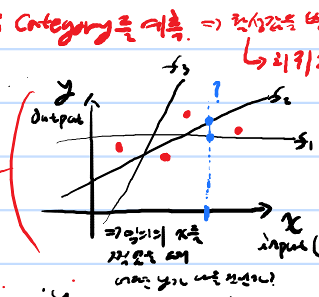
# 회귀 (Regression)

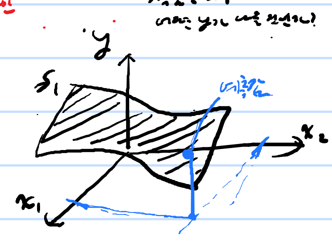
- 주어진 입/출력 데이터쌍들을 바탕으로, 출력값을 예측하는 문제의 일종 (교사학습은 이용)
- 예측을 회귀와 분류로 구분 → *분류 : Category를 예측. ⇒ 확성값을 변경하여 분류에서 Category의 확률을 예측하도록 만듦.
  ↳ 회귀로 볼 수도 있음.
- 실수값을 예측하는 문제를 회귀라고 함.




정확히 실제값을 예측할 수 있을 수도 있음 (w) ⇒ 선형문제 가능 이라면 분리했듯이.

어느 정도 비슷한 값을 추정하는 것이 의미가 있음.

($f_1, f_2, f_3$ ⇒ model)
↳ 주어진 data들을 잘 예측하는 모든 선형모델 (적는 model)
(1차원) ⇒ scalar




$\underline{x} = [x_1, x_2]^T$ ⇒ vector.
($f_1$ ⇒ model)

- 선형회귀모델.

⇒ $y = w_1 x + w_0$ → 예측을 가능하게 하는 모델을 찾는 것이 필요함. 입력값이 하나인 경우 (1차원)

출력값

$w_2 x_2 + w_1 x_1 + w_0$ (2차원)

# 일반적인 선형회귀

- $\underline{X} = \{\underline{x}_1, \underline{x}_2, \cdots, \underline{x}_n\}$ → i-th element
- $\underline{Y} = \{y_1, y_2, \cdots, y_n\}$ — $\underline{x}_i = [x_{i1}, x_{i2}, \cdots, x_{id}]^T$ (특징공간의 크기는 d차원)

  순서쌍 ⇒ 입·출력 데이터쌍

w도 가중치.

⇒ $w_1 x_{i1} + w_2 x_{i2} + \cdots + w_d x_{id} = \boxed{\underline{x}_i \cdot \underline{w} = y_i}$ → 선형시스템 (d차원 n개 해)

벡터의 내적

- 위의 방정식을 정확히 만족한다면, 선형회귀모델을 사용해서 모든 샘플을 정확하게 맞출 수 있다. → 모든 샘플이 하나의 직선 상에 놓인 경우. ($\forall i$)
- 주어진 방정식에서 w를 찾는 문제는, 미지수가 d개이고 방정식의 개수가 n개인 선형시스템의 해를 찾는 문제와 같음. → 해가 무수히 많을 수도 있고, 유일할 수도 있고 없을 수도 있음

$$\Rightarrow X\underline{w} = \underline{y}$$

$$X = \begin{pmatrix} \underline{x}_1^T \\ \vdots \\ \underline{x}_n^T \end{pmatrix} \in \mathbb{R}^{n \times d}, \quad \underline{w} = [w_1, \cdots, w_d]^T \in \mathbb{R}^d$$

$$\underline{y} = [y_1, \cdots, y_n]^T \in \mathbb{R}^n$$

- 실제 회귀문제에서는 모든 sample들이 하나의 직선 상에 놓이는 경우는 잘 없음.
- 위의 방정식을 정확히 만족하는 w를 찾는 것은 어렵다. (불가능)
- ★ 대신, 가장 근접하게 만족하면서 $\boxed{\text{오차}}$를 최소화하는 w를 찾아야 함.
  ↳ norm을 주로 사용 — *PLA에서는 (분류) ⇒ 분류가 틀리는 경우를 최소화

# 선형회귀의 목적함수 (비용함수) → 실제에 정답이 있기 때문

$$\Rightarrow J(\underline{w}) = \frac{1}{2}\sum_{i=1}^{n}(\underline{x}_i^T \underline{w} - y_i)^2$$

오차 → 정확히 만족한다면 0이 됨

scaling factor, 내적, 실제 (label)

⇒ 여러가지 조건이 만족하도록 설계 (항상 양수여야 함)

$$= \frac{1}{2}\|X\underline{w} - \underline{y}\|_2^2$$

미분값 정리하기 위해 · 2norm의 제곱 ↳ 오차 최소화로 설계 ⇒ 0이면 완벽히 회귀

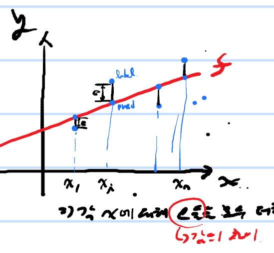
수학적 편의를 위해 (미분시 계산이 편리해지도록)

$$= \frac{1}{2}(X\underline{w} - \underline{y})^T(X\underline{w} - \underline{y})$$

⇒ J를 최소화하는 w를 찾아야 함

- 선형회귀의 목적함수는 2차함수 (quadratic) 형태로 주어짐.
- 이는 각 sample별로 계산한 오차의 제곱을 모두 더한 것.

⇒ $\boxed{\text{J의 최소를 찾기 위해 미분을 통해 기울기가 0이 되는 지점을 찾음}}$




J(w)

$$\frac{\partial J(\underline{w})}{\partial \underline{w}} = X^T X \underline{w} - X^T \underline{y} = \underline{0}$$

$$\Rightarrow \underline{w} = (X^T X)^{-1} \cdot X^T \underline{y} \triangleq \underline{w}^*$$ 최적값

↳ inverse가 존재하지 않으면 쓰일 dataset.

$X \in \mathbb{R}^{n \times d}$

$X^T X \in \mathbb{R}^{d \times d}$

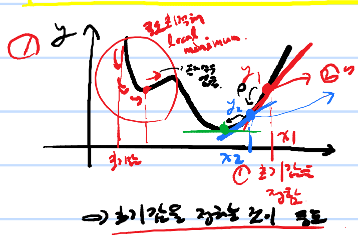
↳ 정해진 수학적 식에 의해 결정 (반복 X)

$$\therefore y_{pred} = \underline{x}_{new} \cdot \underline{w}^*$$

# 선형회귀와 선형분류의 차이

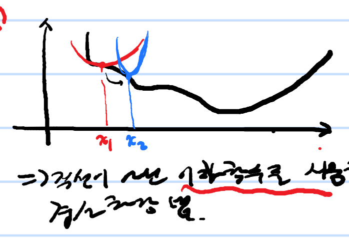
- 선형분류에서는 최적화를 위해 반복 알고리즘 (PLA)을 사용. ⇒ 입력과 출력이 완전히 선형관계가 아님 (활성화함수가 선형함수가 아님)
  ↳ ex) step func. etc.
- 선형회귀에서는 최적화를 위해 반복 기법을 사용할 필요 없이, 한 번에 최적해를 찾을 수 있음.
  ↳ 입력과 출력이 선형관계임. (비선형에서는 어려움?) ⇒ 지점들 나눠 선형으로 각 지점을 해결.

# 경사하강법 (Gradient Descent Method)

① 


⇒ 초기값을 정하는 것이 중요

⇒ 곡선이 낮아지는 방향으로 직선이 낮아지는 방향을 가는거로 따라감
ρ : 일정한 정도로 ↳ 어느 정도를 적절히 움직이는 것도 중요.
⇒ 만족할만한 최소값 / 수렴 (항상 정확한 최소값이 나올 수 없음)

② 


⇒ 접선이 사선 1차함수로 사용하여 경사하강 법.

⇒ 최적해로 도달하는 속도가 느릴 수 있음 (학습이 오래 비효율적일 때)

- 현재 위치에서 내려가는 방향을 찾아 조금씩 내려가면서 최적해를 찾는 수치적 기법
  ↳ 선형회귀에서는 한 번에 최적해를 찾을 수도 있음 (일반적인 상황에서는 그렇지 않음)
- 비용함수를 미분해서 내려가는 방향 (그레이디언트 벡터)을 찾음
  ↳ 비용함수의 내리막 방향 — vector
- 얼마만큼 이동할지를 조절하기 위해 학습률 (learning rate)을 곱함.

$$\Rightarrow \underline{\theta} \leftarrow \underline{\theta} - \rho \frac{\partial J}{\partial \underline{\theta}} = \underline{\theta} - \rho \nabla J$$

w 대신 θ — -(기울기) (비용함수) — -(기울기) (비용함수)
↳ Model을 결정하게 되는 parameter.

- 그레이디언트를 계산하는 데이터 포인트는 학습데이터 샘플을 사용
  X, Y