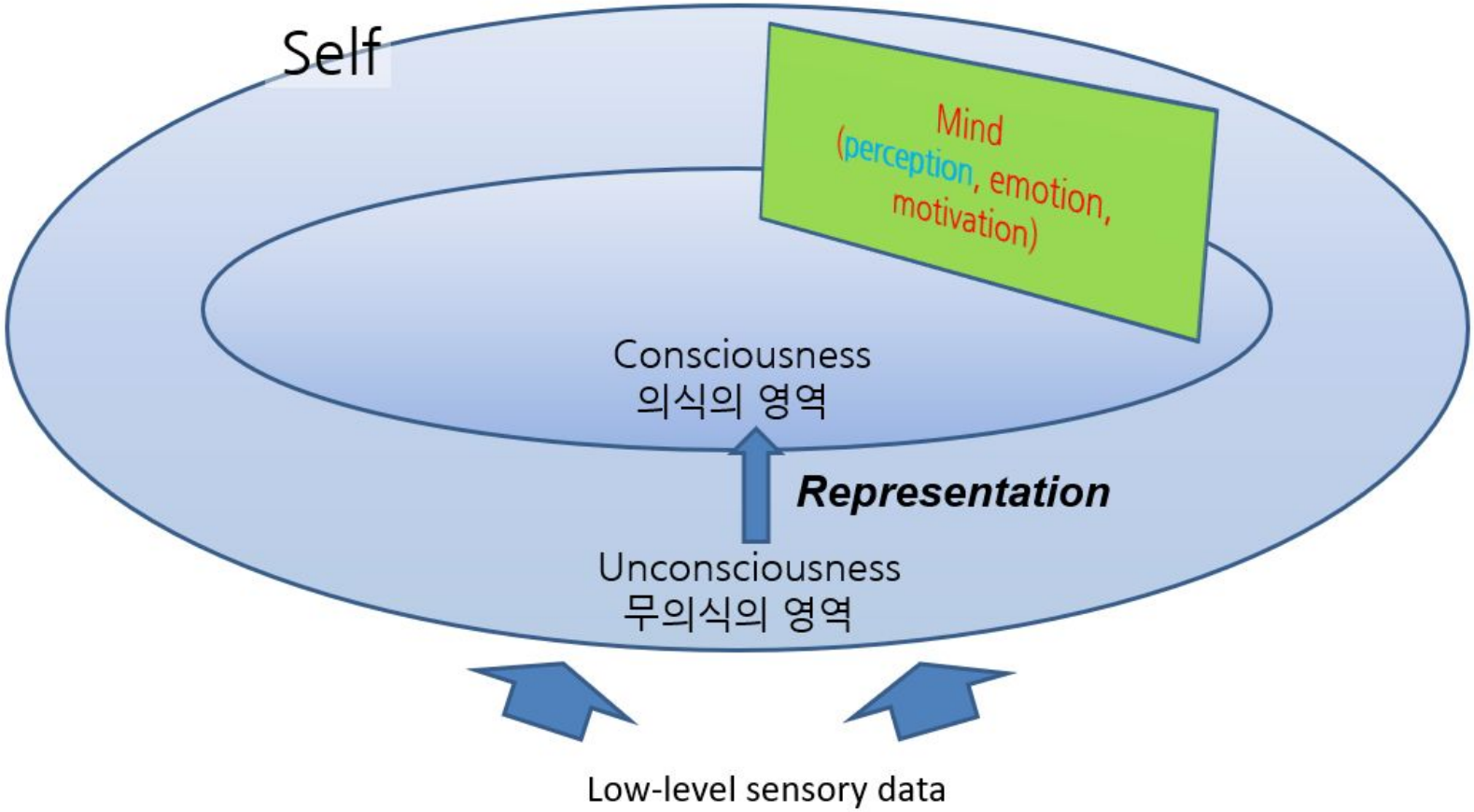
IEEE Spectrum, Feb. 2015

WIKIPEDIA
The Free Encyclopedia

Deep learning is a branch of machine learning based on a set of algorithms that attempt to model **high level abstractions** in data by using a deep graph with multiple processing layers, composed of multiple linear and non-linear transformations.

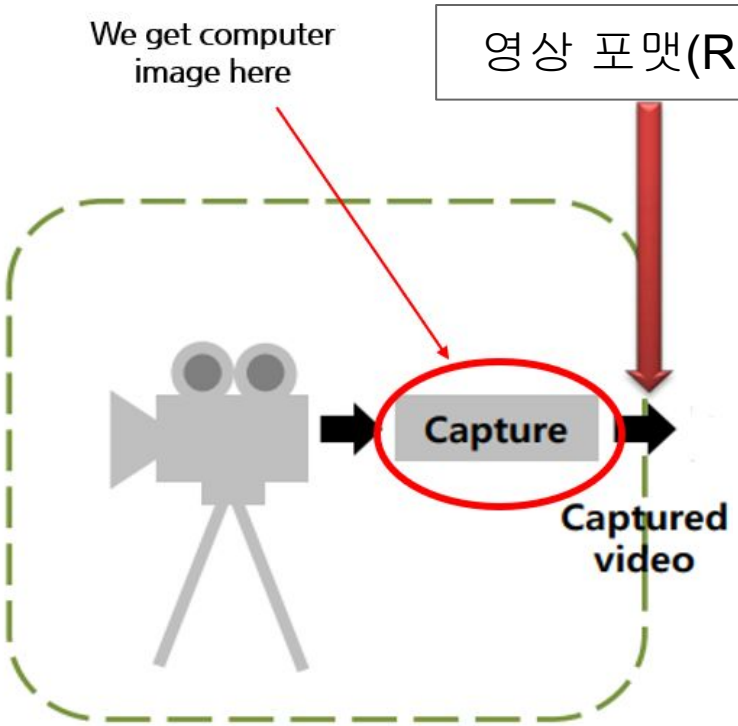
Human vision system

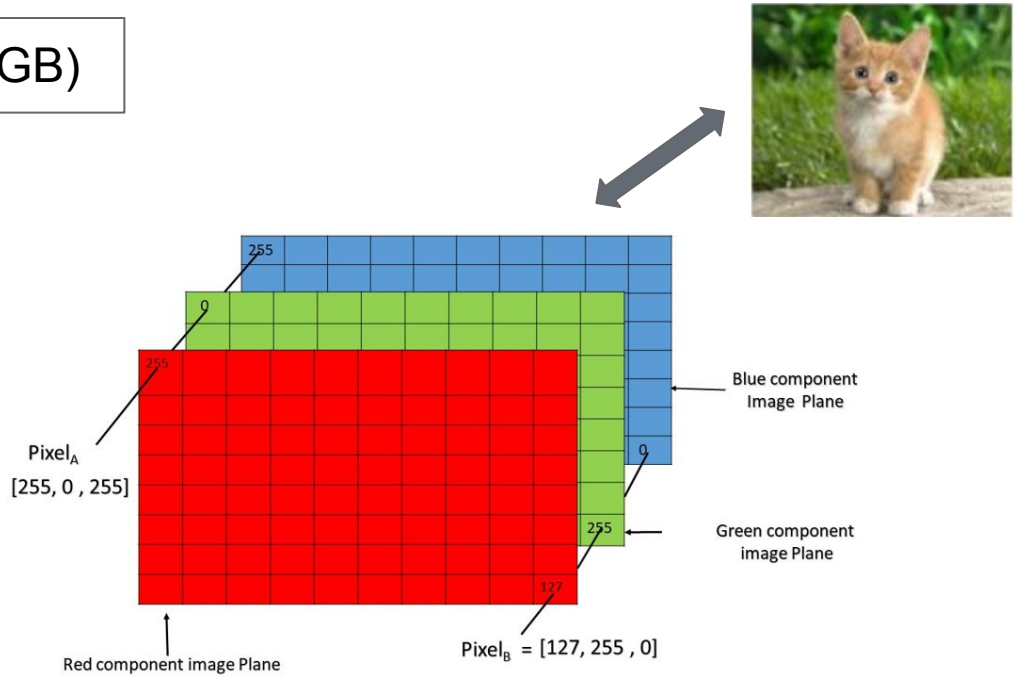




<Information processing in human brain>

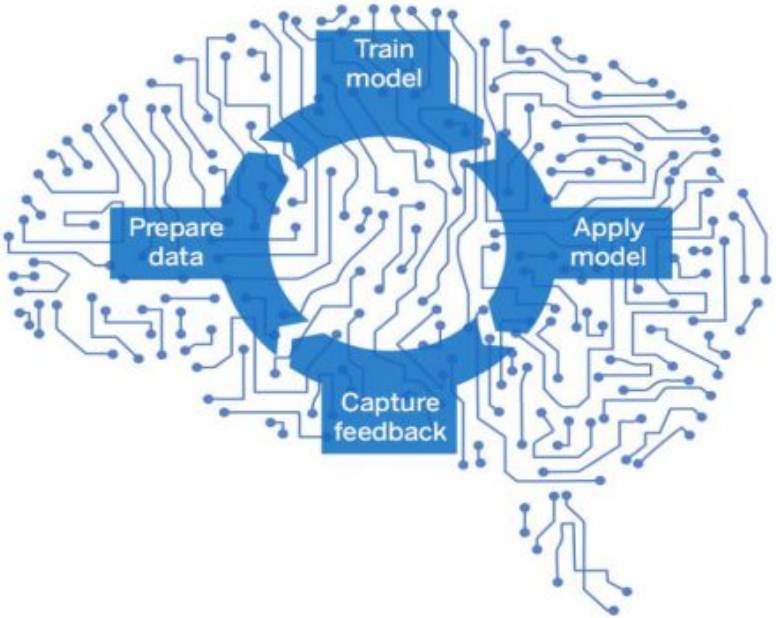
Machine vision system

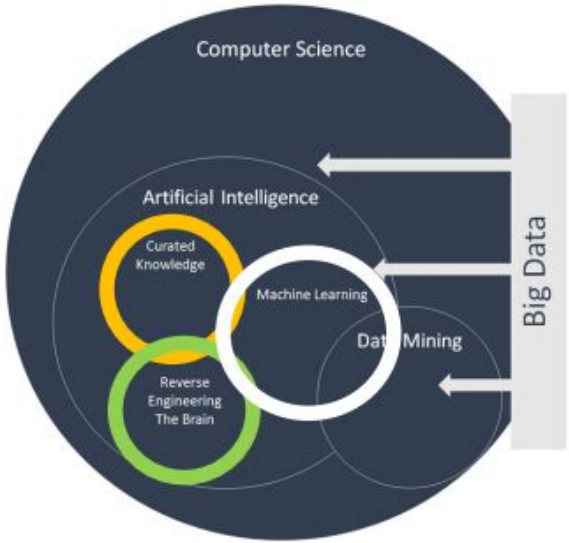




Artificial Intelligence diagram

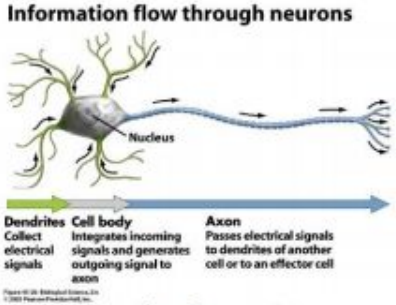
From Knowledge-based Approach to Data-driven Approach



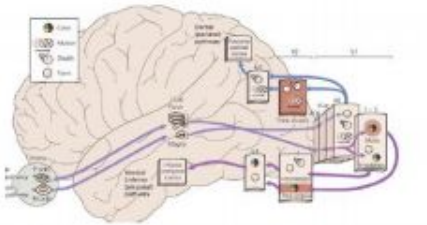


Brain-inspired Learning

Learn massive data

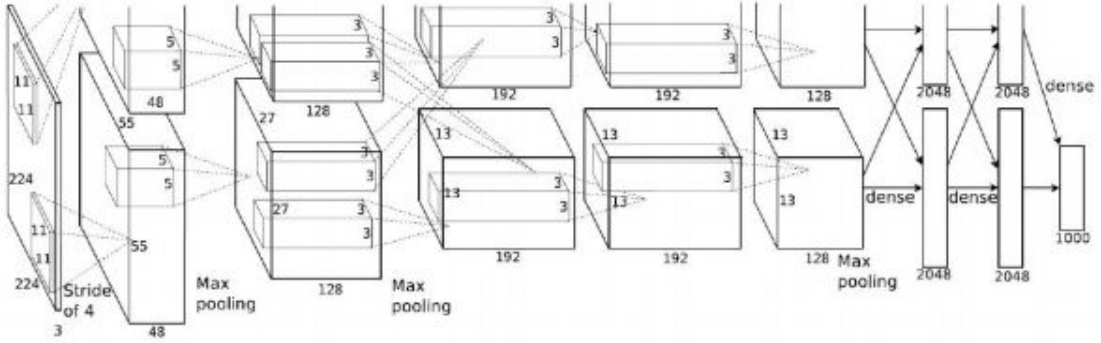


simple functions



(Van Essen&Gallant, 1994)

Multi-layered



Intro. to ML

머신러닝의 정의

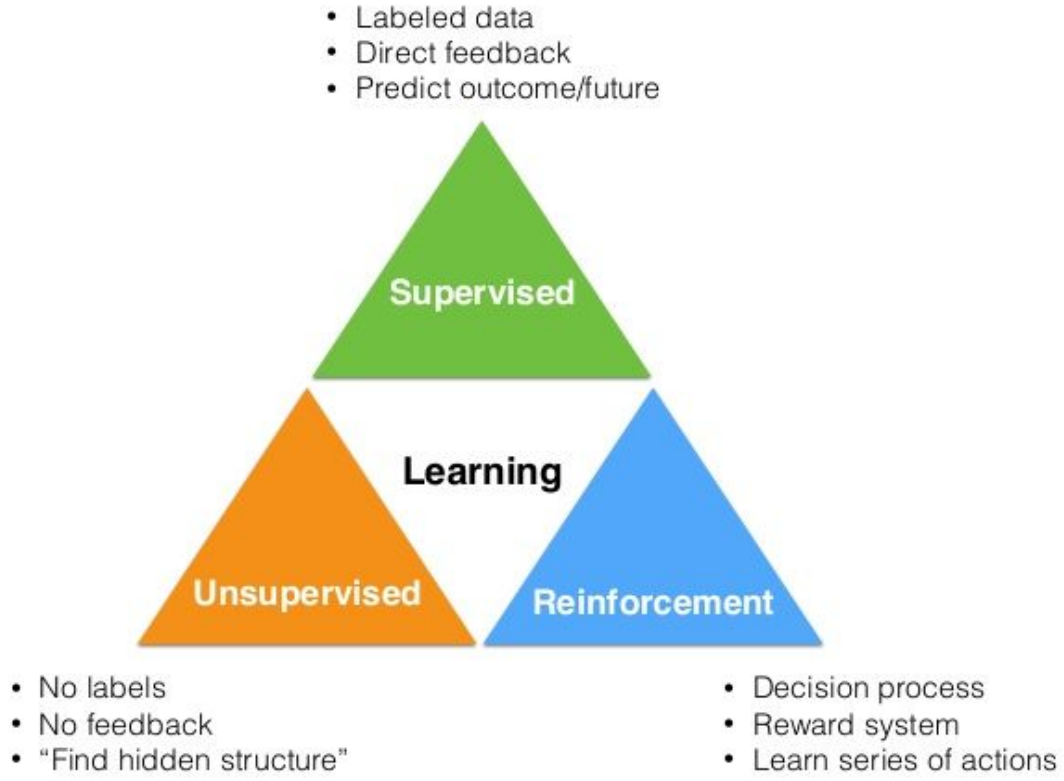
What is Machine Learning?

- **Simple answer**
  인공지능의 한 분야로 데이터로부터 컴퓨터가 특정 문제를 해결하기 위한 함수나 알고리즘을 학습하는 것

- **Deep answer**

  A computer program is said to **learn from experience E** with respect to **task T** and **performance measure P**, if its performance at tasks T, improves with experience E
  Tom Mitchell, 1957

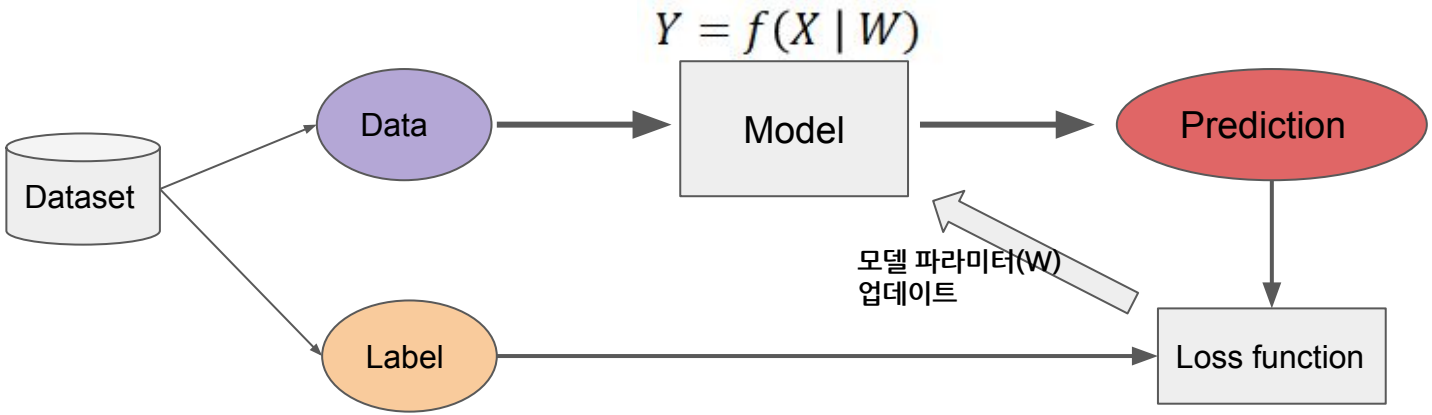
머신러닝의 분류(학습방법)



머신러닝의 분류(학습방법)

(1) Supervised learning

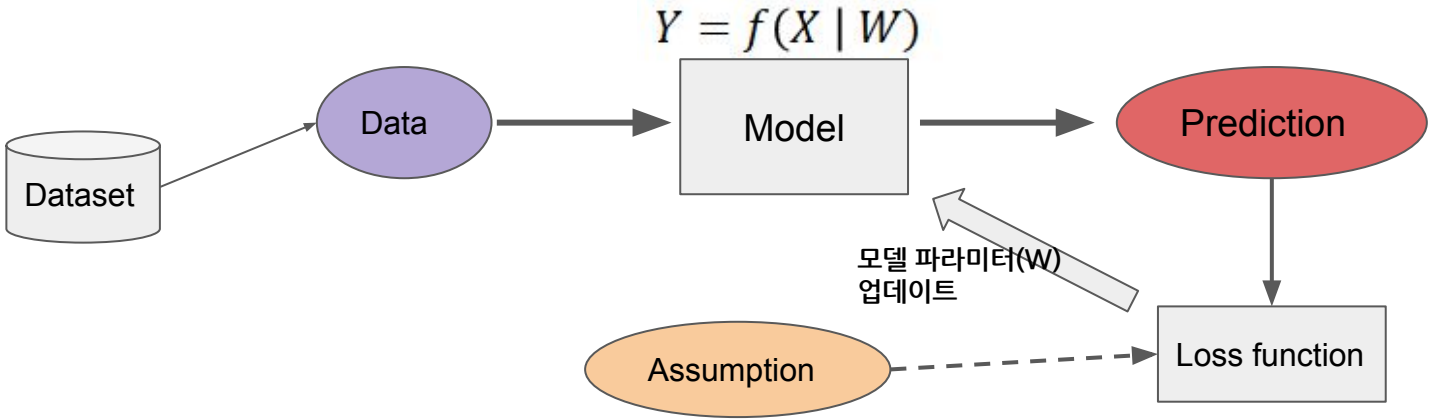
- 지도학습(Supervised learning) 방식의 학습은 **모델 f(x)** 와 **데이터 x** 그리고 **레이블 y** 로 구성된다.
- 목표는 모델 함수가 데이터를 입력으로 받아 **레이블과 동일한 정답값 또는 분포를 예측**하도록 모델의 파라미터를 최적화시키는 것이다.
- 크게 회귀(Regression)과 분류(Classification)으로 나눌 수 있다.
- 비용(Loss)를 낮추는 것은 학습 과정의 일관된 하나의 목표이다.



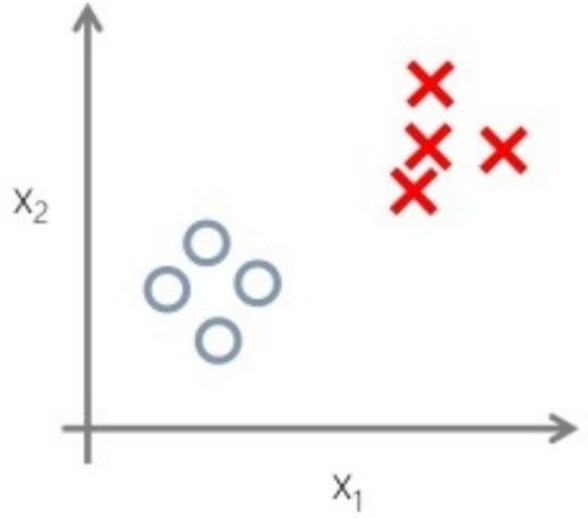
머신러닝의 분류(학습방법)

(2) Unsupervised learning

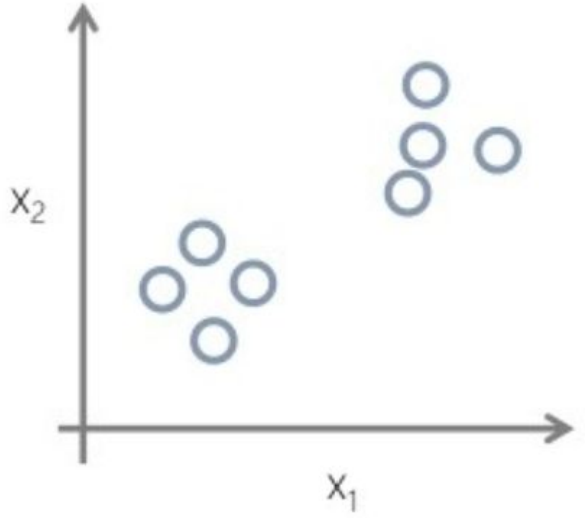
- 비지도학습(Unsupervised learning) 방식의 학습은 **모델 f(x)** 와 **데이터 x** 로 구성된다.
- 목표는 모델 함수가 데이터를 입력으로 받아 **각 결과의 패턴 또는 결과들의 분포를 바탕으로** 학습하도록 모델의 파라미터를 최적화시키는 것이다. 이는 대개 문제에 대한 강력한 가정(Assumptions) 또는 제약(Constraints)를 알고 있기 때문에 가능하다.
- 대표적으로 클러스터링과 생성 모델이 속한다.
- 마찬가지로 비용(Loss)를 낮추는 것은 학습 과정의 일관된 하나의 목표이다.



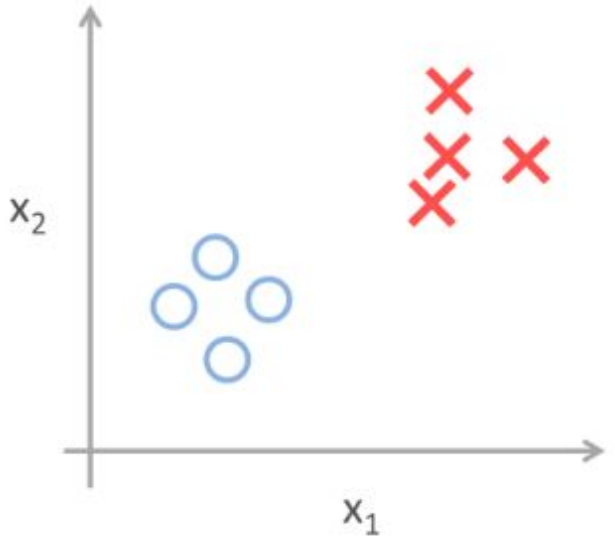
Supervised Learning



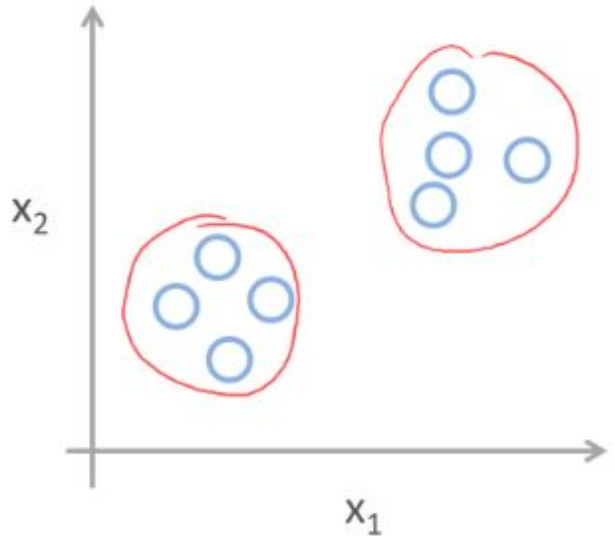
Unsupervised Learning



Supervised Learning



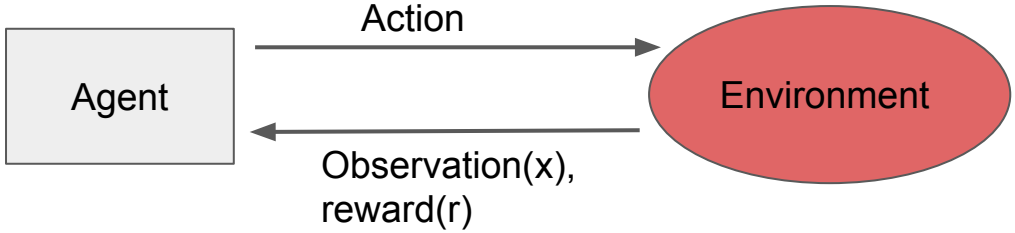
Unsupervised Learning



머신러닝의 분류(학습방법)

(3) Reinforcement learning

- 강화학습(Reinforcement learning) 방식의 학습은 **에이전트 f(x,r |s,p)** 와 **관측 정보 x와 보상치 r**로 구성된다.
- 목표는 에이전트가 주어진 환경과 현재의 상태에서 장기적으로 가장 최대한의 보상을 받는 행위를 하도록 자신의 정책(Policy)를 학습하는 것이다.
- 대표적으로 로봇제어와 게임봇이 강화학습의 응용에 속한다.
- 장기적인 기대 보상(reward)를 극대화하는 것이 학습 과정의 일관된 하나의 목표이다.



Machine learning algorithm map

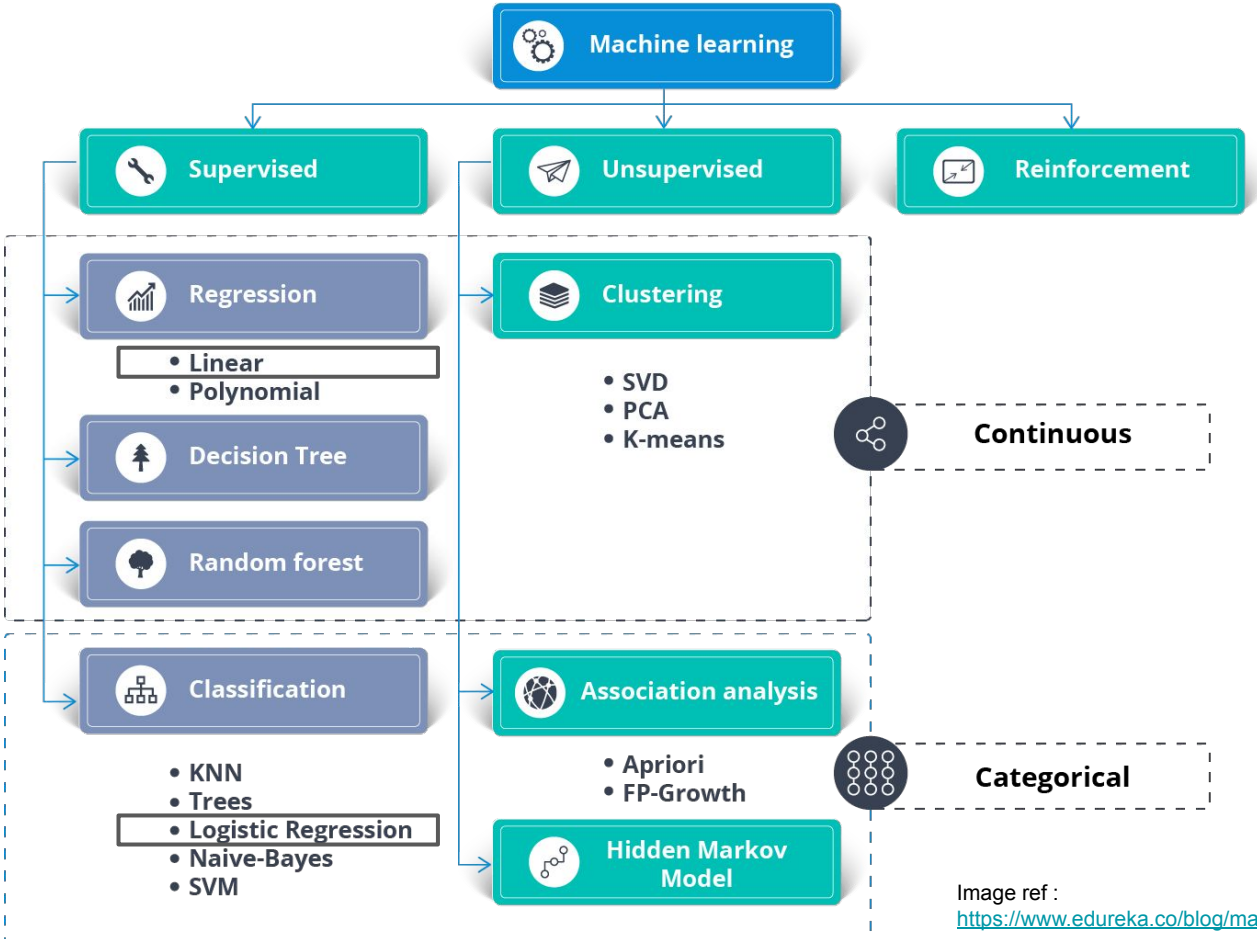


Image ref :
https://www.edureka.co/blog/machine-learning-algorithms/

모델과 학습

모델이란?

머신러닝에서 가장 간단한 정의로 모델을 설명하자면, 입력되는 데이터를 처리하는 함수를 의미한다.

일반적으로 프로그램은 입력 데이터를 처리하여 원하는 정보를 얻기 위한 절차를 약속된 명령어로 표현된 것이다. 머신러닝은 이러한 프로그램을 사람이 아닌 데이터로부터 학습이라는 방법으로 구현하는 AI의 한 방법론이다. 학습을 통해 얻어진 함수 형태의 프로그램이 모델이라 할 수 있다.

모델과 학습

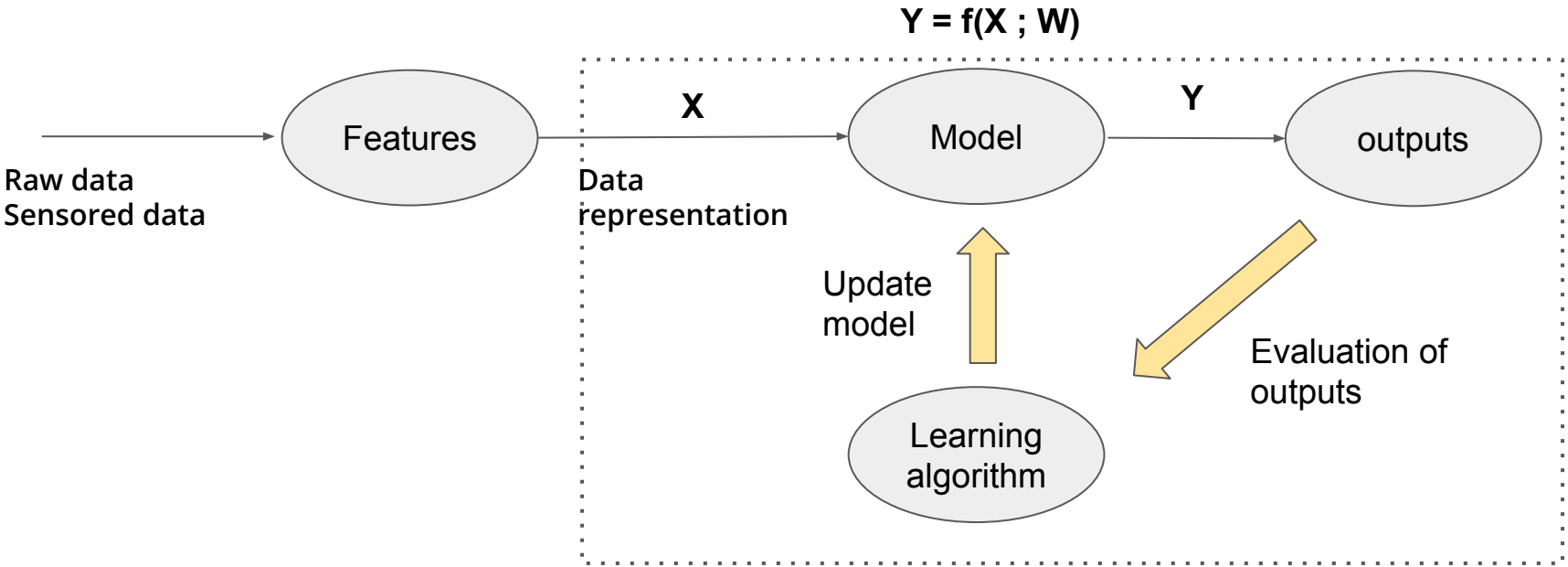
학습이란?

학습은 다루는 데이터에 대해 가장 적합한 모델을 찾는(fitting) 과정이다.

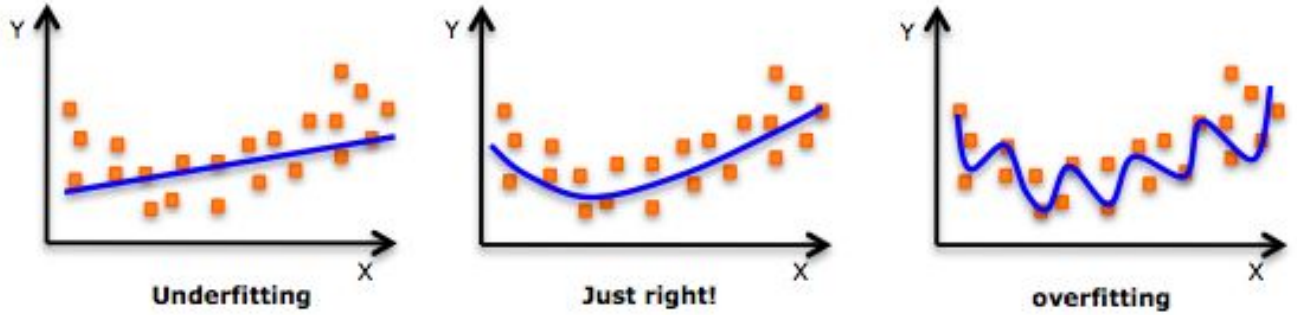
임의의 모델은 다양한 문제에 대해 적용할 수 있도록 파라미터를 갖는데 모델의 파라미터가 문제에 얼마나 적합한 지에 따라 분류, 예측에 대한 성능이 결정된다.

파라미터의 문제에 대한 적합성은 목적함수(Objective function)에 의해 정의되어 평가된다.
정량화된 수치로 평가하기 위한 것이 바로 비용함수(Loss or Cost or Error function)이다.

머신러닝의 모델 학습 개요

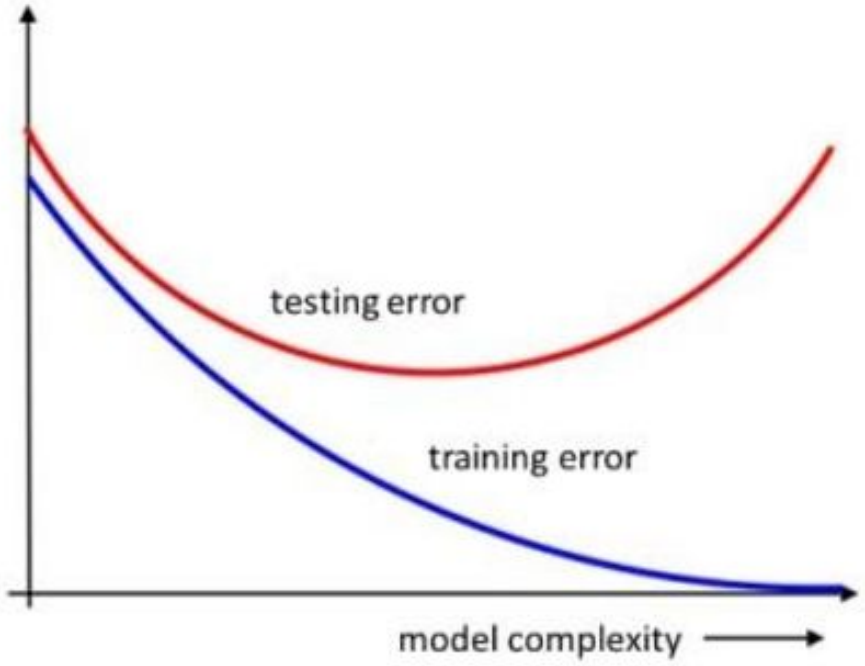


과적합과 일반화



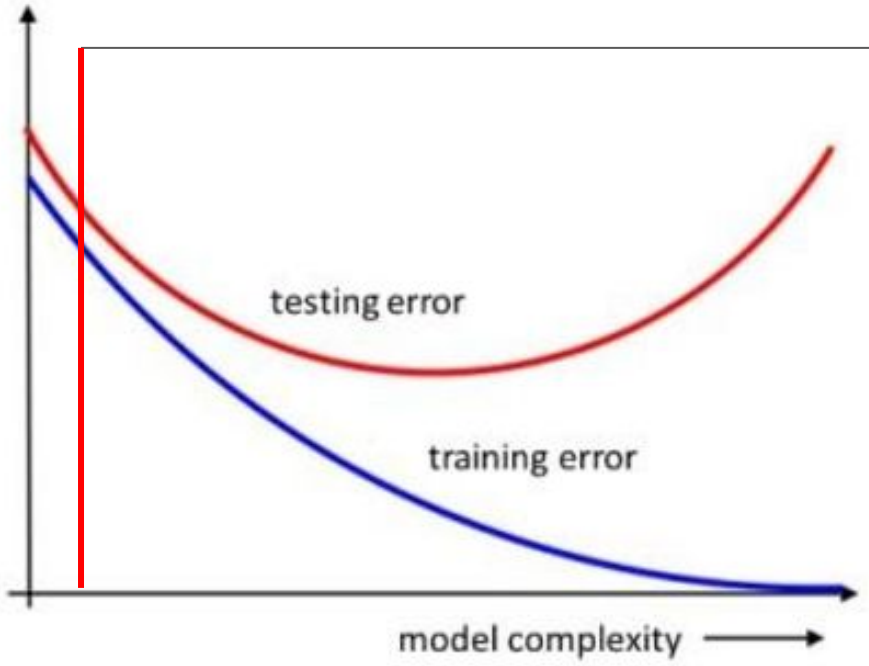
모델 복잡도 (model capacity or complexity) 증가

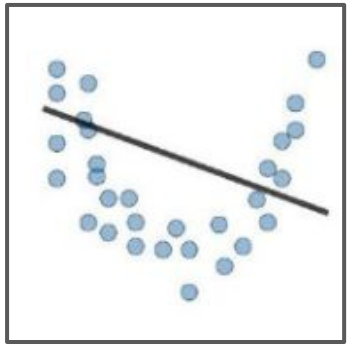
과적합과 일반화



적절한 수준의 복잡도를 가진 모델을
사용하는 것이 중요하다!

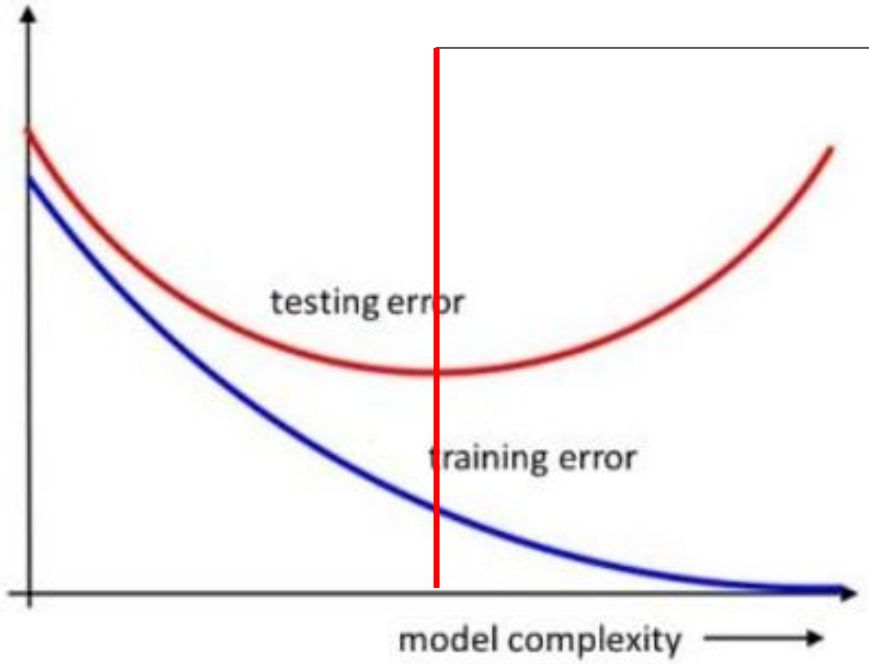
과적합과 일반화

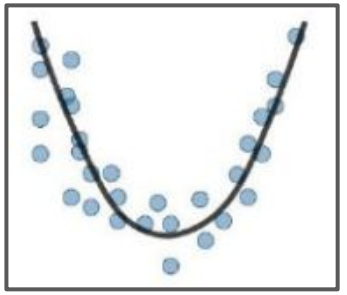




Underfitting!

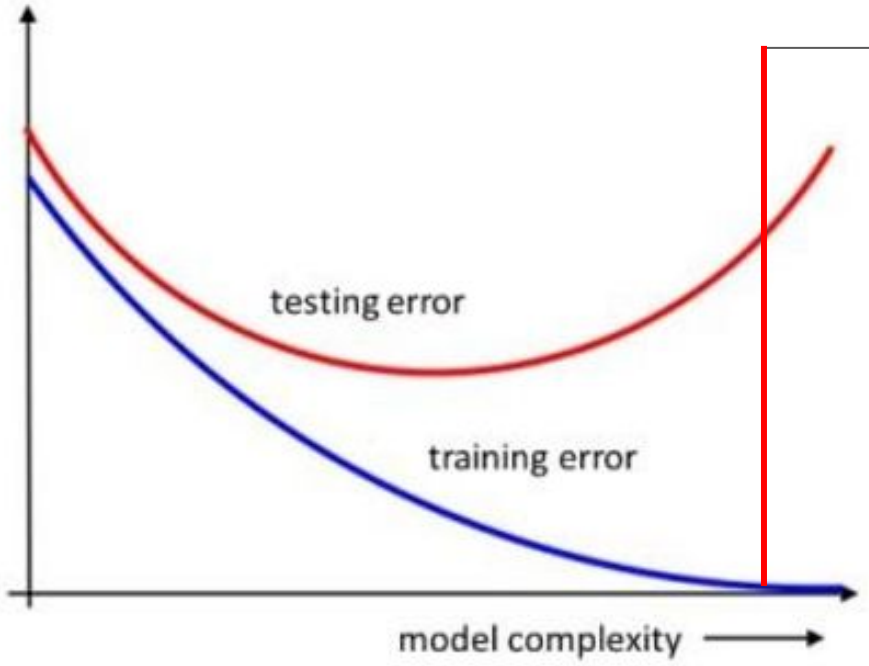
과적합과 일반화

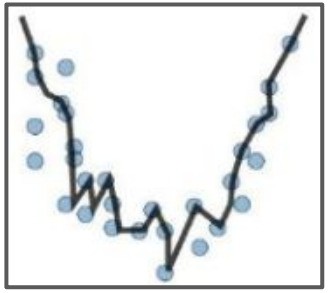




Good fitting!

과적합과 일반화

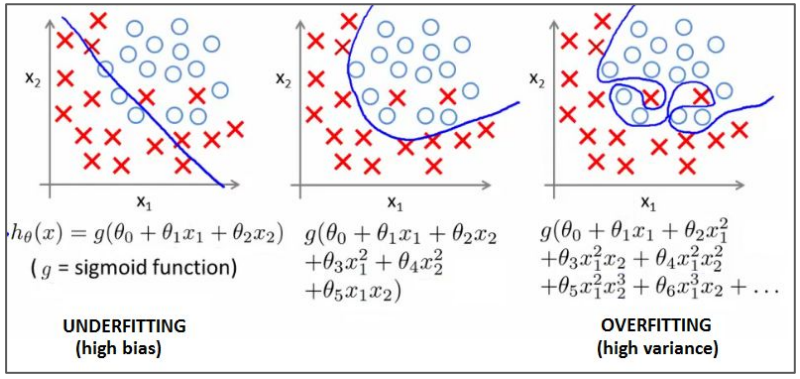




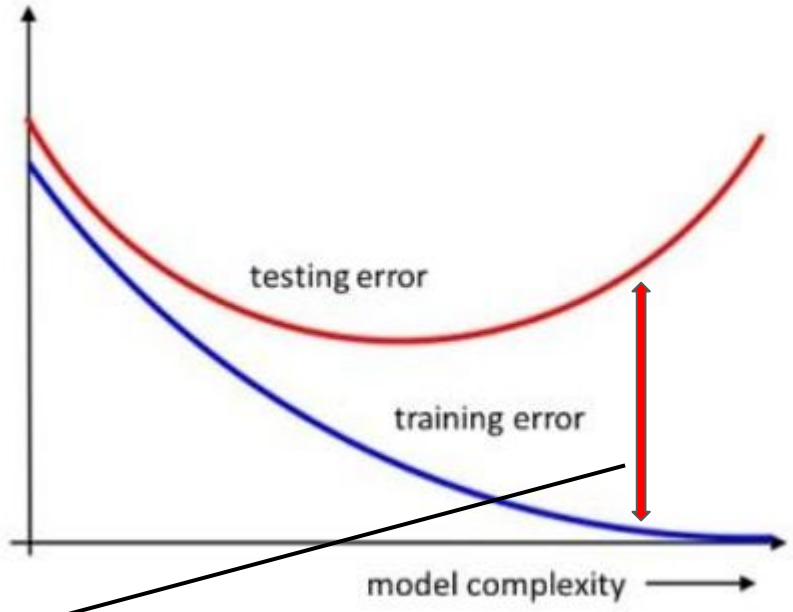
Overfitting!

Training error는 더욱 작아지지만, 새로운 데이터에 대한 성능인 Testing error는 증가하는 일반화 문제(Generalization error)가 크게 발생함!

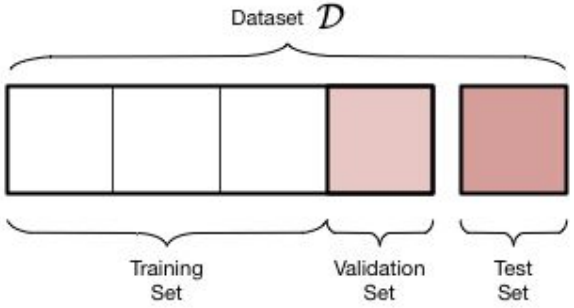
과적합과 일반화



https://doc.plob.org/machine_learning/07_Regularization.html



Generalization error! 적절한 수준의 Capacity를 갖는 모델을 사용하는 것이 필요!



일반화 오류를 줄이는 방법!

Training set으로는 모델의 Parameter를 학습하고!
Validation set으로는 모델의 Hyper-parameter를 튜닝하고!
Test set으로 모델의 최종 성능을 통해 모델을 선택(model selection)한다!

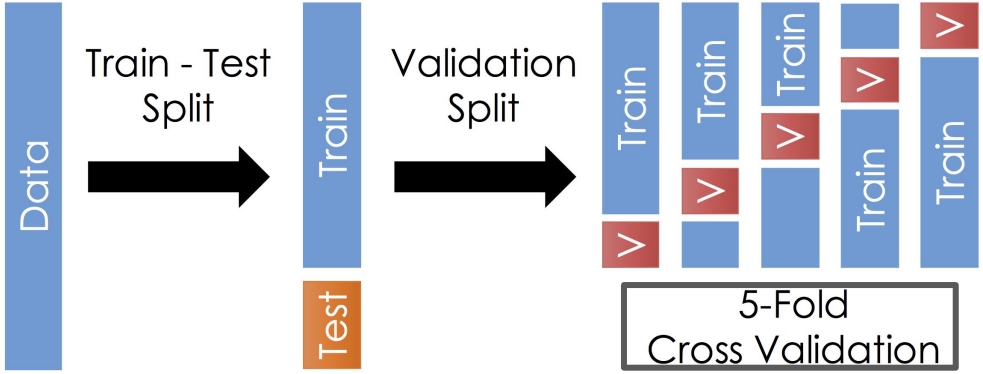


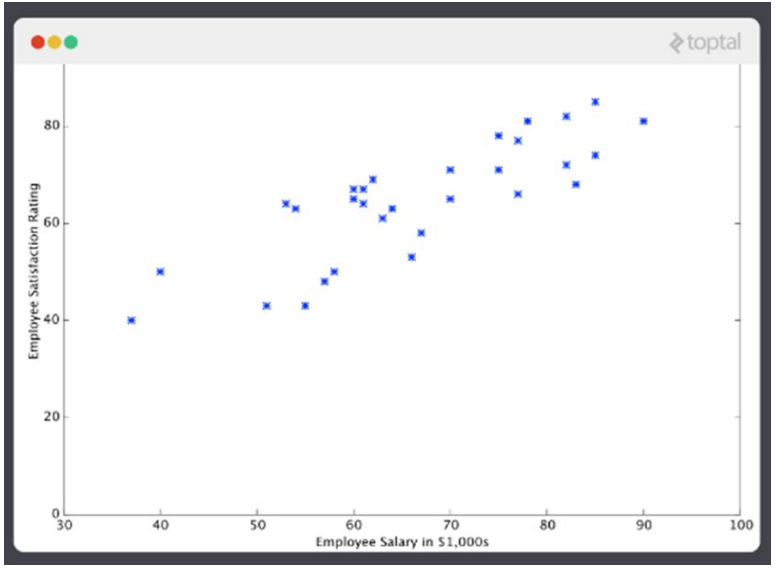
Image ref
http://www.ds100.org/sp17/assets/notebooks/linear_regression/Feature_Engineering_Part2.html

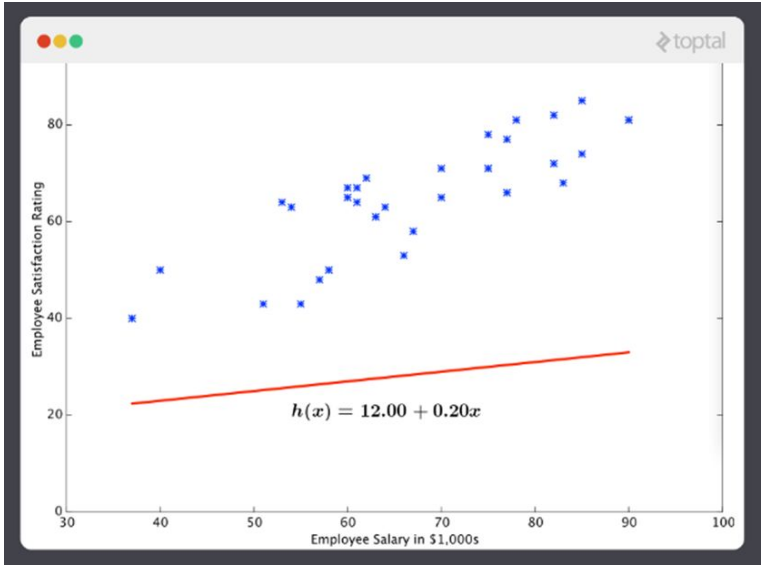
선형회귀 알고리즘

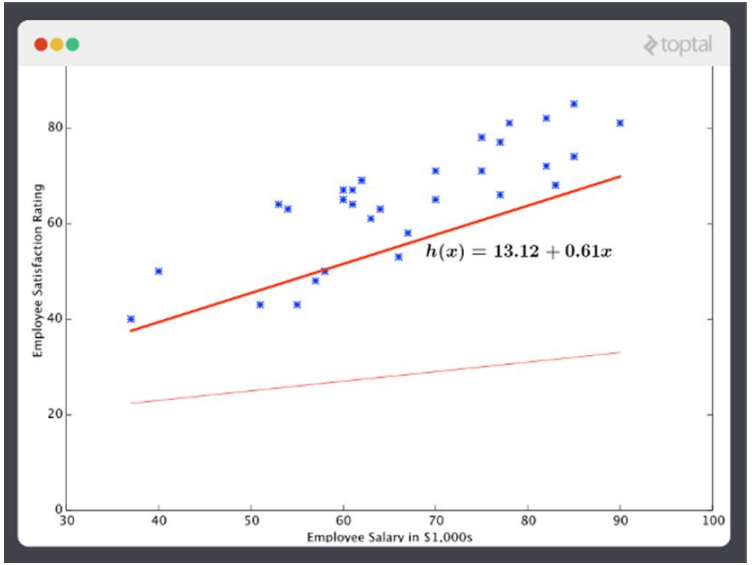
선형회귀(Linear regression)

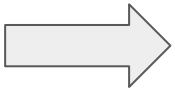
선형회귀(regression) 방법은 종속변수 y와 한 개 이상의 독립변수 x와의 상관 관계를 모델링하는 방법

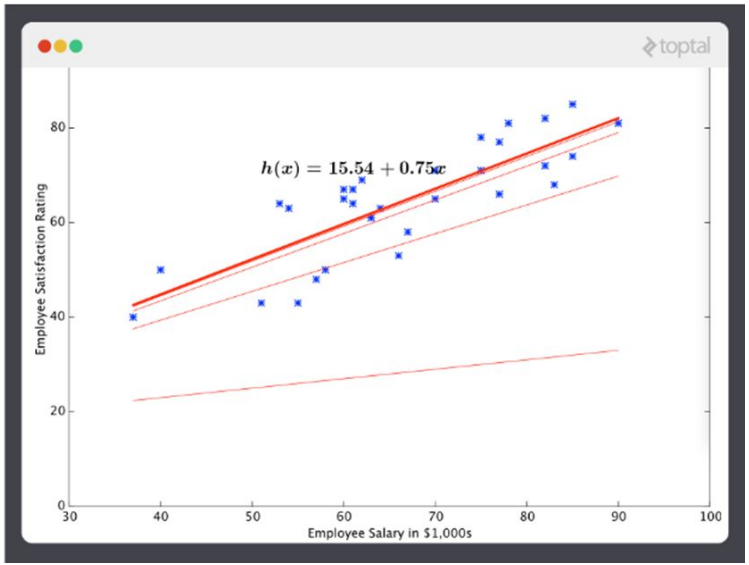
- Supervised learning
- Labeled dataset을 바탕으로 x를 통해 y를 예측 또는 설명
- The goal of ML is never to make “perfect” guesses, because ML deals in domains where there is no such thing. The goal is to make guesses that are good enough to be useful.











Complexity!

Consider a predictor that looks like this :

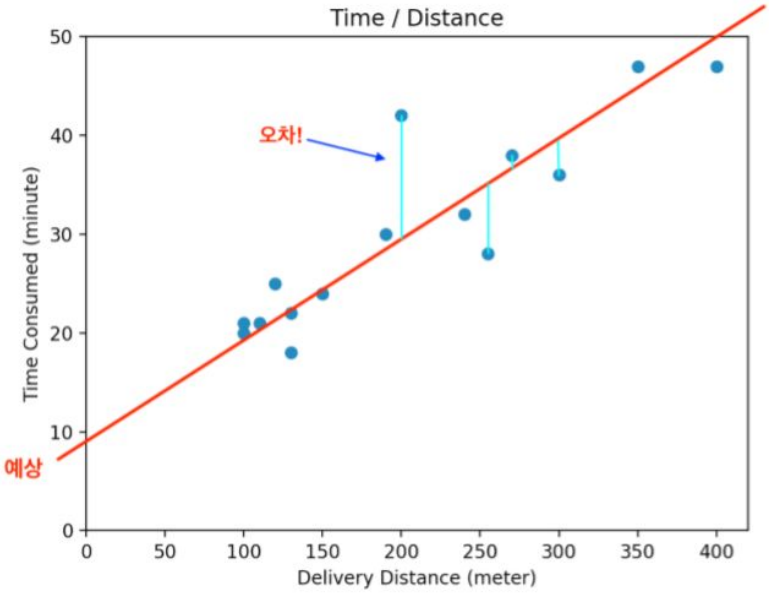
$$h(x_1, x_2, x_3, x_4) = \theta_0 + \theta_1 x_1 + \theta_2 x_3^2 + \theta_3 x_3 x_4 + \theta_4 x_1^3 x_2^2 + \theta_5 x_2 x_3^4 x_4^2$$

- **가설 (hypothesis)**

$$h_\theta(x) = \theta_0 + \theta_1 x \ (\theta_i : \text{Parameter})$$

- **비용 함수 (cost function)**

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^i) - y^i)^2$$



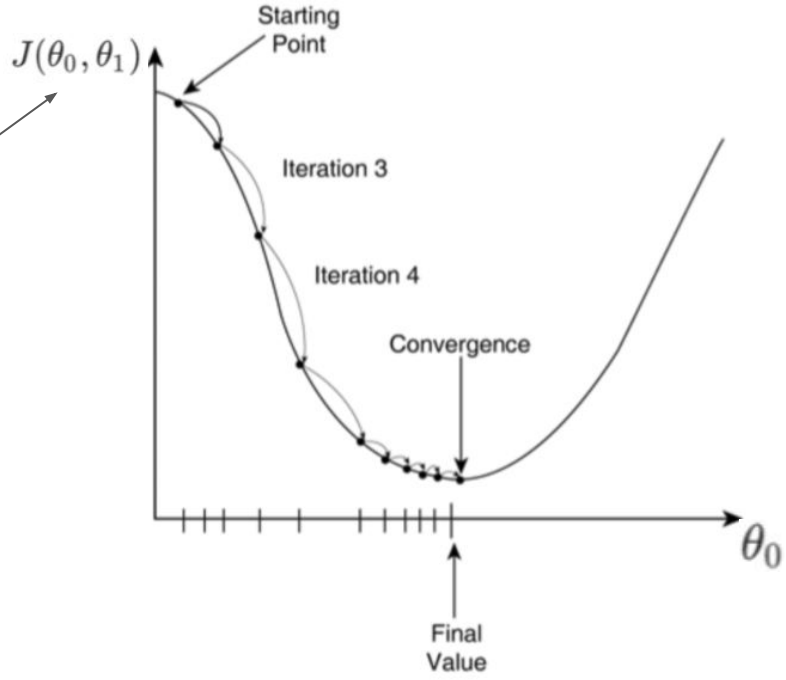
오차를 최소화하는 파라미터 (Θ) 를 찾는 것이 목적 !

= 데이터를 가장 잘 설명하는 모델을 찾는다 (model fitting)

경사하강법(Gradient descent)

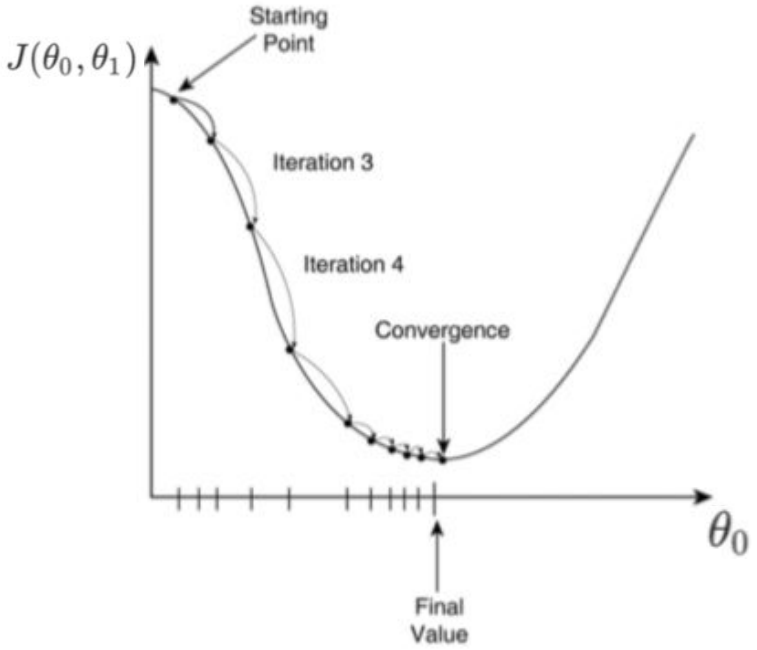
비용 함수를 최소화하기 위한 최적화 알고리즘
(기울기로 함수의 최소값을 찾는다)

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^i) - y^i)^2$$



$h_\theta(x) = \theta_0 + \theta_1 x$ (θ_i : Parameter)

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^i) - y^i)^2$$



1. 파라미터 초기화
2. 해당 파라미터에 대한 기울기 계산
3. 파라미터 업데이트
4. 비용 함수가 수렴할 때까지
(최솟값을 가질 때까지) 반복

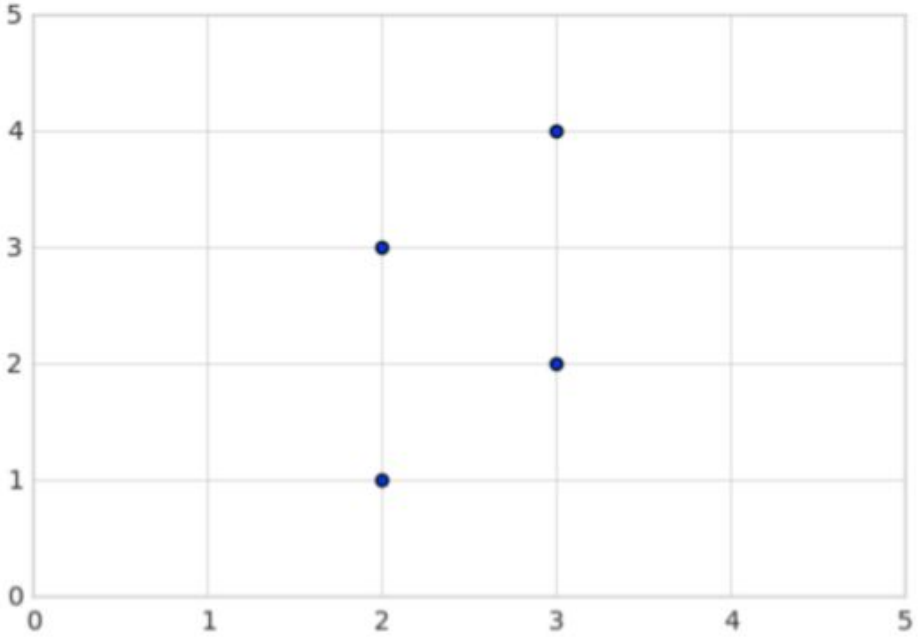
repeat until convergence {

$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$ (for j=0 and j=1)

}

α : learning rate(학습률)

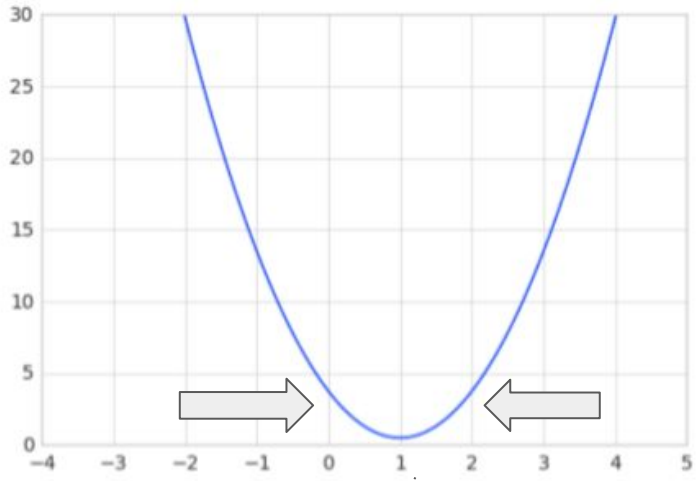
Gradient descent example



$$h_\theta(x) = \theta_1 x$$

$$J(\theta_1) = \frac{1}{2*4} \sum_{i=1}^{4} (h_\theta(x^i) - y^i)^2 = \frac{1}{8}(26\theta_1^2 - 52\theta_1 + 30)$$

Gradient descent example



$$J(\theta_1) = \frac{1}{8}(26\theta_1^2 - 52\theta_1 + 30)$$

θ_1 의 초깃값 = 0 , $\alpha = 0.1$

```
for i in range(38):
    theta = update_theta(theta, alpha)
    print (theta)
```

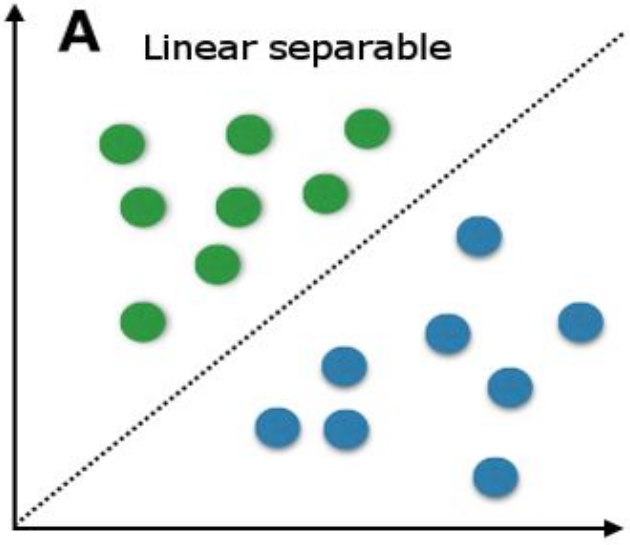
```
0.65
0.8775000000000001
0.957125
0.98499375
0.9947478125
0.998161734375
0.99935660703125
0.9997748124609375
0.9999211843613282
0.9999724145264649
0.9999903450842628
0.9999966207794919
0.9999988172728221
0.9999995860454878
0.9999998551159207
0.9999999492905722
0.9999999822517003
0.9999999937880951
0.9999999978258333
0.9999999992390416
0.9999999997336646
0.9999999999067826
0.9999999999673739
0.9999999999885809
0.9999999999960033
0.9999999999986011
0.9999999999995104
0.9999999999998286
0.99999999999994
0.999999999999979
0.9999999999999927
0.9999999999999974
0.9999999999999991
0.9999999999999997
0.9999999999999999
1.0
1.0
1.0
```

선형분류 알고리즘

로지스틱 회귀(Logistic regression)

로지스틱 회귀(regression) 방법은 이진 분류(binary classification) 문제를 해결하기 위한 알고리즘

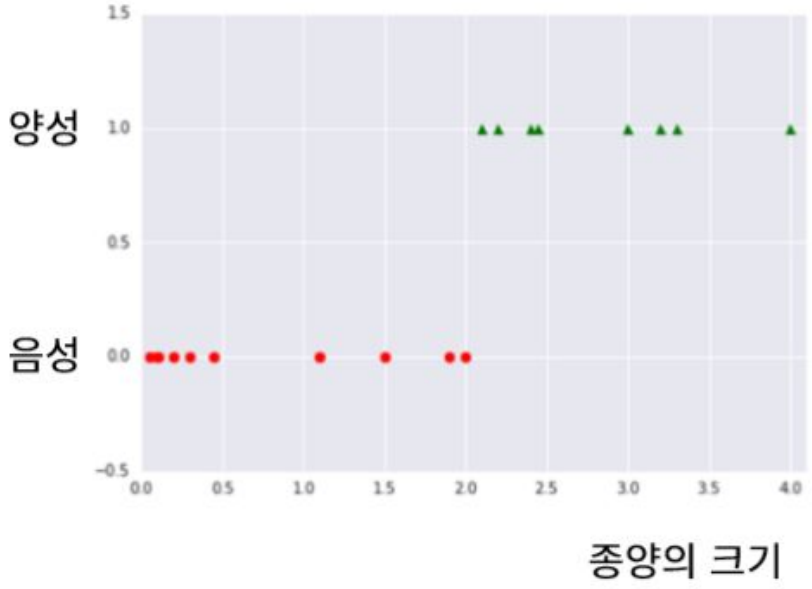
- Supervised learning
- Labeled dataset을 바탕으로 x를 통해 x의 클래스(y)를 분류

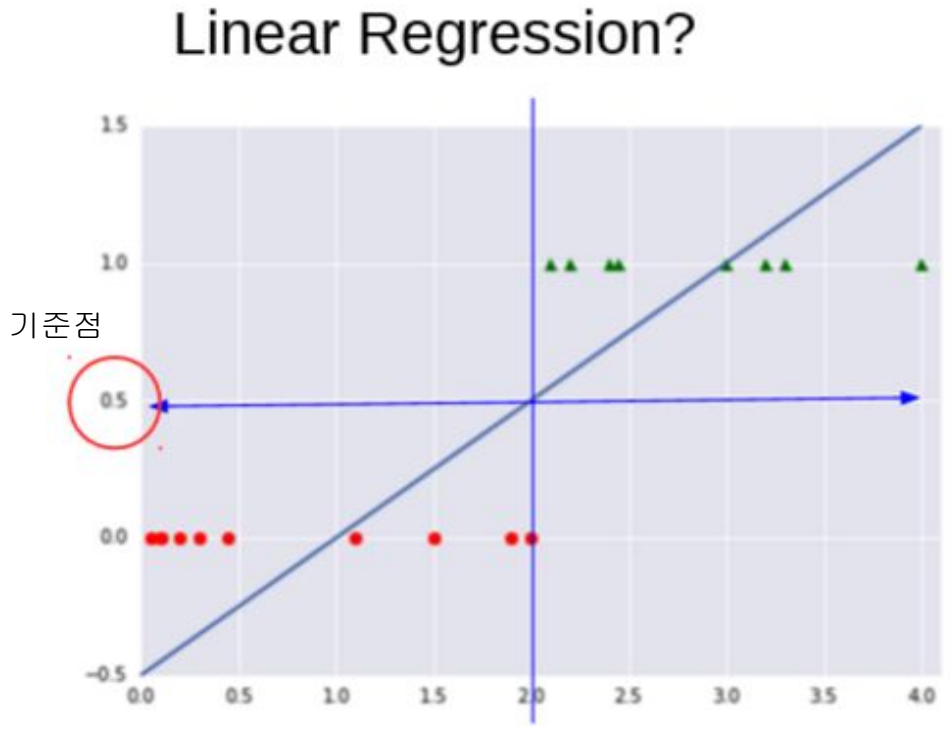


- 고객이 5 년 내에 사망할 확률 ?

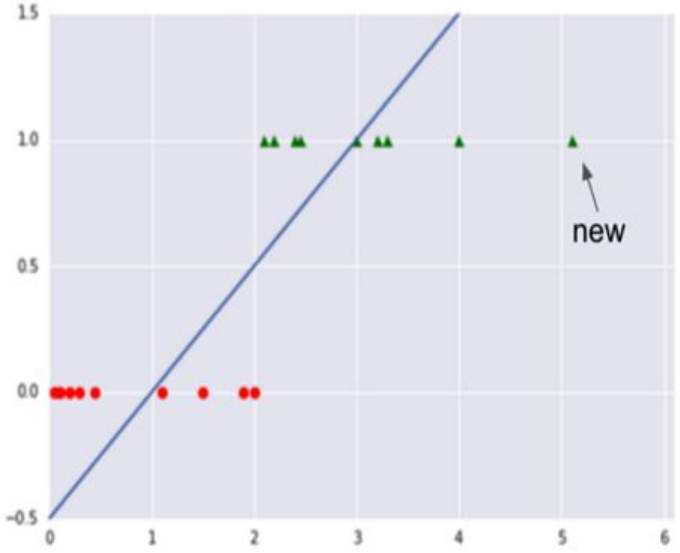
Sex	Age	Diabetes	...	Life span
Male	20	Yes	...	65
Female	40	Yes	...	68
Male	60	No	...	64

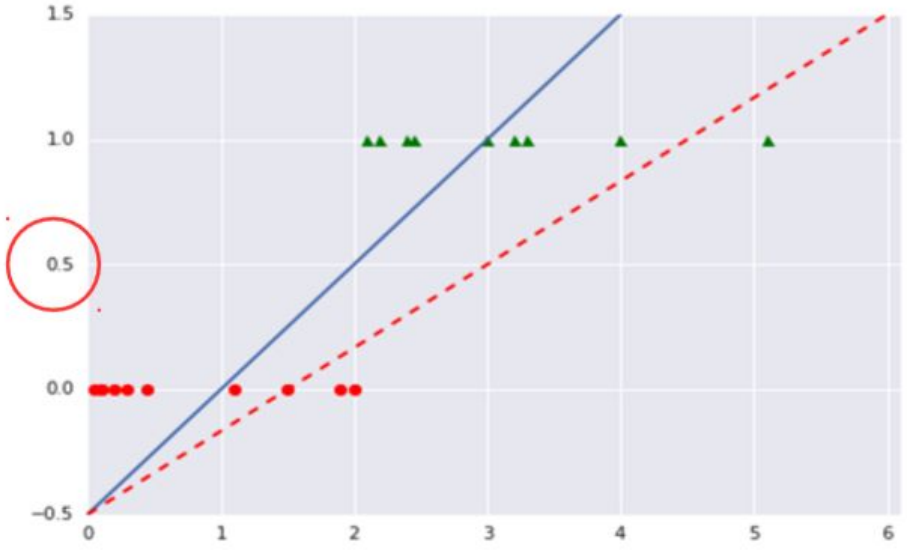
선형회귀 방식으로 이진분류를 하면 안 될까?





선형회귀 방식으로 이진분류를 하면 안 될까?

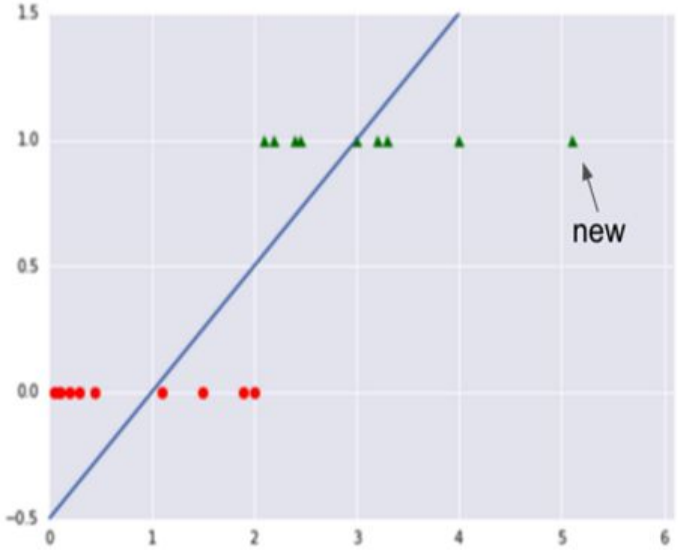


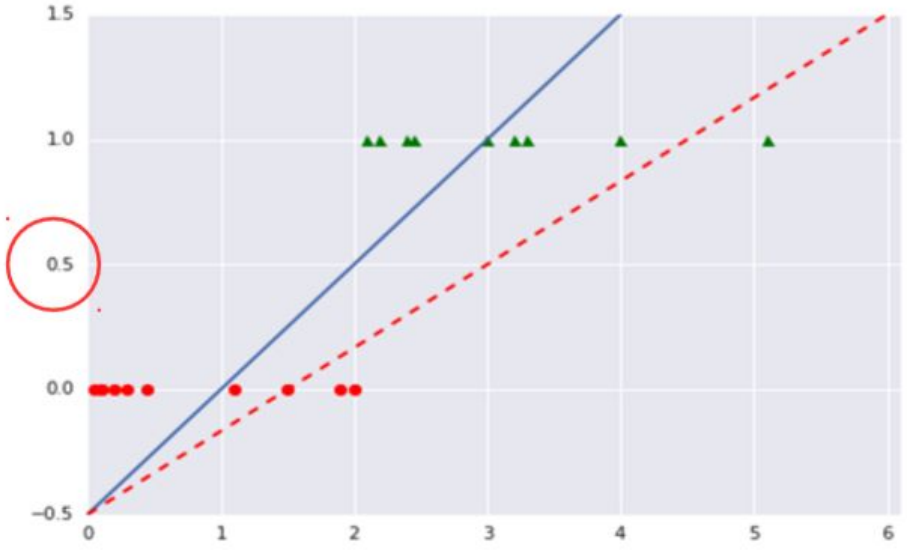


Linear regression for classification?

Problem 1) Threshold 2) Range

선형회귀 방식으로 이진분류를 하면 안 될까?





Linear regression for classification?

Problem

1) Threshold

기준점의 변화 : 기존의 0.5를 더 이상 해당 문제에 대한 분류 기준으로 쓸 수 없음.

2) Range

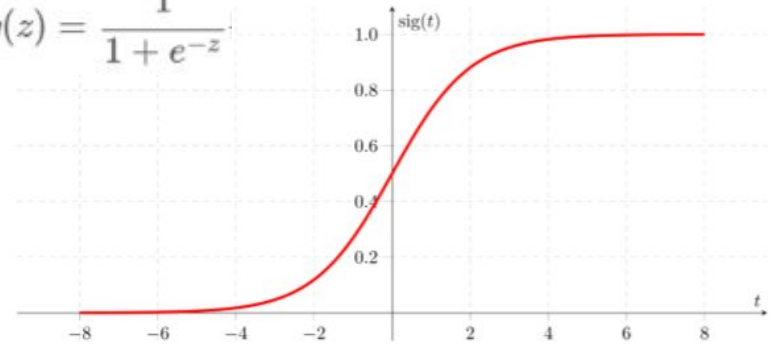
종양이 커지거나, 작을 수록 무한히 음성 또는 양성에 대한 점수가 무한히 커질까? 확률로 나타내자!

시그모이드 함수(Sigmoid function)의 사용

선형회귀 모델의 예측값을 시그모이드 함수에 넣으면 범위를 0~1 사이의 값으로 한정할 수 있다.

이는 확률의 범위와 같으므로 분류 문제의 예측 결과로도 적합

$$g(z) = \frac{1}{1+e^{-z}}$$



t = a*X + b
h = sigmoid (t)

Q. 양성 VS 음성 ?

$h = p(Y = 양성 | X = 데이터)$

If h > 0.5 → 양성
If h < 0.5 → 음성

↓

t > 0 → 양성 (y=1)
t < 0 → 음성 (y=0)

Cost function의 정의

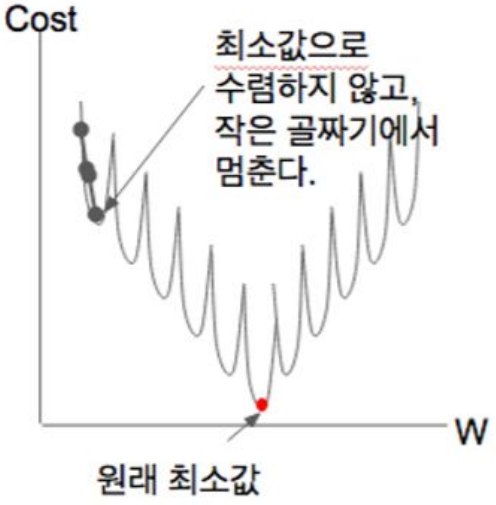
- **가설 (hypothesis)**

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

- **비용 함수 (cost function)**

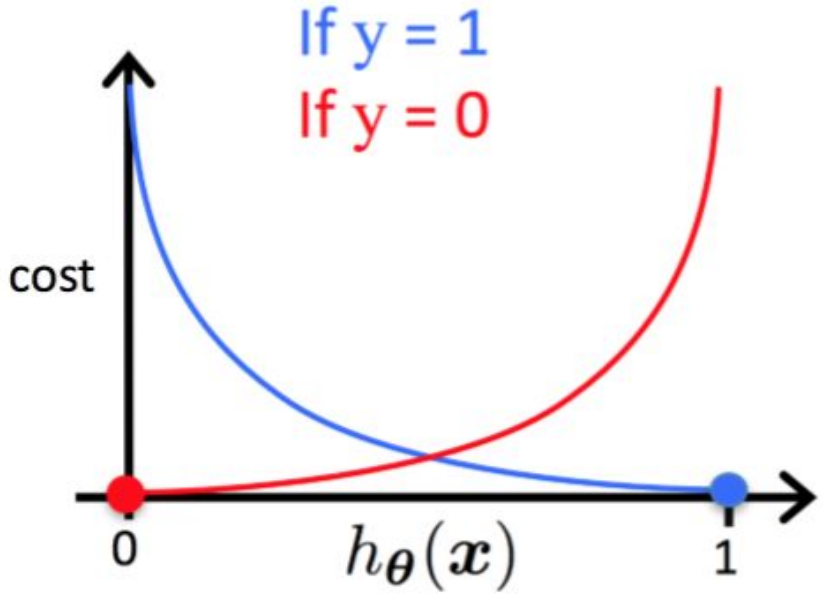
$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$$

새로운 비용 함수 필요



- **비용 함수 (cost function)**

$$\text{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$



- **비용 함수 (cost function)**

$$\text{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

$$-\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log (1 - h_\theta(x^{(i)}))]$$

비용 함수 (오차) 를 최소화 하는 파라미터 (θ) 를 찾는다 .

→ 경사 하강법 (Gradient descent)

Logistic regression의 학습

$$J(\boldsymbol{\theta}) = \frac{1}{m}\sum_{i=1}^{m} \text{Cost}(h_\theta(x), y)$$

$$\text{Cost}(h_\theta(\mathbf{x}), y) = -y log(h_\theta(\mathbf{x})) - (1-y) log(1 - h_\theta(\mathbf{x}))$$

Gradient descent for logistic regression:

while not converged {

$$\theta_j^{\text{new}} = \theta_j^{\text{old}} - \alpha \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)} \text{ for } j = 0, 1, \ldots, n$$

}

$$output = \underbrace{p(Class_{(+)}|X)}_{=y} = \frac{1}{1+e^{-W^T X}}$$

θ

If $W^T X > 0$,

then $p(Class_{(+)}|X) > 0.5$ and positive class

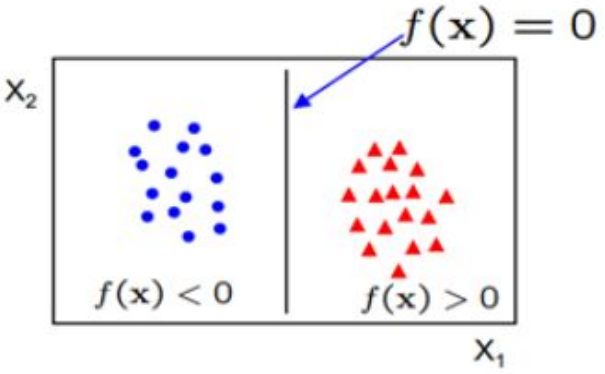
θ

If $W^T X < 0$,

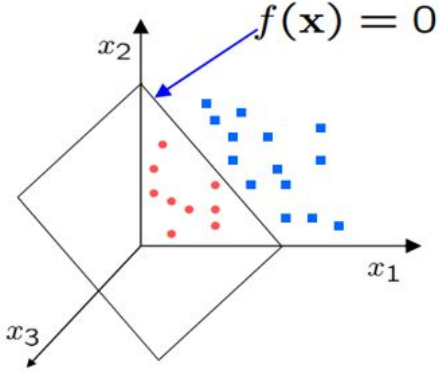
then $p(Class_{(+)}|X) < 0.5$ and negative class

이것이 갖는 기하학적 의미?

$$f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$$



(1)데이터가 2차원일 때



(2)데이터가 3차원일 때

$$h_\theta(x^{(i)}) = \frac{1}{1+e^{-(\theta^T x^{(i)}+b)}}$$

$$J(\theta) = -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} \log(h_\theta(x^{(i)})) + (1-y^{(i)}) \log(1-h_\theta(x^{(i)}))]$$

$$= \frac{1}{m}[\sum_{i=1}^{m} -y^{(i)} \log(\frac{1}{1+e^{-(\theta^T x^{(i)}+b)}}) - (1-y^{(i)}) \log(1-\frac{1}{1+e^{-(\theta^T x^{(i)}+b)}})]$$

$$= \frac{1}{m}[\sum_{i=1}^{m} -y^{(i)} (\log 1 - log(1+e^{-(\theta^T x^{(i)}+b)})) - (1-y^{(i)}) \log(\frac{e^{-(\theta^T x^{(i)}+b)}}{1+e^{-(\theta^T x^{(i)}+b)}})]$$

$$= \frac{1}{m}[\sum_{i=1}^{m} y^{(i)} log(1+e^{-(\theta^T x^{(i)}+b)}) - (1-y^{(i)}) \log(e^{-(\theta^T x^{(i)}+b)}) + (1-y^{(i)}) \log(1+e^{-(\theta^T x^{(i)}+b)})]$$

$$= \frac{1}{m}[\sum_{i=1}^{m} (1-y^{(i)})(\theta^T x^{(i)} + b) + \log(1+e^{-(\theta^T x^{(i)}+b)})]$$

$$= \frac{1}{m}[\sum_{i=1}^{m} \log e^{\theta^T x^{(i)}+b} - y^{(i)}(\theta^T x^{(i)} + b) + \log(1+e^{-(\theta^T x^{(i)}+b)})]$$

$$= \frac{1}{m}[\sum_{i=1}^{m} \log(e^{\theta^T x^{(i)}+b} + 1) - y^{(i)}(\theta^T x^{(i)} + b)]$$

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m}[\sum_{i=1}^{m} \frac{x_j e^{\theta^T x^{(i)}+b}}{e^{\theta^T x^{(i)}+b}+1} - y^{(i)} x_j] = \frac{1}{m} x_j \sum_{i=1}^{m} [\frac{1}{1+e^{-(\theta^T x^{(i)}+b)}} - y^{(i)}] = \frac{1}{m} x_j \sum_{i=1}^{m} [h_\theta(x^{(i)}) - y^{(i)}]$$

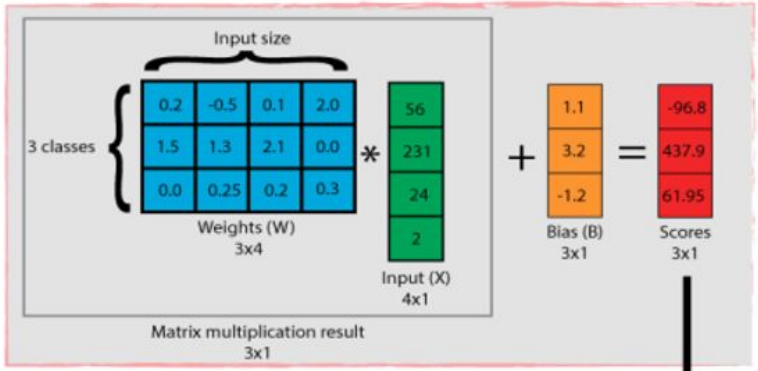
$$\frac{\partial J(\theta)}{\partial b} = \frac{1}{m}[\sum_{i=1}^{m} \frac{e^{\theta^T x^{(i)}+b}}{e^{\theta^T x^{(i)}+b}+1} - y^{(i)}] = \frac{1}{m} \sum_{i=1}^{m} [\frac{1}{1+e^{-(\theta^T x^{(i)}+b)}} - y^{(i)}] = \frac{1}{m} \sum_{i=1}^{m} [h_\theta(x^{(i)}) - y^{(i)}]$$

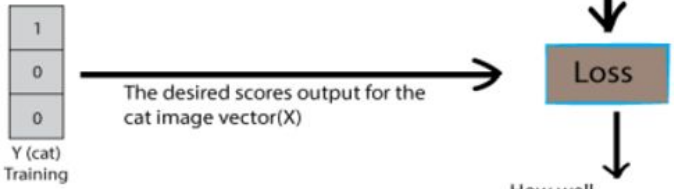
다중 클래스 분류기(Multi-class classifier)

클래스가 여러 개 일때는 어떻게 할까?

Idea transform a vector of numbers into scores for each possible class.
On this case we transform the matrix of numbers (image) on a 1d vector (All spatial information is lost)
On this example our image became an array 4x1, and we need to classify images as (cat,dog,ship)

Linear model f(x,W,b)





How well the parameters W, B map the input to the desired output

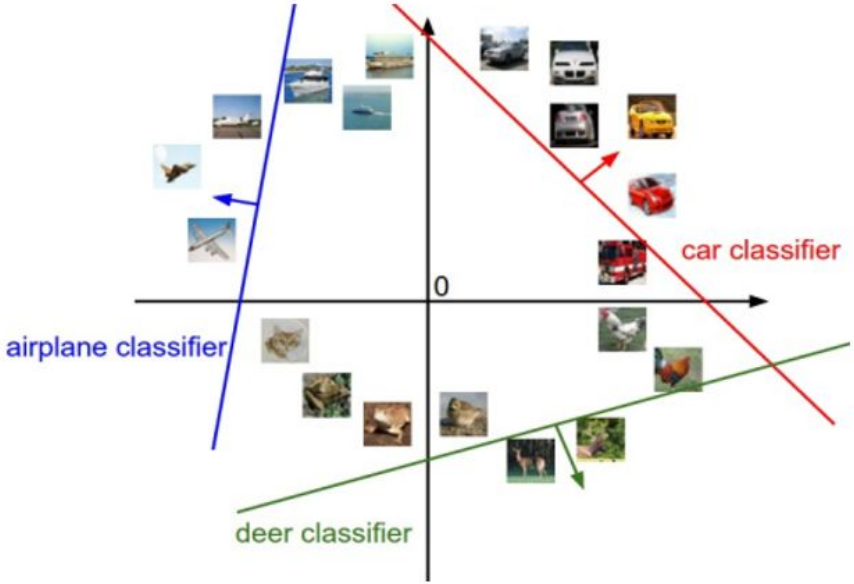
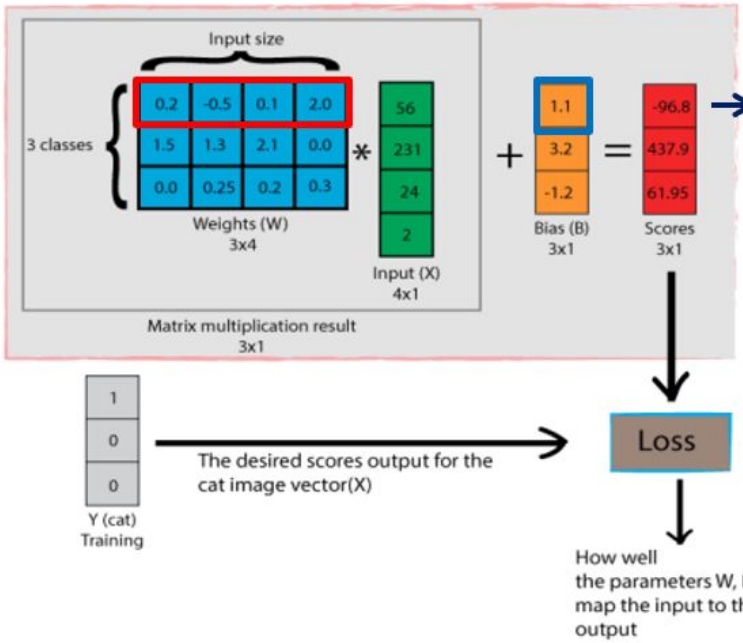


Image ref : stanford cs 231n(http://cs231n.stanford.edu/)

Idea transform a vector of numbers into scores for each possible class.
On this case we transform the matrix of numbers (image) on a 1d vector (All spatial information is lost)
On this example our image became an array 4x1, and we need to classify images as (cat,dog,ship)

Linear model f(x,W,b)



$Z = W^T X + b$

First one(airplane)
among three classifiers

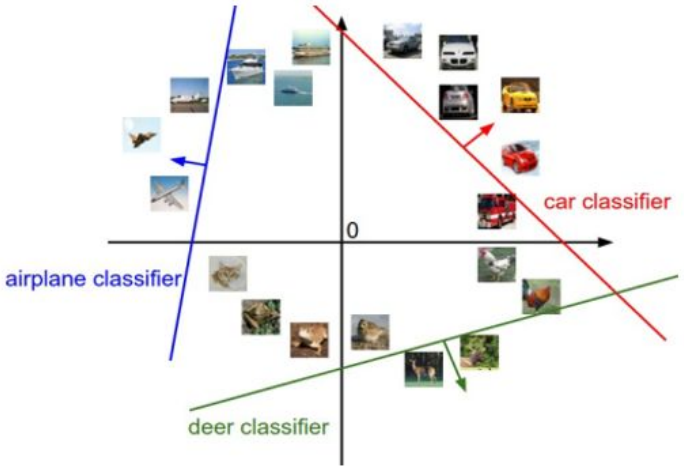
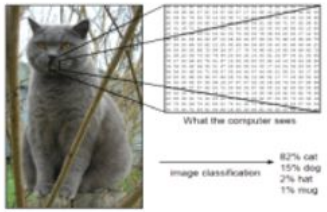
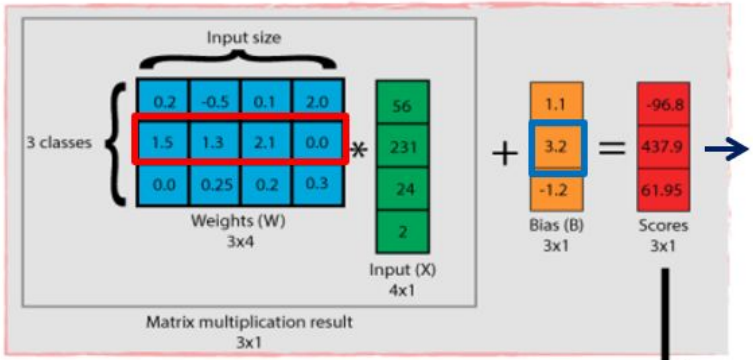


Image ref : stanford cs 231n(http://cs231n.stanford.edu/)



Idea transform a vector of numbers into scores for each possible class.
On this case we transform the matrix of numbers (image) on a 1d vector (All spatial information is lost)
On this example our image became an array 4x1, and we need to classify images as (cat,dog,ship)

Linear model f(x,W,b)



$Z = W^T X + b$

The desired scores output for the cat image vector(X)

Loss

How well the parameters W, B map the input to the desired output

Second one(Deer) among three classifiers

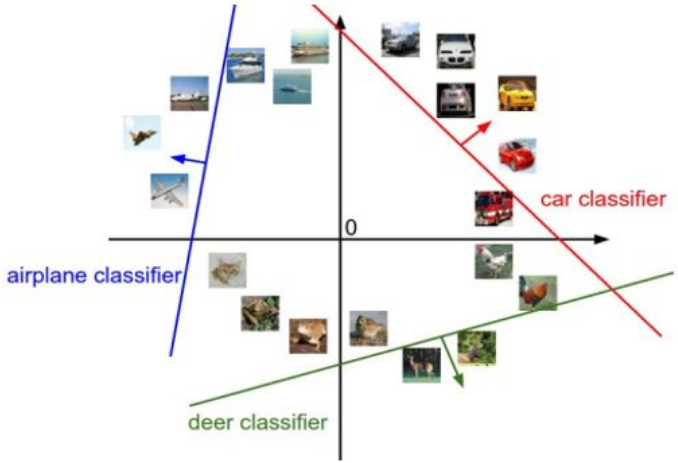
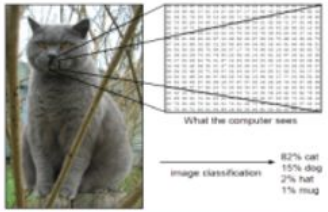
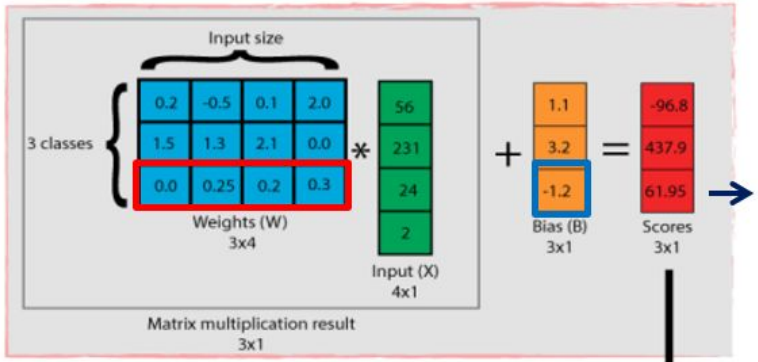


Image ref : stanford cs 231n(http://cs231n.stanford.edu/)



Idea transform a vector of numbers into scores for each possible class.
On this case we transform the matrix of numbers (image) on a 1d vector (All spatial information is lost)
On this example our image became an array 4x1, and we need to classify images as (cat,dog,ship)

Linear model f(x,W,b)



$Z = W^T X + b$

Y (cat)
Training

The desired scores output for the cat image vector(X)

Loss

How well
the parameters W, B
map the input to the desired output

Third one(Car)
among three classifiers

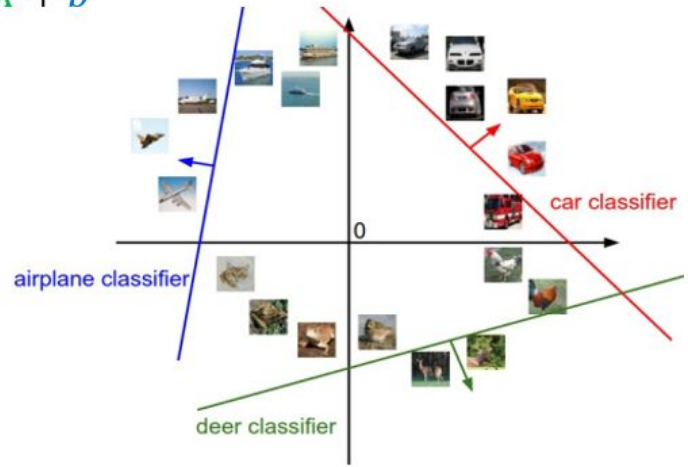
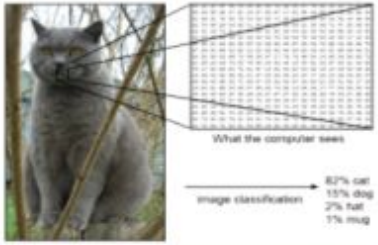
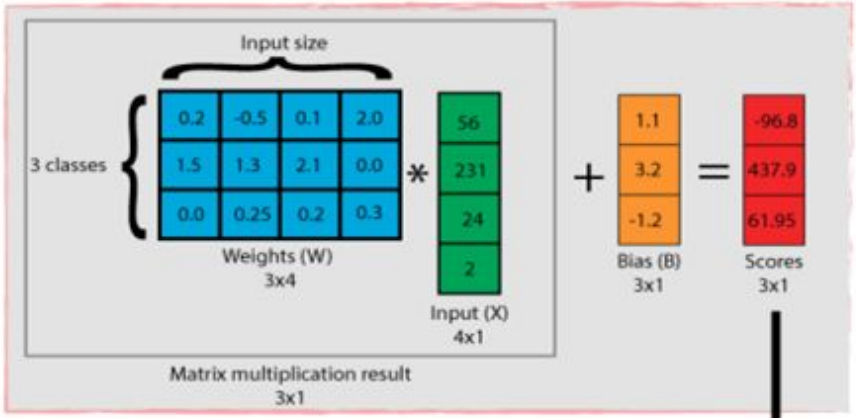


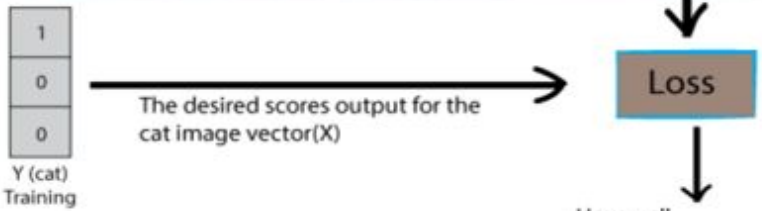
Image ref : stanford cs 231n(http://cs231n.stanford.edu/)



Idea transform a vector of numbers into scores for each possible class.
On this case we transform the matrix of numbers (image) on a 1d vector (All spatial information is lost)
On this example our image became an array 4x1, and we need to classify images as (cat,dog,ship)

Linear model f(x,W,b)





How well the parameters W, B map the input to the desired output

아래 영역들에 대한 분류기의 스코어 값은 어떨까?

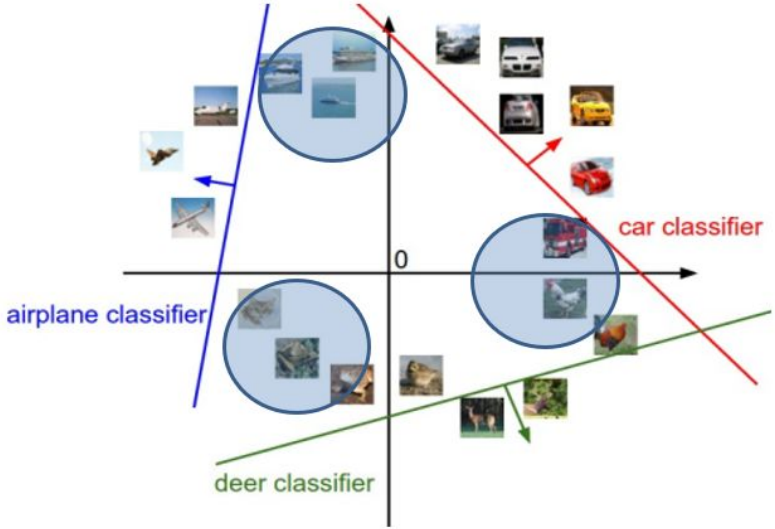
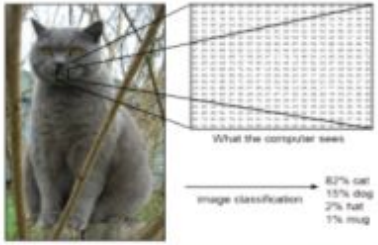
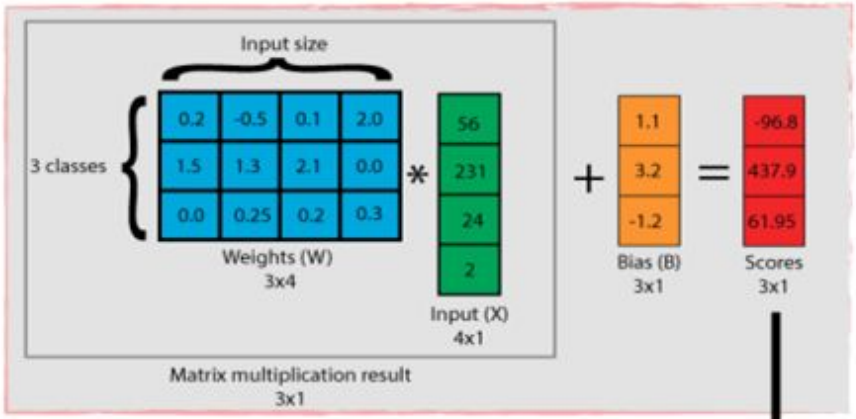


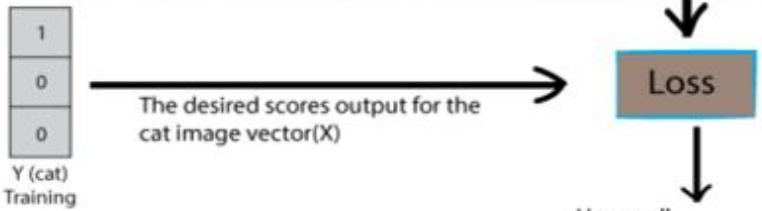
Image ref : stanford cs 231n(http://cs231n.stanford.edu/)



Idea transform a vector of numbers into scores for each possible class.
On this case we transform the matrix of numbers (image) on a 1d vector (All spatial information is lost)
On this example our image became an array 4x1, and we need to classify images as (cat,dog,ship)

Linear model f(x,W,b)





아래 영역들에 대한 분류기의 스코어 값은 어떨까?
이상적으로 Negative

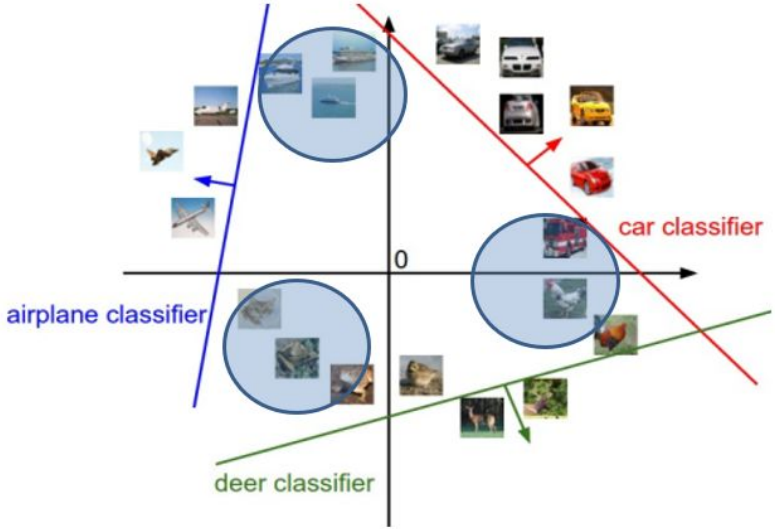
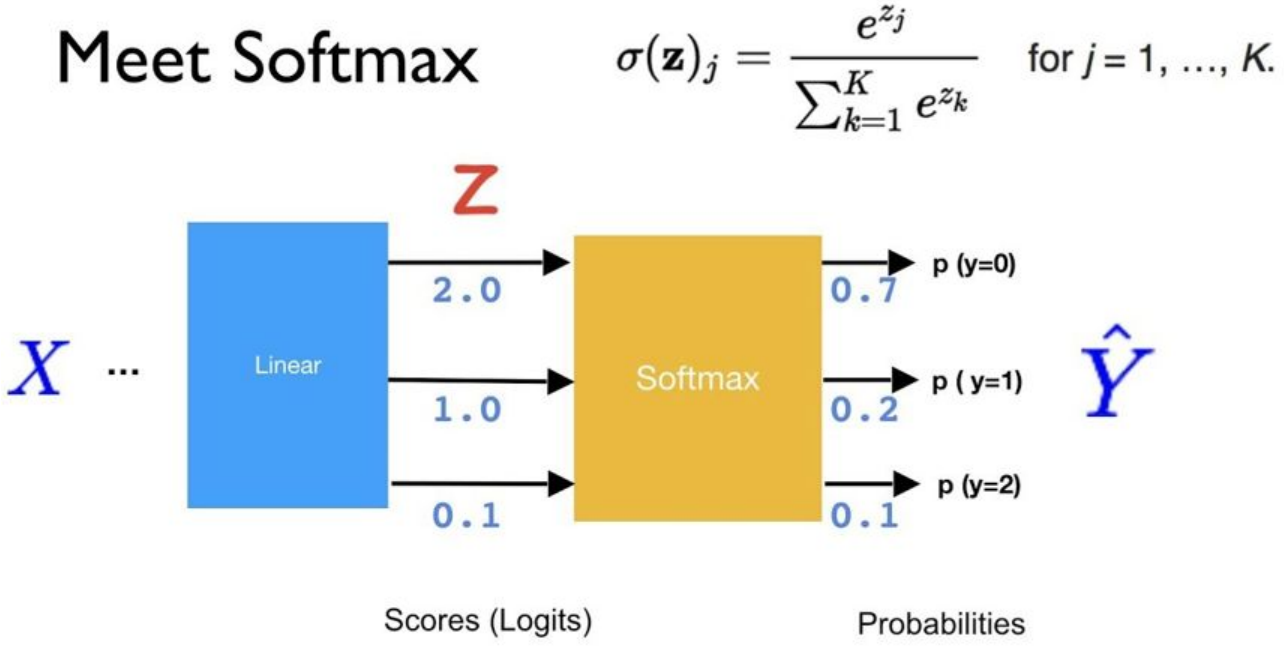


Image ref : stanford cs 231n(http://cs231n.stanford.edu/)

소프트맥스(Softmax)와 크로스엔트로피(Cross-entropy)

Softmax function

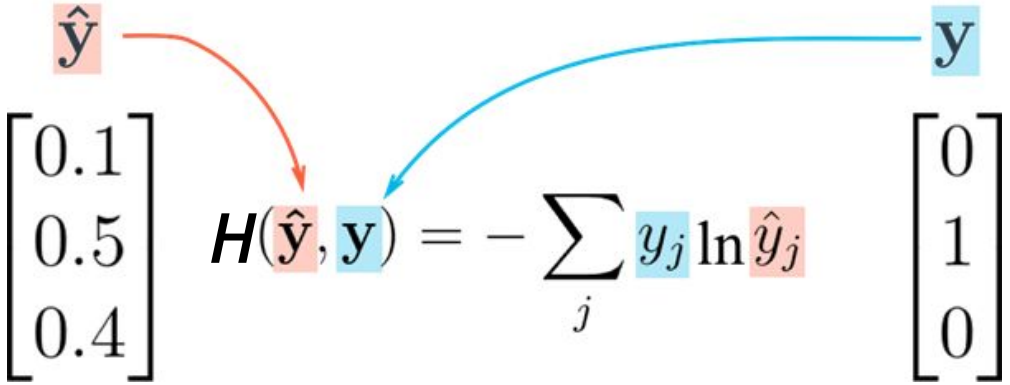


소프트맥스(Softmax)와 크로스엔트로피(Cross-entropy)

Cross entropy

$$H(p,q) = -\sum_{x} p(x)\,\log q(x).$$

모델이 예측한 데이터에 대한 각 클래스별 확률



실제 정답 : 정답 레이블을 **one-hot coding**한 형태 **(**확률 분포의 형태로 만들어주기 위해 사용**)**

소프트맥스(Softmax)와 크로스엔트로피(Cross-entropy)

왜 Softmax function를 쓸까? 이름은 왜 Softmax일까?

An example. Say $K = 4$, and your log score x is vector $[2, 4, 2, 1]$. The simple argmax function outputs:

$$[0, 1, 0, 0]$$

The argmax is the goal, but it's not differentiable and we can't train our model with it :(A simple normalization, which is differentiable, outputs the following probabilities:

$$[0.2222, 0.4444, 0.2222, 0.1111]$$

That's really far from the argmax! :(Whereas the softmax outputs:

$$[0.1025, 0.7573, 0.1025, 0.0377]$$

That's much closer to the argmax! Because we use the natural exponential, we hugely increase the probability of the biggest score and decrease the probability of the lower scores when compared with standard normalization. Hence the "max" in softmax.

Thank you!