

05. MACHINE LEARNING

☞ Machine Learning의 의미

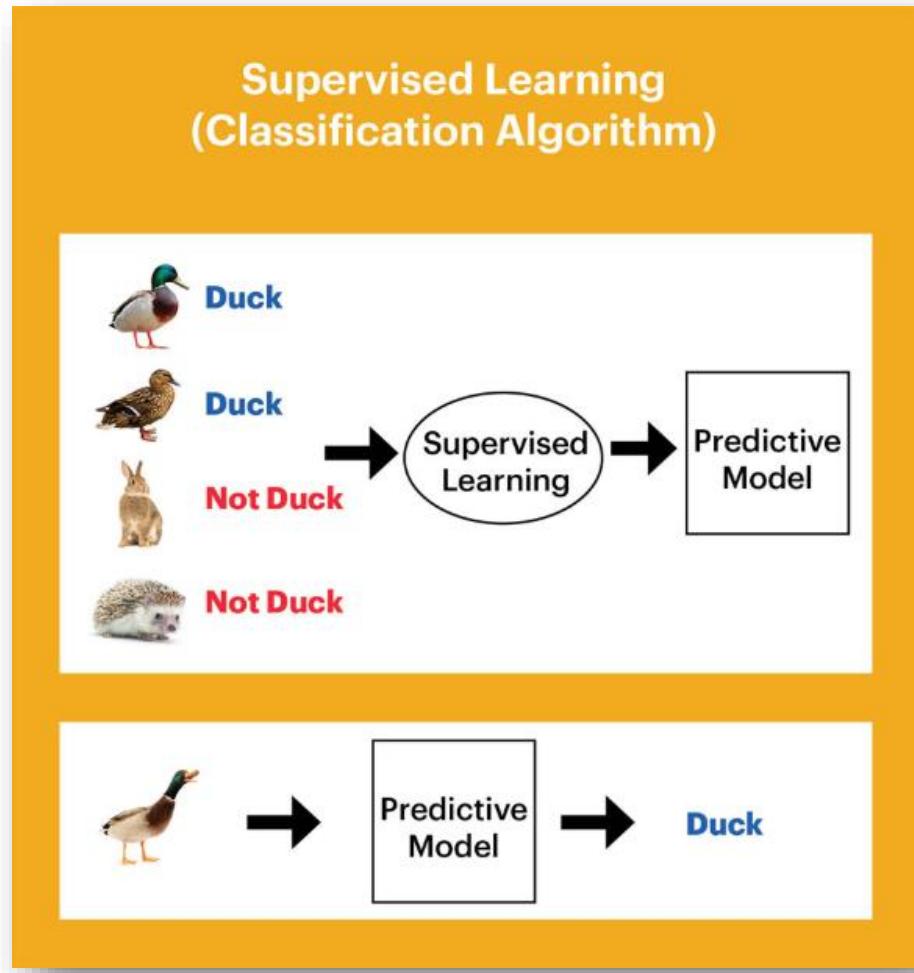
- Machine Learning은 Software.
- Explicit programming의 한계때문에 주목을 받음.
 - Explicit Programming은 우리가 일반적으로 하는 프로그래밍
 - Explicit Programming으로 해결할 수 없는 문제들이 있음. (경우의 수가 너무 많음)
 - 대표적인 것들로 Email Spam Filter, 자동 운전 시스템, 바둑 프로그래밍 등.
- 1959년 Arthur Samuel에 의해서 시작.
 - Machine Learning : 프로그램 자체가 데이터를 기반으로 학습을 통해 배우는 능력을 가지는 프로그래밍

05. MACHINE LEARNING

☞ Learning의 종류(1)

● Supervised Learning

- training set이라고 불리는 label화 된 데이터를 통해 학습.

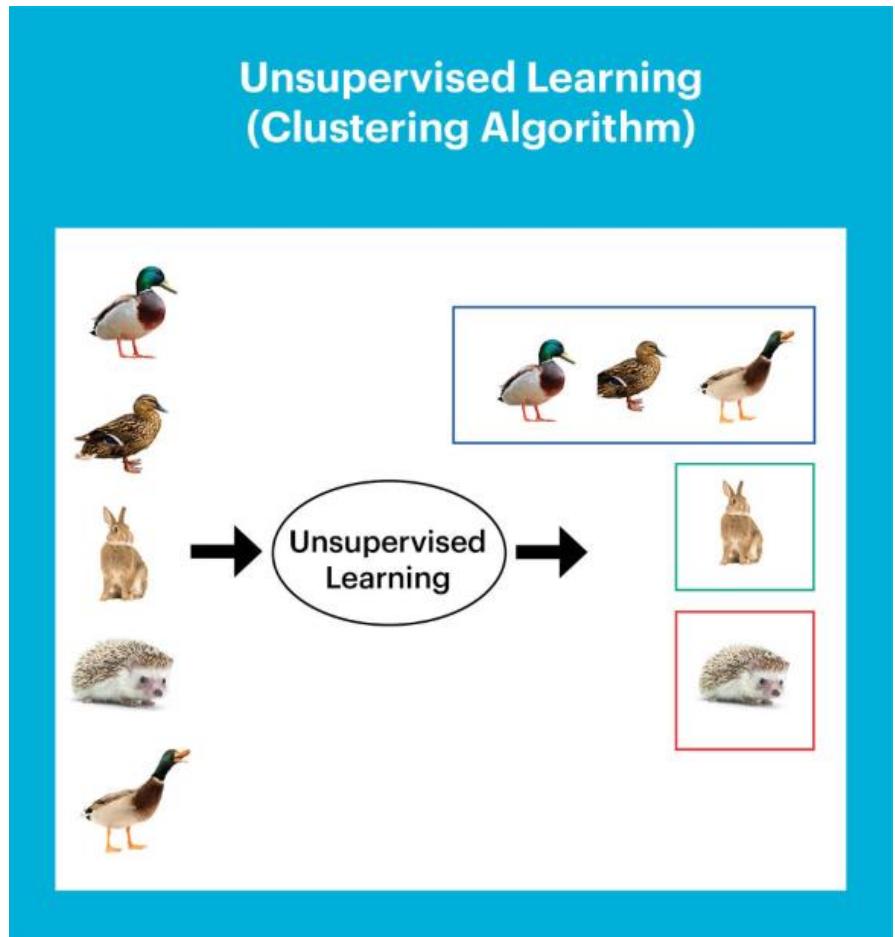


05. MACHINE LEARNING

☞ Learning의 종류(2)

● Unsupervised Learning

- label화 되지 않은 데이터를 통해 학습.
- News Grouping 같은 경우.
- 데이터를 이용해 스스로 학습



05. MACHINE LEARNING

☞ Machine Learning의 일반적인 응용

- 이미지 검색
- 고객 등급 예측
- 도난 신용카드 판별
- 시험 성적 예측
- 예상 매출액 산출

05. MACHINE LEARNING

☞ Supervised Learning Type(1)

● Linear Regression

➤ 7시간 공부하면 0 ~ 100점 중 몇 점을 받을 수 있는가?

x(시간)	y(점수)
10	95
8	80
5	68
2	15
1	5

05. MACHINE LEARNING

☞ Supervised Learning Type(2)

- Logistic Regression (Binary Classification)

➤ 7시간 공부하면 합격인가 혹은 불합격 인가? (2개 중 1개 , 0 or 1)

x(시간)	y(결과)
10	PASS(1)
8	PASS(1)
5	FAIL(0)
2	FAIL(0)
1	FAIL(0)

05. MACHINE LEARNING

☞ Supervised Learning Type(3)

- Multinomial Classification

➤ 7시간 공부하면 어떤 grade를 받을 수 있는가? (A,B,C,D,F – N개 중 1개)

x(시간)	y(결과)
10	A
8	B
5	D
2	F
1	F

05. MACHINE LEARNING

☞ Regression Analysis (회귀분석)

- 통계학에서 중요한 역할을 담당하는 자료분석 방법.
- 출발점 : “관찰된 자료들이 어떤 특정한 경향성을 가지고 있지 않을까?” 라는 의문
- 관찰된 자료의 변수들 사이에 나타나는 경향성 혹은 의존성을 수학적으로 판별하고자 하는 기법
- 이러한 경향성이나 의존성을 발견한다면 앞으로 발생할 일들에 대한 예측(prediction)이 가능.

05. MACHINE LEARNING

☞ Regression Analysis (회귀분석)

- 예) 여름철의 기온과 코카콜라의 판매량

- 두 변수 사이에 어떤 관계가 있을 것이라 생각되고 이에 따른 여러가지 가설(regression model)을 만들 수 있다.
- 만약, 우리가 얻은 회귀모형이 타당하다면 “독립변수와 의존변수 간에 의존관계가 있다” 라고 말할 수 있다.
- 또한 독립변수의 값을 토대로 종속변수의 값을 예측할 수 도 있다.

- 회귀분석에서 독립변수가 하나일 경우 단순회귀분석(simple regression), 독립변수가 둘 이상인 경우 다중회귀분석(multiple regression) 이라고 한다.

05. MACHINE LEARNING

☞ Simple Linear Regression Model (단순선형회귀모델)

- 독립변수와 종속변수의 경향을 알아보는 가장 기본적인 방법은 데이터 간의 관계를 1차 함수 (Linear function) 형태로 표현하는 것.

- 1차 함수는 종속변수와 독립변수 이외에 2가지 요소가 더 포함
 - 기울기(slop), y절편 (y-intercept)

$$y = \beta_0 + \beta_1 x$$

- 수학 공부한 시간과 수학시험성적의 관계
 - 수학 공부한 시간을 독립변수 x로 설정
 - 수학시험성적을 종속변수 y로 설정

05. MACHINE LEARNING

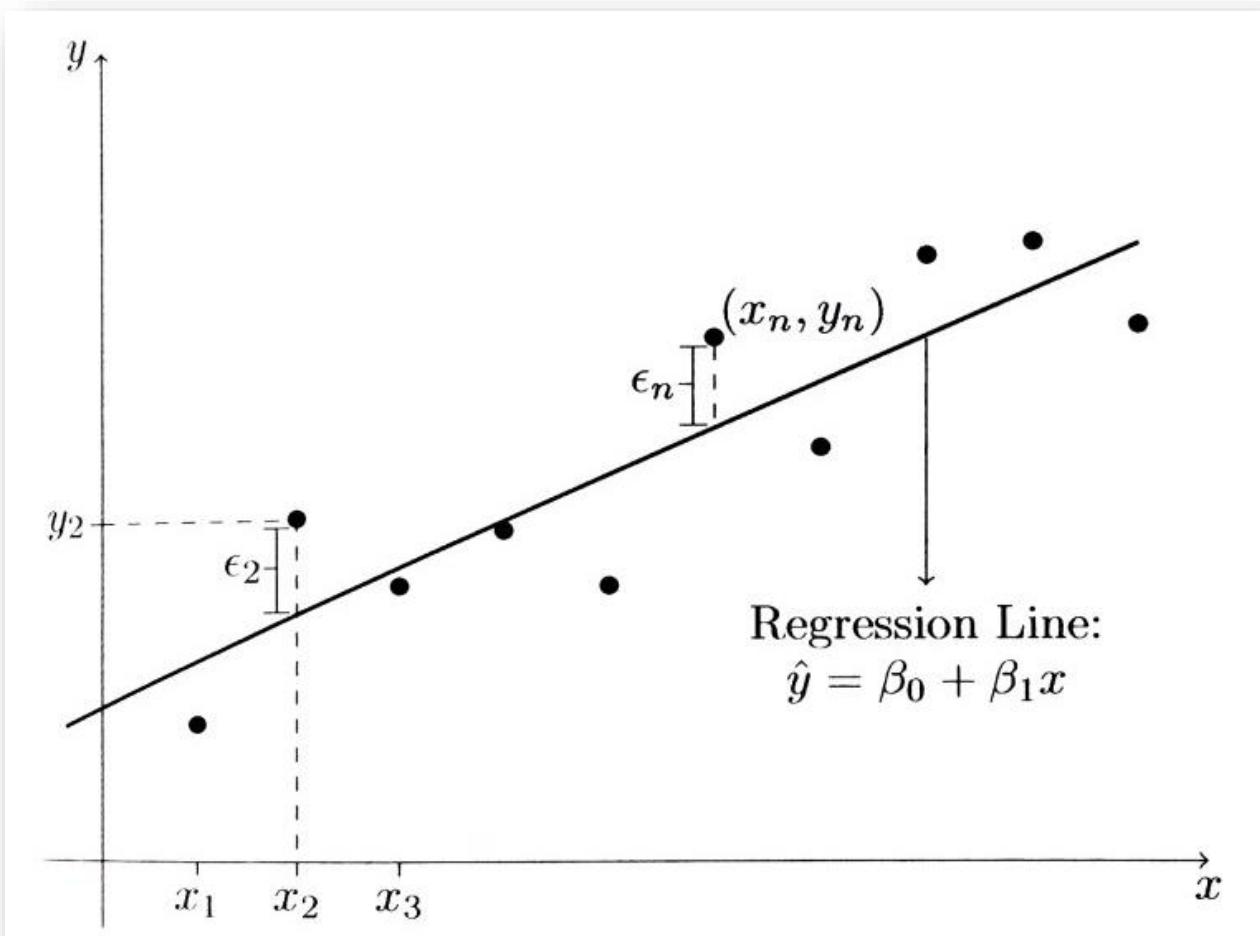
☞ Simple Linear Regression Model (단순선형회귀모델)

- 우리의 목표는 자료를 종합하여 오차가 가장 작은 일차함수를 만드는 것.
즉, 자료를 잘 표현하는 기울기와 절편을 구하는 것.

➤ 이렇게 얻은 일차함수식을 회귀선(regression line) 혹은 단순선형회귀모델(simple linear regression model)이라고 한다.

05. MACHINE LEARNING

☞ Simple Linear Regression Model (단순선형회귀모델)



05. MACHINE LEARNING

☞ Simple Linear Regression Model (단순선형회귀모델)

- 이전 그림과 같은 regression line을 만들려면 주어진 자료를 잘 표현하는 β_0 (y절편)와 β_1 (기울기)를 구해야 한다.
- β_0 (intercept coefficient)와 β_1 (slop coefficient)라고 불리는 regression coefficient(회귀계수)를 구하는 방법은 크게 2가지
 - 첫번째 방법은 기댓값과 공분산을 이용해 구하는 방법
 - 두번째 방법은 최소 제곱법
 - Tensorflow를 이용한 machine learning에서는 최소 제곱법을 이용해서 회귀계수를 구한다.

05. MACHINE LEARNING

☞ Linear Regression – Supervised Learning Type(remind)

- Linear regression

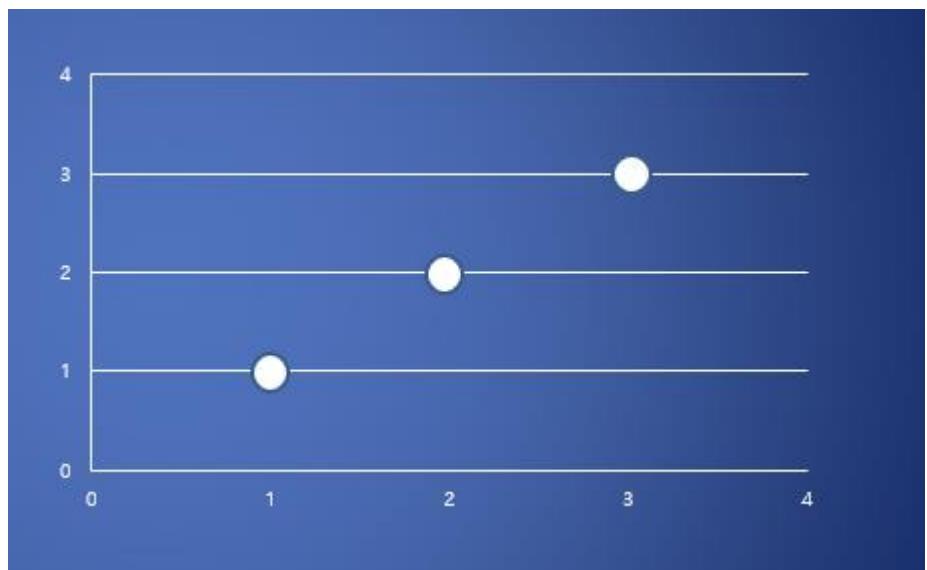
➤ 7시간 공부하면 0 ~ 100점 중 몇 점을 받을 수 있는가?

x(시간)	y(점수)
10	95
8	80
5	68
2	15
1	5

05. MACHINE LEARNING

☞ Linear Regression – 사용할 training data set

x	y
1	1
2	2
3	3



05. MACHINE LEARNING

☞ Linear Hypothesis

- 많은 데이터와 현상들이 linear한 형태를 가진다.

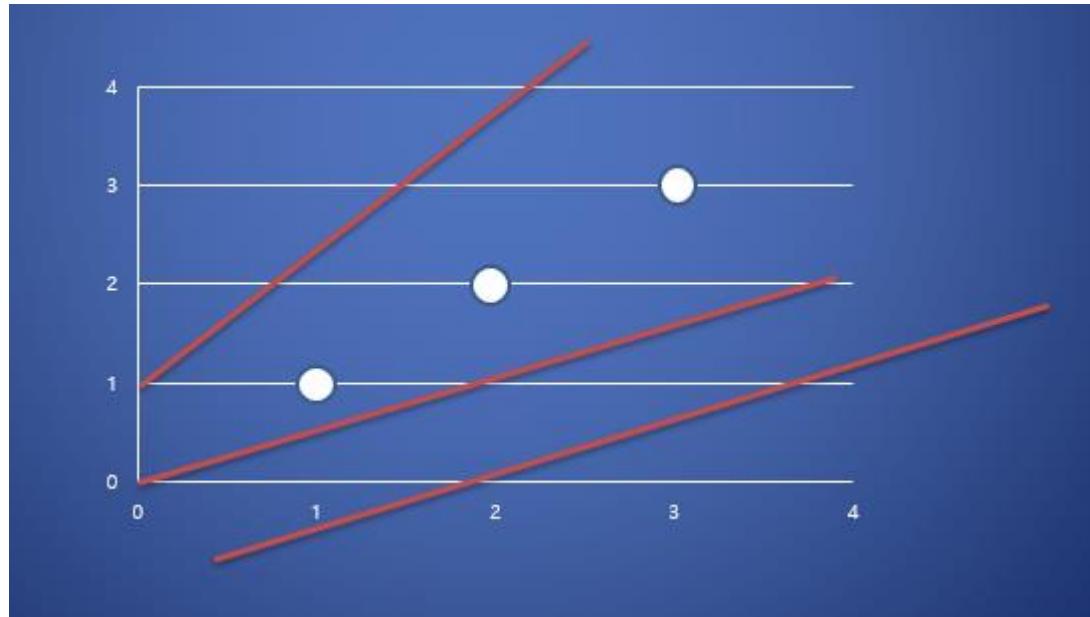
- 일반적으로,
- 많은 시간 공부를 하면 점수를 더 많이 받고
- 오랜 시간 일을 하면 더 많은 돈을 벌고
- 공을 세게 던지면 던질수록 더 멀리 나아가고
- 배달지역이 멀수록 배달시간이 오래 걸린다.

- 우리 예제도 이러한 linear한 형태라고 가정하고 하나의 가설(Hypothesis)을 세운다.

05. MACHINE LEARNING

☞ Linear Hypothesis

- Linear Hypothesis를 정의하는 것은 training data set에 잘 맞는 linear한 선을 긋는 것으로 생각할 수 있다.
- Hypothesis를 수정해 나가면서 데이터에 가장 적합한 선을 찾는 과정이 바로 학습.



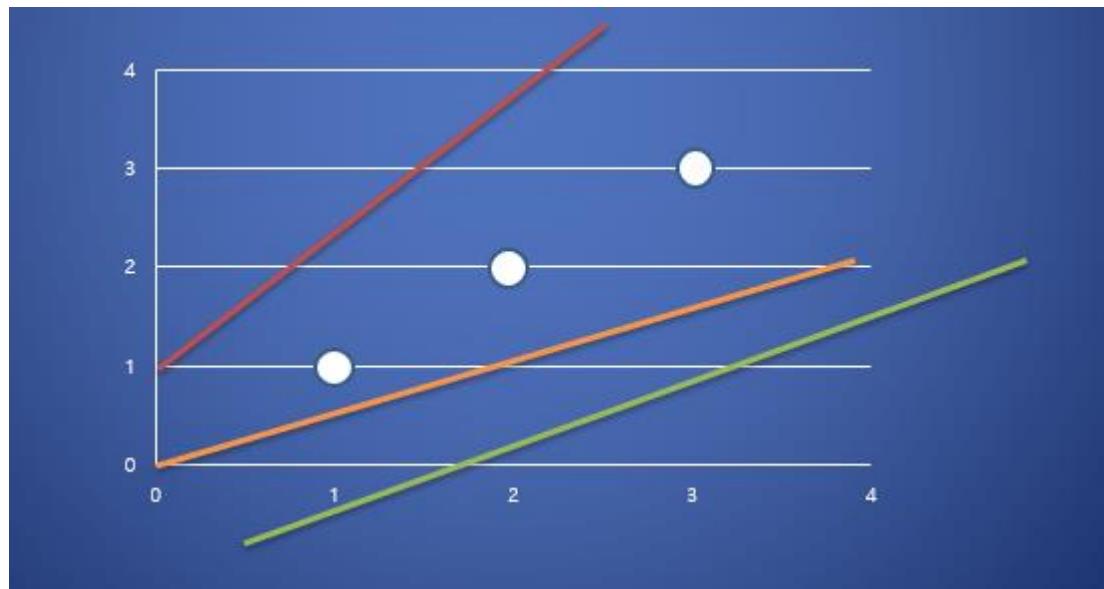
05. MACHINE LEARNING

☞ Linear Hypothesis

- 이러한 선들을 수학적으로 표현하면 다음과 같다. 

Hypothesis(가설)

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

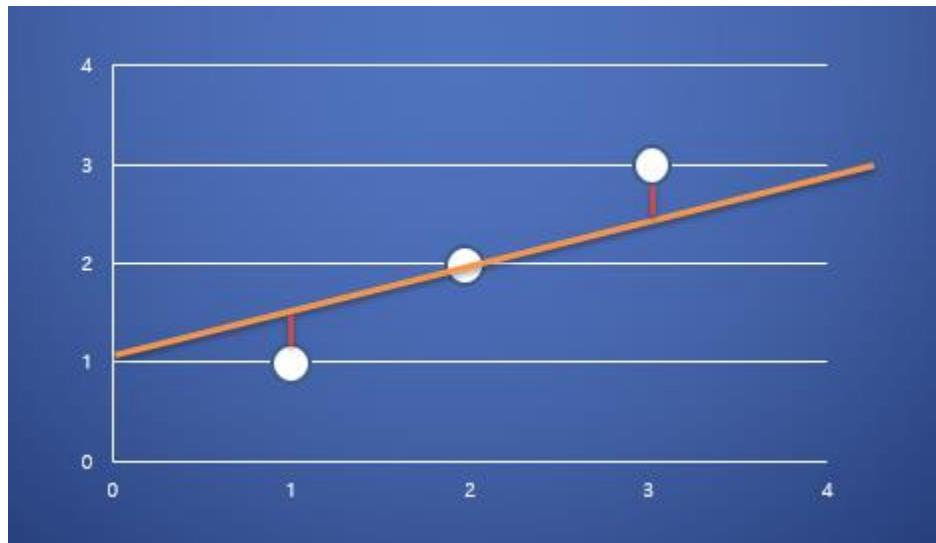


05. MACHINE LEARNING

☞ Linear Hypothesis

- 우리가 원하는 것은 training data set에 가장 적합한 W와 b의 값을 찾는 것.
 - 가장 적합한 W와 b의 값을 찾으려면 어떻게 해야 하는가?
 - Cost(Loss) Function 사용 – 최소제곱법 

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

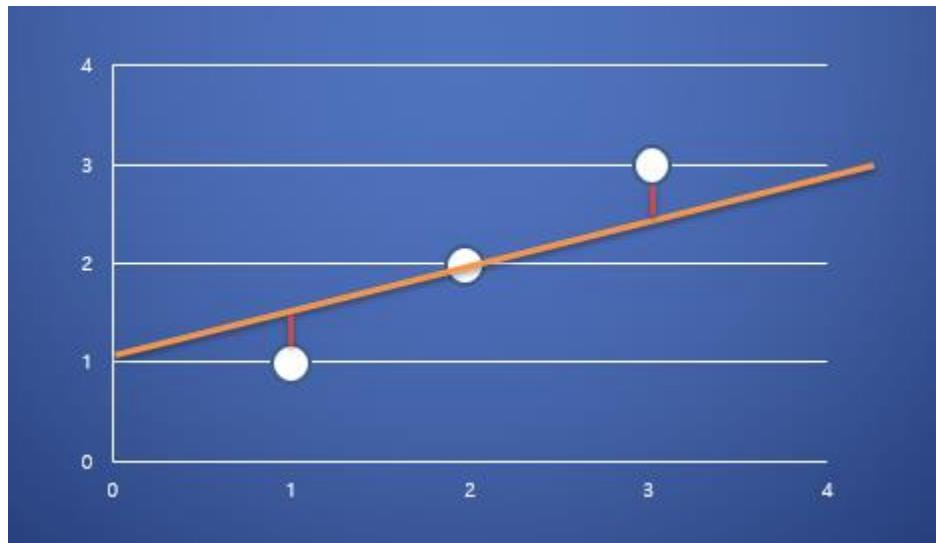


05. MACHINE LEARNING

☞ Cost(Loss) Function

- 가설과 training data간의 값의 차를 계산하는 함수 $H(x) = Wx + b$

$$\frac{(H(x^{(1)}) - y^{(1)})^2 + (H(x^{(2)}) - y^{(2)})^2 + (H(x^{(3)}) - y^{(3)})^2}{3}$$

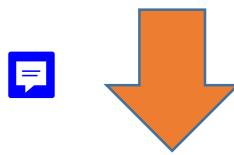


05. MACHINE LEARNING

☞ Cost(Loss) Function

- 가설과 training data간의 값의 차를 계산하는 함수

$$\frac{(H(x^{(1)}) - y^{(1)})^2 + (H(x^{(2)}) - y^{(2)})^2 + (H(x^{(3)}) - y^{(3)})^2}{3}$$



$$cost(W, b) = \frac{1}{m} \sum_{i=0}^m (H(x^{(i)}) - y^{(i)})^2$$

05. MACHINE LEARNING

☞ Cost(Loss) Function

- 결국 Cost function의 값을 최소로 만드는 W와 b의 값을 찾아내는 것이 학습의 목표

$$cost(W, b) = \frac{1}{m} \sum_{i=0}^m (H(x^{(i)}) - y^{(i)})^2$$

05. MACHINE LEARNING

☞ 실습

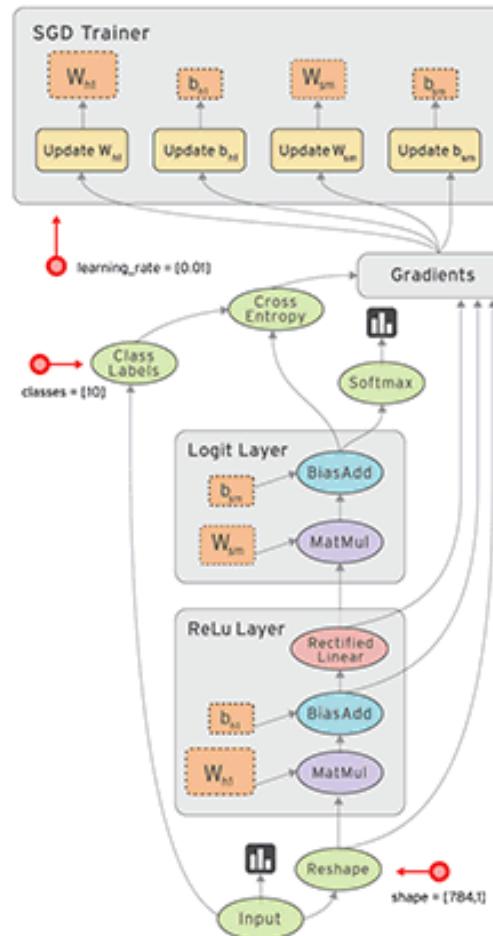
- 아래의 코드를 이용하여 Hello World 출력
 - Node는 mathematical operation, 데이터의 입출력 작업을 수행.

05. MACHINE LEARNING

실습

● TensorFlow

- TensorFlow is an open source software library for numerical computation using data flow graphs.
- Data flow graphs는 Node와 Edge로 구성된 방향성 있는 그래프로 표현.
- Node는 mathematical operation, 데이터의 입출력 작업을 수행.
- Edge는 동적 크기의 다차원배열(Tensor)을 Node로 실어 나르는 역할



05. MACHINE LEARNING

☞ 실습

- Jupyter Notebook 단축키 (참고)

단축키	기능
Shift-Enter	run cell, select below
Ctrl-Enter	run cell
Alt-Enter	run cell, insert below
A	insert cell above
B	insert cell below
X	cut selected cell
C	copy selected cell
V	paste cell below
Shift-V	paste cell above
D, D	delete selected cell
Z	undo last cell deletion
L	toggle line number
O	toggle output

05. MACHINE LEARNING

☞ 실습

- 간단한 수학 연산 수행

05. MACHINE LEARNING

☞ 실습

- placeholder 사용

05. MACHINE LEARNING

☞ Tensors

- Tensors는 다차원 매트릭스

05. MACHINE LEARNING

☞ Tensors

- Types은 Tensor의 데이터 타입을 의미, Type 변경 가능

05. MACHINE LEARNING

☞ Linear regression(remind)

- Hypothesis와 Cost function

Hypothesis(가설)

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

Cost function

$$cost(W, b) = \frac{1}{m} \sum_{i=0}^m (H(x^{(i)}) - y^{(i)})^2$$

05. MACHINE LEARNING

☞ Linear regression 실습 (1)

05. MACHINE LEARNING

☞ Linear regression 실습 (2)

05. MACHINE LEARNING

- ❖ Linear regression 실습 – placeholder를 이용하여 초기데이터 세팅

05. MACHINE LEARNING

☞ 학습 종료 후 데이터를 이용한 추정

05. MACHINE LEARNING

☞ Cost(Loss) function 심화

- 목적은 Cost function을 최소화 시키는 W와 b의 값을 찾는 것.

05. MACHINE LEARNING

☞ 이해를 위해 Hypothesis & Cost Function을 단순화 (상수제거)

$$H(x) = Wx$$

x	y
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8

$$cost(W) = \frac{1}{m} \sum_{i=0}^m (H(x^{(i)}) - y^{(i)})^2$$

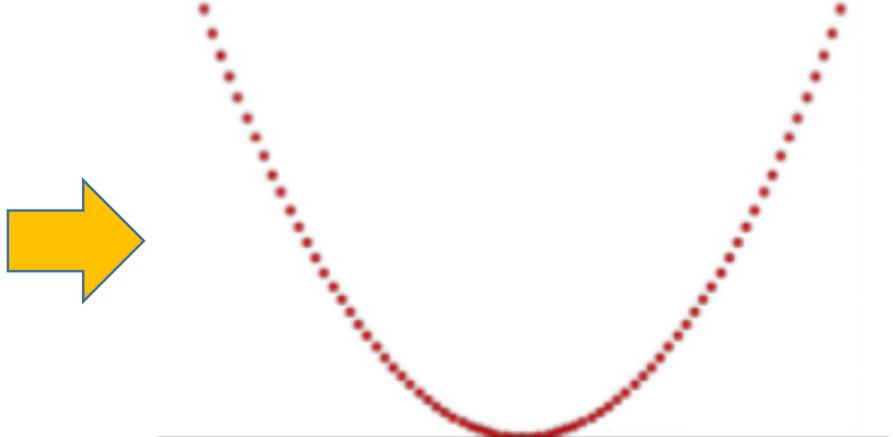
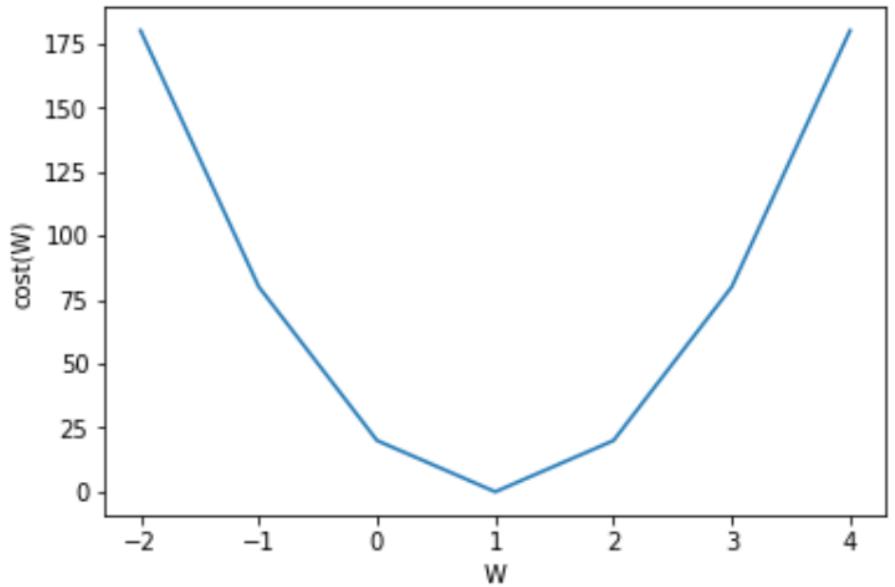
W	cost(W)
-2	?
-1	?
0	?
1	?
2	?
3	?
4	?

05. MACHINE LEARNING

☞ Cost Function의 그래프 생성 코드

05. MACHINE LEARNING

☞ Cost Function의 그래프



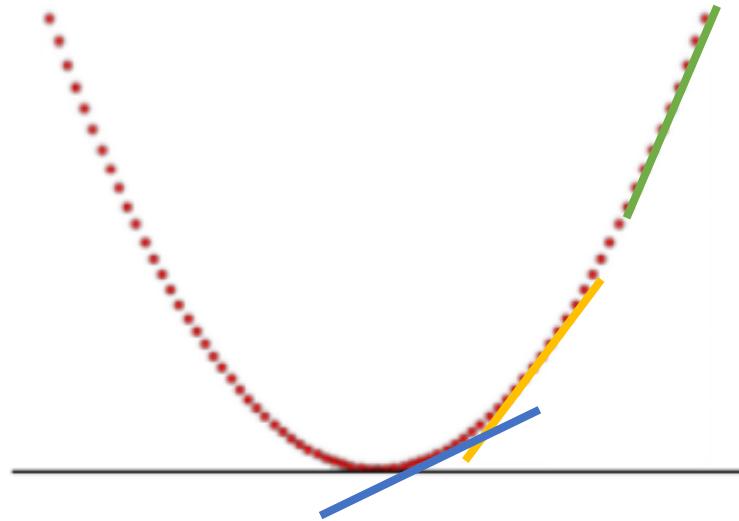
05. MACHINE LEARNING

☞ Cost function의 값을 최소화 시키는 W를 어떻게 알아낼 수 있는가?

- Gradient Descent Algorithm을 이용
- 최소값을 찾기 위해 많이 사용되는 대표적인 Algorithm
- 미분을 이용

Learning rate(상수)

$$W := W - \alpha \frac{\partial}{\partial W} cost(W)$$



05. MACHINE LEARNING

☞ Cost function의 값을 최소화 시키는 W의 값을 구하는 최종 수식

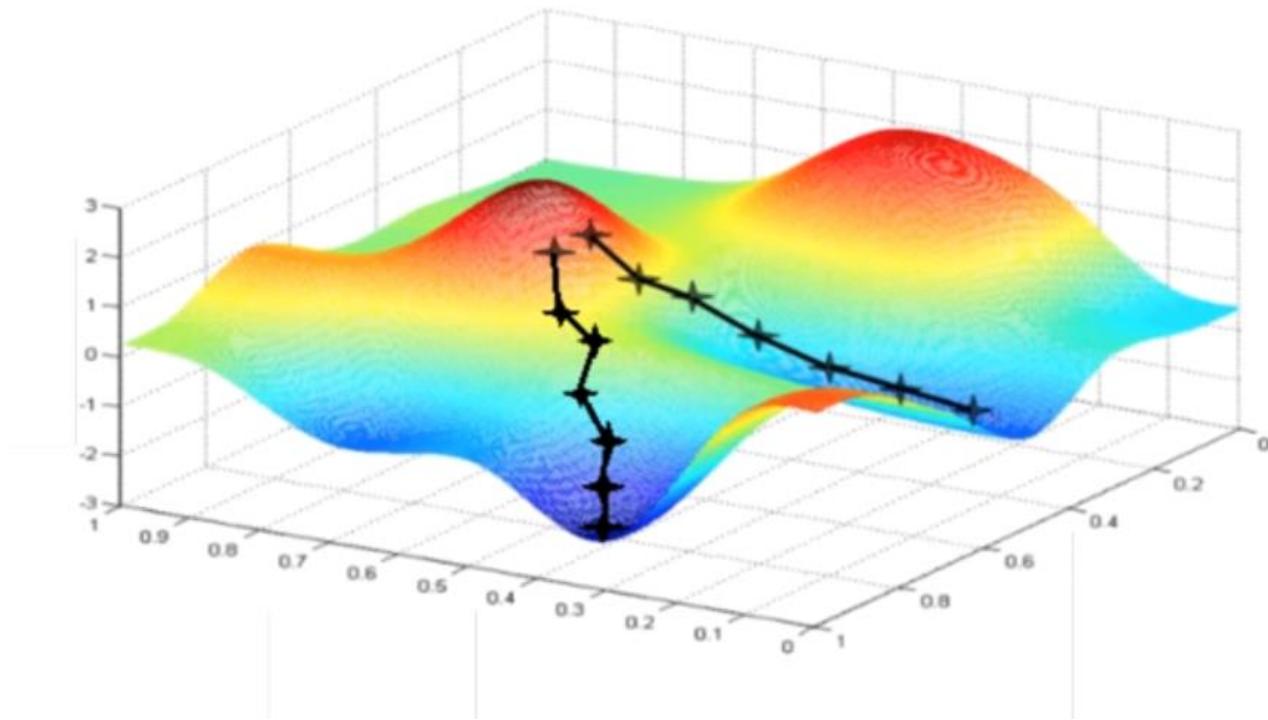
$$W := W - \alpha \frac{1}{m} \sum_{i=1}^m (Wx^{(i)} - y^{(i)})x^{(i)}$$

- 이렇게 수식을 얻었으면 이 수식을 프로그래밍으로 표현하면 된다.
- TensorFlow는 이런 복잡한 수식의 계산을 우리가 직접 프로그래밍 할 필요없이 함수 호출을 통해 계산할 수 있도록 해주는 library

05. MACHINE LEARNING

☞ Gradient Descent Algorithm의 예외

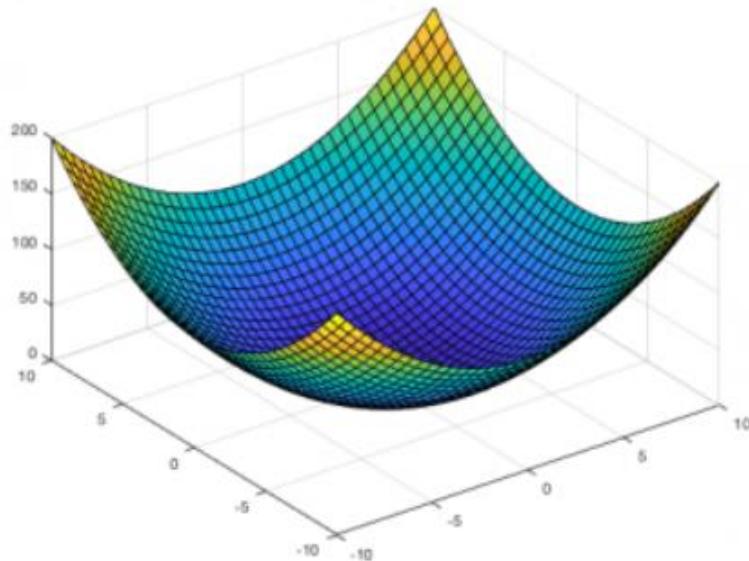
- Gradient Descent Algorithm은 항상 사용할 수 있는 것은 아니다.



05. MACHINE LEARNING

☞ Convex Function

- Cost Function은 다음 그림과 같은 Convex function 형태가 되어야 Gradient Descent Algorithm을 사용할 수 있다.
 - Cost function을 설계할 때 Convex function의 형태를 가지는지를 확인해야 한다.



05. MACHINE LEARNING

☞ Multi-variable(multiple) linear regression

- 여러 개의 input을 이용한 linear regression

➤ 3번의 quiz 점수를 기반으로 시험성적 예측

x1(quiz1)	x2(quiz2)	x3(quiz3)	y(exam)
73	80	75	152
93	88	93	185
89	91	90	180
96	98	100	196
73	66	70	142