

수업 기획

- **대상**

- 배경지식이 전혀 없거나 너무 오래전에 배운 내용이라 수학의 대부분을 잊어버린 상태에서 이제 막 머신러닝 · 딥러닝에 흥미를 느끼는 분
- 머신러닝 · 딥러닝에 이미 관심이 높지만 당장 구현보다는 체계적으로 내용을 이해하고 싶은 개발자 분
- 아래 식을 한번 정도는 직접 미분하여 코드로 적어보고 싶은 입문자

$$\mathbf{w} := \mathbf{w} - \alpha \frac{\partial}{\partial \mathbf{w}} \left(\frac{1}{2N} \sum_{n=1}^N \{y(x_n, \mathbf{w}) - t_n\}^2 \right)$$

- **선수 지식**

- 다항식이 무엇인지 대충 알고 있다.
- 식으로 표현된 직선을 알 수 있다. 식으로 표현된 이차곡선을 알 수 있다.

- **진행 방식** : 선형회귀의 소개 → 선형회귀를 구성하는 몇 가지 수학 요소 공부 → 다시 선형회귀 공략
- **수강 후 결과** : 머신러닝 · 딥러닝 공부의 출발점에 선다.

수업 순서(오프라인)

- 0주 : python문법
- 1주 : 머신러닝과 선형 회귀의 개념, 함수, 미분의 개념(4)
- 2주 : 여러 함수의 미분, 행렬과 벡터(4)
- 3주 : 자동미분, 최적화1 (2)
- 4주 : 최적화2, 인공신경망(2)
- 5주 : 확률이론1 (2)
- 6주 : 확률이론2, 선형회귀 다시 보기 (2)
- 7주 : 선형회귀 마무리 (2)

수업 순서(온라인)

- 머신러닝과 선형 회귀의 개념,
- 함수
- 미분
- 행렬과 벡터
- 자동 미분
- 최적화
- 인공신경망
- 선형회귀 다시 보기

수업 보조 자료

- 수업의 이해를 돋기 위해 다음 아이콘이 나올 때 보조 자료와 함께 보기 권장

<https://github.com/metamath1/ko-IT>



주피터 노트북 파일



자바스크립트 보조 프로그램

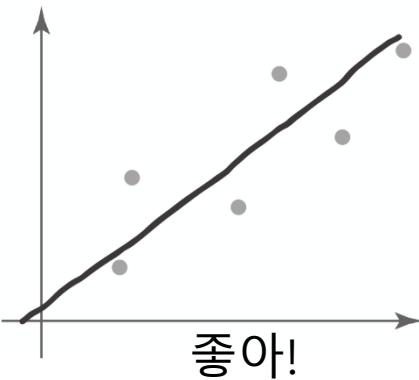
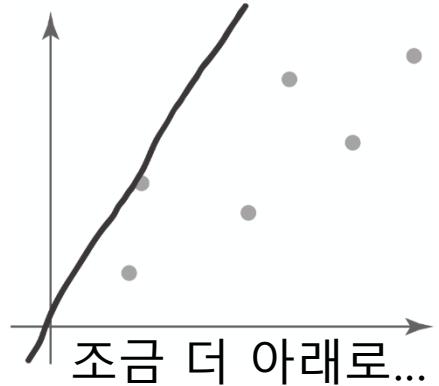
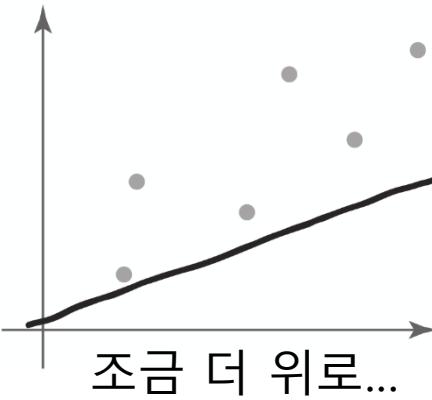
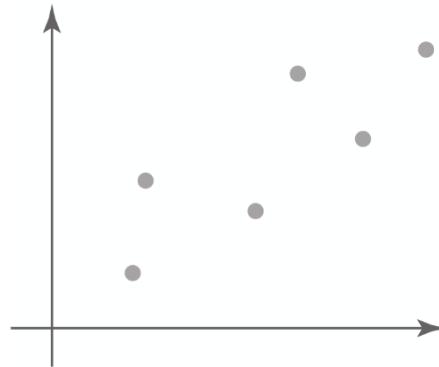
머신러닝?

- **넓은 의미** : 컴퓨터를 이용한 문제 해결에 있어서 지식기반, 규칙 기반 방법이 아닌 데이터와 범용 알고리즘으로 퍼포먼스를 개선하는 방법
- **Tom Mitchell (1998)** : Well-posed Learning Problem: A computer program is said to learn from experience E with respect to some **task T** and some performance **measure P**, if its performance on T, as measured by P, improves with **experience E**.
"Machine Learning", Coursera, Andrew Ng
- **좁은 의미** : 주어진 데이터를 가장 잘 표현하는 함수를 찾는 것



<https://medium.com/@ezralazuardy/how-machine-learn-c2f73f60ef14>

아이가 선을 그린다면...

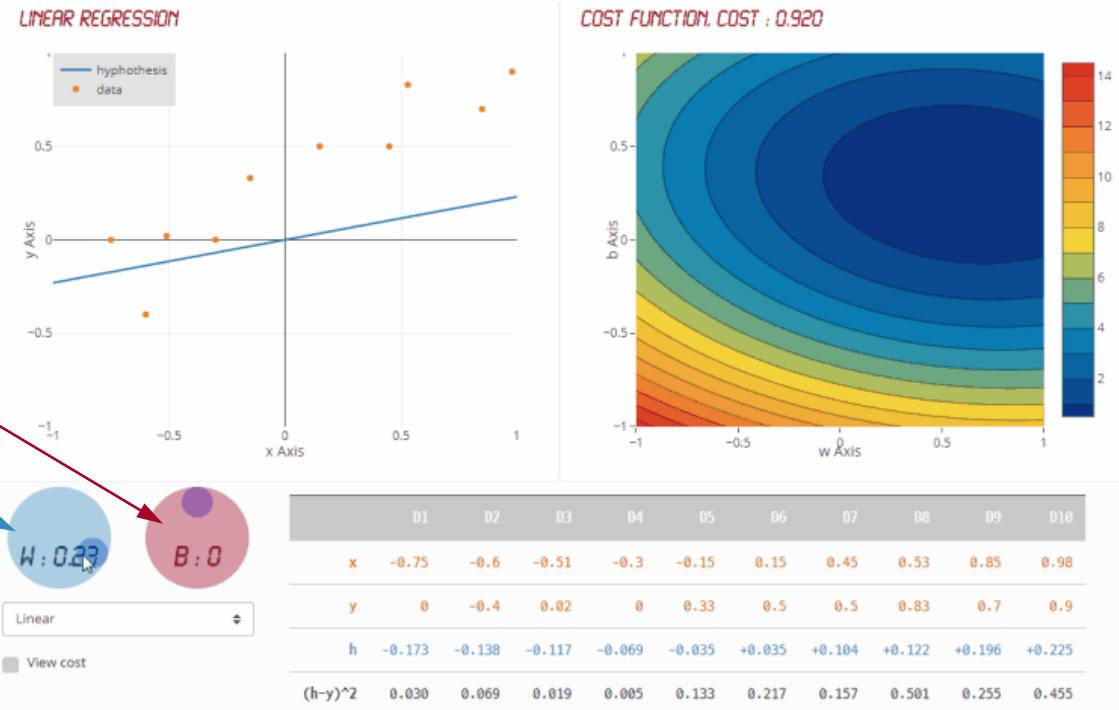


- **Task T** : 선을 그린다.
- **Measure P** : 선이 점에 잘 맞는가?
- **Experience E** : 만족할 때 까지 계속 선을 그린다. 빨간 점에서 선이 멀어지면 좋지 않다.

손으로 선형회귀...

- W, B 훈련을 돌리면서 가장 알맞은 직선 찾기

$$y = Wx + B$$



머신러닝 분류

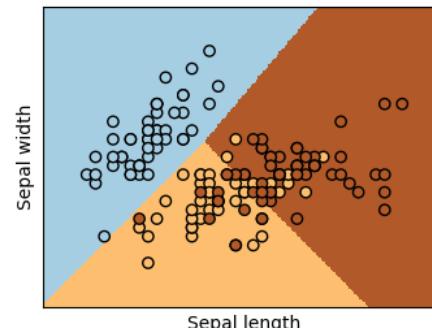
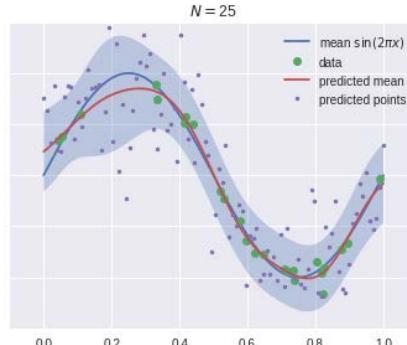
- 지도학습Supervised

- 회귀 문제Regression : 선형 회귀Linear Regression

- 정답 : 연속된 실수
 - 예 : 대지면적에 따른 집값, 시간에 따른 트랜지스터 집적 개수, 기온에 따른 빙과류 판매량

- 분류 문제Classification : 로지스틱 회귀Logistic Regression

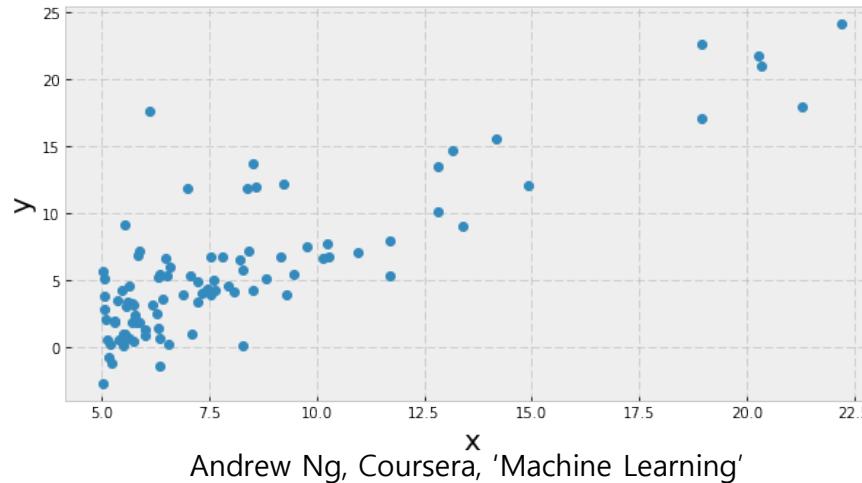
- 정답 : (0,1) 또는 (0, 1, ..., K)
 - 예 : 개-고양이 분류, 양성증양-악성증양 분류



http://scikit-learn.org/stable/auto_examples/linear_model/plot_iris_logistic.html#sphx-glr-auto-examples-linear-model-plot-iris-logistic-py

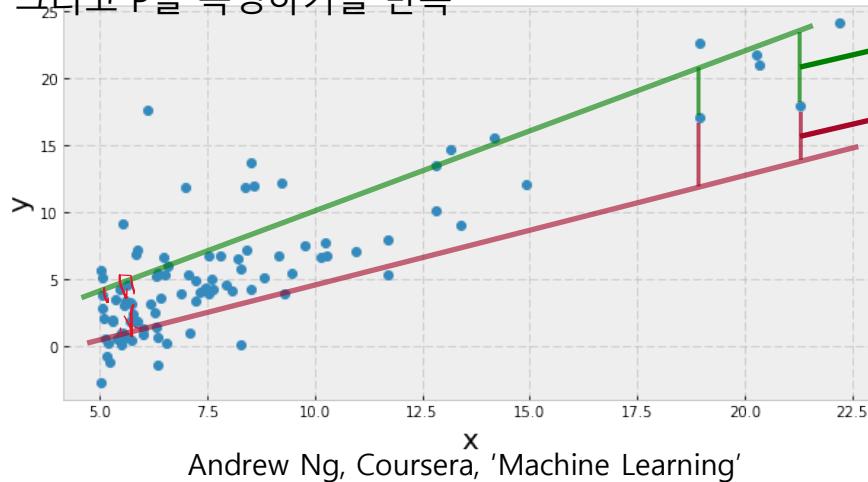
직선 찾기

- 데이터 출처 : Coursera 강의 'Machine Learning', Andrew Ng
(코드:01-01-linreg-intro.ipynb)
- 데이터의 의미
 - x : 주택 대지 면적, y : 주택 가격
 - x : 기온, y : 커피판매량
 - 기타 등등...
- 데이터가 보이고 있는 직선의 관계를 찾을 수 있는가?



직선 찾기 - 개념

- 아이의 직선 그리기처럼.....
- 모든 점을 지나는 직선은 그릴 수 없으니 선을 그려보고 점과 선과의 거리를 최대한 줄여보자
- **Task T** : 선을 그린다.
 - $y = w_0 + w_1x$
 - w_0 와 w_1 을 결정하기
- **Measure P** : 선과 점의 수직거리의 총합
- **Experience E** : 선을 그리고 P를 측정하기를 반복



$$\frac{1}{2N} \sum_{n=1}^N \{h(x_n, \mathbf{w}_g) - y_n\}^2$$

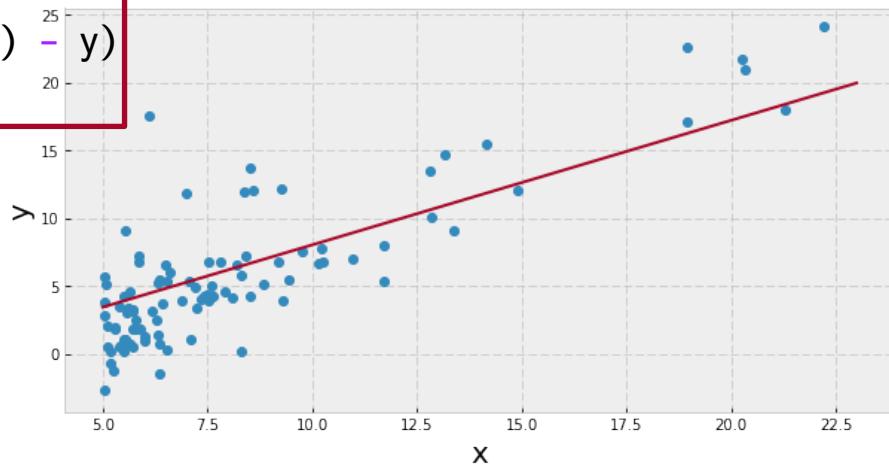
ERROR

$$\frac{1}{2N} \sum_{n=1}^N \{h(x_n, \mathbf{w}_r) - y_n\}^2$$

ERROR

직선 찾기 - 경사하강법

- num_iters = 1500
- n = X.shape[0]
- w = np.random.rand(2).reshape(2,1)
- alpha = 0.01
- ```
for i in range(num_iters) :
 c = (1/n) * np.dot(X.T, np.dot(X,w))
 w -= alpha * c
```



# 경사하강법 수학요소

- 핵심 코드에 숨어 있는 수학 요소들

$$\mathbf{w} = \mathbf{w} - \alpha \frac{\partial}{\partial \mathbf{w}} \left( \frac{1}{2N} \sum_{n=1}^N \{h(x_n, \mathbf{w}) - y_n\}^2 \right)$$

```
for i in range(num_iters):
 c = (1/n) * np.dot(X.T, np.dot(X,w) - y)
 w -= alpha * c
```

Linear Algebra(선형대수)

Calculus(미분)

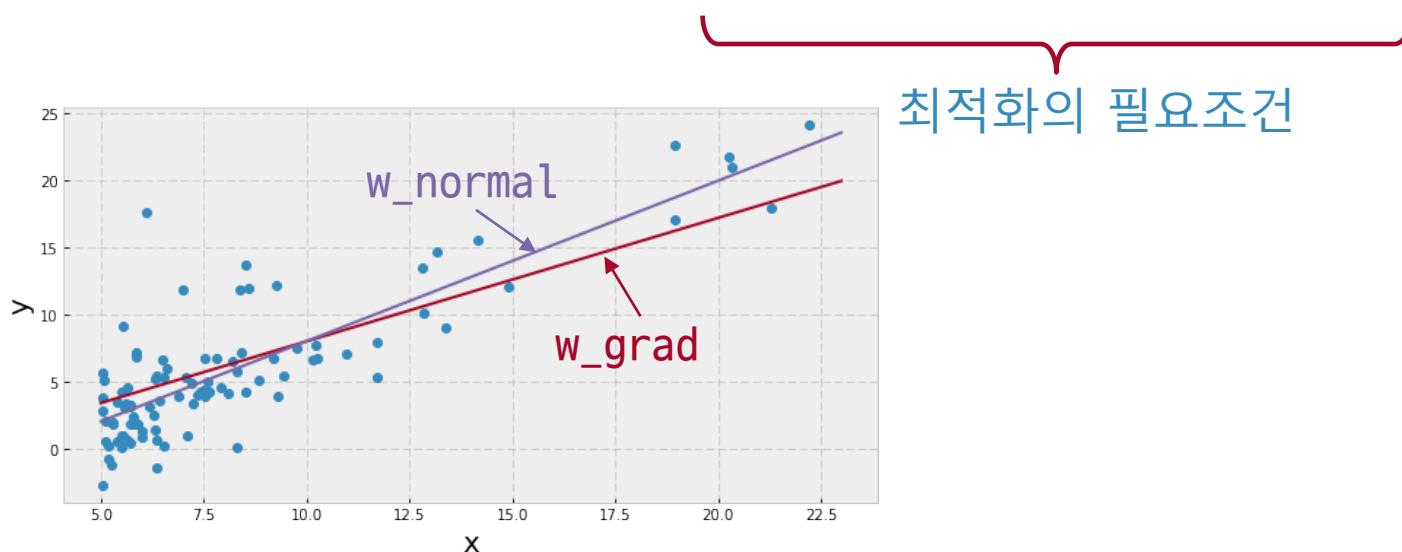
Optimization(최적화)

# 직선 찾기 - 정규방정식

$$\mathbf{w} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{t}$$

#normal eq.

```
w_normal = np.linalg.solve(np.dot(X.T, X), np.dot(X.T, y))
```



# 왜 선형회귀?

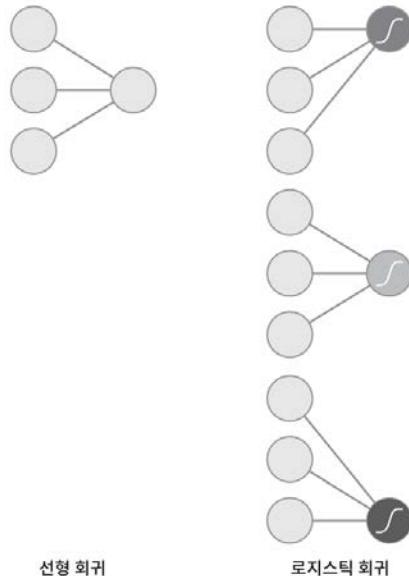
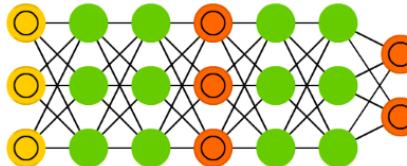
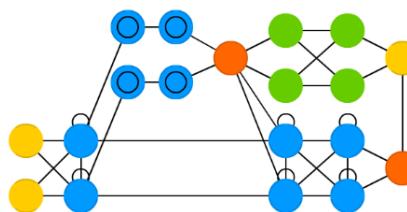


그림 1-13 선형 회귀에서 다중 퍼셉트론까지

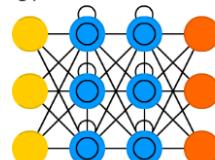
Generative Adversarial Network (GAN)



Attention Network (AN)



Long / Short Term Memory (LSTM)



딥러닝으로  
향하는  
여정의 시작

# 목표설정

- 지금까지 논의를 바탕으로 다음 식을 적어도 “합리적 수준”에서 이해하고 구현하기 위해.....

$$\mathbf{w} = \mathbf{w} - \alpha \frac{\partial}{\partial \mathbf{w}} \left( \frac{1}{2N} \sum_{n=1}^N \{h(x_n, \mathbf{w}) - y_n\}^2 \right)$$

최적화      미분      함수

$$\mathbf{w} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

- 파이썬 기본 문법
- 머신러닝 관련 몇 가지 이슈들

행렬과 벡터연산