Chap.1 선형 회귀 (Linear Regression)

방 수 식 교수
(bang@tukorea.ac.kr)

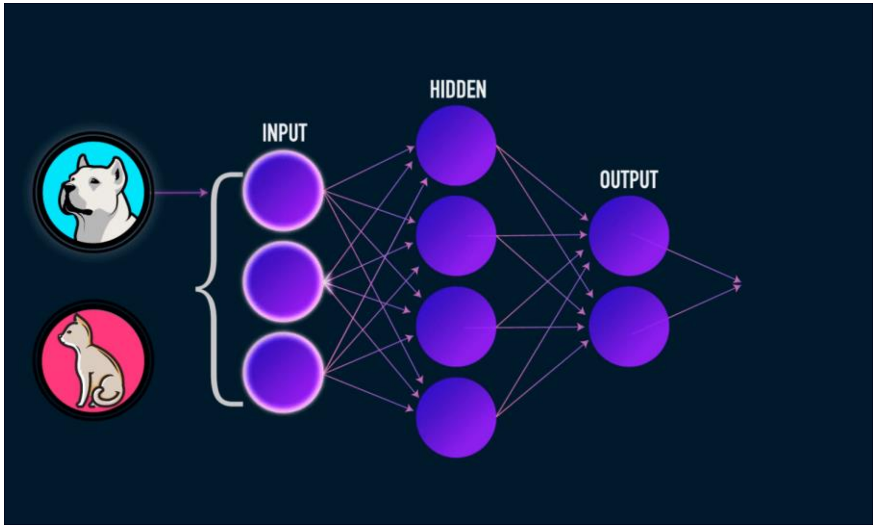
한국공학대학교 전자공학부

2024년도 1학기
머신러닝실습 & 인공지능설계실습1

본 강의의 목표

지능형 예측·진단 연구실
Intelligent Prognostics & Diagnostics Lab.

* AI Starter — Train and test your first neural network classifier in Keras from scratch



우리는 자비스를 만들려는 것이 아닙니다.
그저 <u>단순한 형태의 분류기 정도</u>를 만드는 겁니다.

두려워하지 마세요!

준비 되셨나요?

벡터와 행렬

- 벡터(Vector)
  - 샘플을 특징 벡터로feature vector 표현
  - 예) Iris 데이터에서 꽃받침의 길이, 꽃받침의 너비, 꽃잎의 길이, 꽃잎의 너비라는 4개의 특징이 각각 5.1, 3.5, 1.4, 0.2인 샘플

Input data

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 5.1 \\ 3.5 \\ 1.4 \\ 0.2 \end{pmatrix}$$

그림 3-2 iris의 세 가지 품종(왼쪽부터 Setosa, Versicolor, Virginica)

  - 여러 개의 특징 벡터를 첨자로 구분

$$\mathbf{x}_1 = \begin{pmatrix} 5.1 \\ 3.5 \\ 1.4 \\ 0.2 \end{pmatrix}, \mathbf{x}_2 = \begin{pmatrix} 4.9 \\ 3.0 \\ 1.4 \\ 0.2 \end{pmatrix}, \mathbf{x}_3 = \begin{pmatrix} 4.7 \\ 3.2 \\ 1.3 \\ 0.2 \end{pmatrix}, \cdots, \mathbf{x}_{150} = \begin{pmatrix} 5.9 \\ 3.0 \\ 5.1 \\ 1.8 \end{pmatrix}$$

Input data 1 ***Input data 2*** ***Input data 3***

벡터와 행렬

- iris 데이터
  - 특징이 4개이므로 4차원 특징 공간을 형성
  - 150개 샘플 각각은 4차원 특징 공간의 한 점

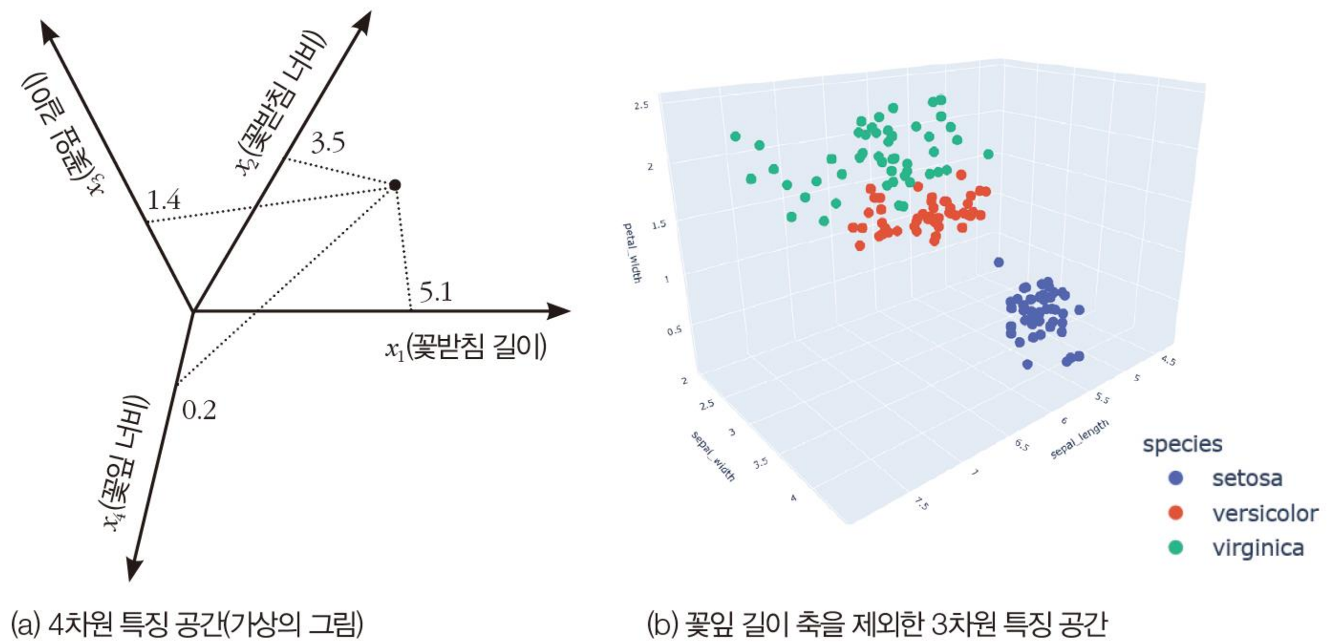


그림 3-5 iris 데이터를 특징 공간에 그리기

특징이 많을수록 사람이 다차원 특징 공간에서 분석하는 것은 불가능
=> Machine은 가능하다. (행렬 연산을 통해)

벡터와 행렬

- 행렬 (Matrix)
  - 여러 개의 벡터를 담음
  - 훈련집합 (Training set)을 담은 행렬
  - 예) Iris 데이터에 있는 150개의 샘플을 행렬 $\mathbf{X}$로 표현

$$\mathbf{X} = \begin{pmatrix} 5.1 & 3.5 & 1.4 & 0.2 \\ 4.9 & 3.0 & 1.4 & 0.2 \\ 4.7 & 3.2 & 1.3 & 0.2 \\ 4.6 & 3.1 & 1.5 & 0.2 \\ \vdots & \vdots & \vdots & \vdots \\ 6.2 & 3.4 & 5.4 & 2.3 \\ 5.9 & 3.0 & 5.1 & 1.8 \end{pmatrix} = \begin{pmatrix} x_{1,1} & x_{1,2} & x_{1,3} & x_{1,4} \\ x_{2,1} & x_{2,2} & x_{2,3} & x_{2,4} \\ x_{3,1} & x_{3,2} & x_{3,3} & x_{3,4} \\ x_{4,1} & x_{4,2} & x_{4,3} & x_{4,4} \\ \vdots & \vdots & \vdots & \vdots \\ x_{149,1} & x_{149,2} & x_{149,3} & x_{149,4} \\ x_{150,1} & x_{150,2} & x_{150,3} & x_{150,4} \end{pmatrix}$$

행row

열column

벡터와 행렬

지능형 예측·진단 연구실
Intelligent Prognostics & Diagnostics Lab.

- 행렬 $\mathbf{A}$의 전치행렬 $\mathbf{A}^{\mathrm{T}}$

Transpose matrix

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix}, \quad \mathbf{A}^{\mathrm{T}} = \begin{pmatrix} a_{11} & a_{21} & \cdots & a_{n1} \\ a_{12} & a_{22} & \cdots & a_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1m} & a_{2m} & \cdots & a_{nm} \end{pmatrix}$$

예를 들어, $\mathbf{A} = \begin{pmatrix} 3 & 4 & 1 \\ 0 & 5 & 2 \end{pmatrix}$라면 $\mathbf{A}^{\mathrm{T}} = \begin{pmatrix} 3 & 0 \\ 4 & 5 \\ 1 & 2 \end{pmatrix}$

- Iris의 설계 행렬을 전치 행렬 표기에 따라 표현하면,

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1^{\mathrm{T}} \\ \mathbf{x}_2^{\mathrm{T}} \\ \vdots \\ \mathbf{x}_{150}^{\mathrm{T}} \end{pmatrix}$$

벡터와 행렬

- 행렬 연산
  - 행렬 곱셈 $\mathbf{C} = \mathbf{AB}$, 이때 $c_{ij} = \sum_{k=1,s} a_{ik} b_{kj}$

$2*3$ 행렬 $\mathbf{A} = \begin{pmatrix} 3 & 4 & 1 \\ 0 & 5 & 2 \end{pmatrix}$와 $3*3$행렬 $\mathbf{B} = \begin{pmatrix} 2 & 0 & 1 \\ 1 & 0 & 5 \\ 4 & 5 & 1 \end{pmatrix}$을 곱하면 $2*3$ 행렬 $\mathbf{C} = \mathbf{AB} = \begin{pmatrix} 14 & 5 & 24 \\ 13 & 10 & 27 \end{pmatrix}$

  - 교환법칙 성립하지 않음: $\mathbf{AB} \neq \mathbf{BA}$
  - 분배법칙과 결합법칙 성립: $\mathbf{A}(\mathbf{B}+\mathbf{C}) = \mathbf{AB} + \mathbf{AC}$이고 $\mathbf{A}(\mathbf{BC}) = (\mathbf{AB})\mathbf{C}$

- 벡터의 내적

벡터의 내적 $\mathbf{a} \cdot \mathbf{b} = \mathbf{a}^{\mathrm{T}}\mathbf{b} = \sum_{k=1,d} a_k b_k$

$\mathbf{x}_1 = \begin{pmatrix} 5.1 \\ 3.5 \\ 1.4 \\ 0.2 \end{pmatrix}$와 $\mathbf{x}_2 = \begin{pmatrix} 4.9 \\ 3.0 \\ 1.4 \\ 0.2 \end{pmatrix}$의 내적 $\mathbf{x}_1 \cdot \mathbf{x}_2$는 37.49

회귀(Regression) 란?

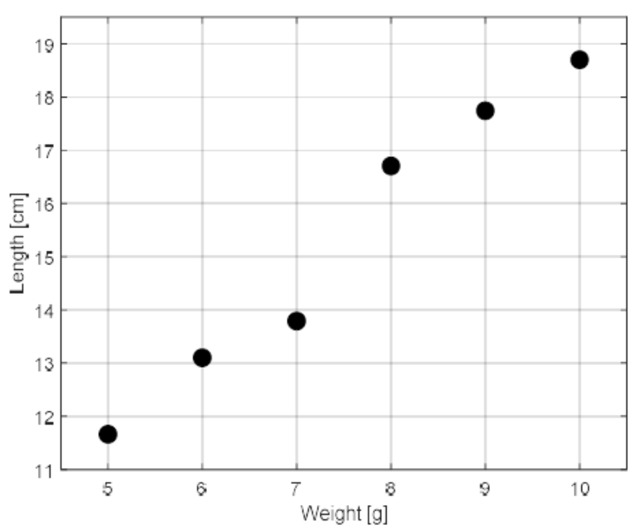
- 여러 개의 독립변수와 한 개의 종속변수 간의 상관관계를 모델링하는 기법
  - 독립변수 (Independent Variable)
    - 다른 변수의 변화와 관계없이 독립적으로 변하는 값
  - 종속변수 (Dependent Variable)
    - 다른 변수의 변화에 따라 변하는 값
  - 예) 추의 무게에 따라 늘어나는 용수철의 길이

독립변수 종속변수

훅(Hooke)의 법칙

$F = kx$

F: 용수철의 무게 [N]
k: 탄성 계수
x: 늘어난 길이 [m]



[그림 1] 데이터 그래프

탄성 계수를 구하고 싶다!!

How??

선형 회귀(Linear Regression)

- 선형 조합(Linear Combination)으로 모델링하는 회귀 기법

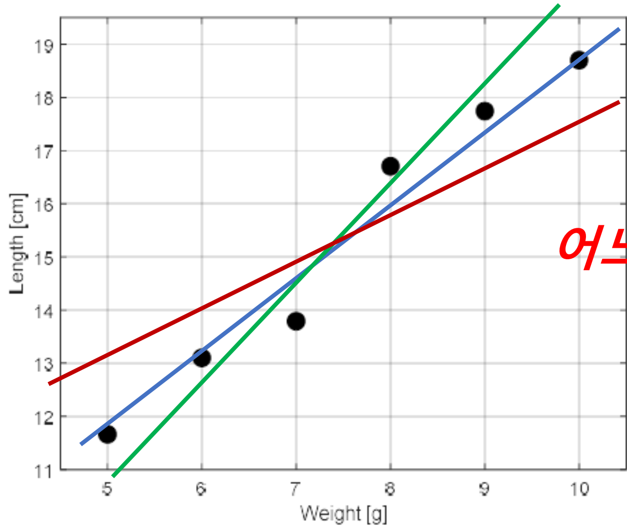
- 단순 선형 회귀 (Simple Linear Regression)
  - 하나의 x값 만으로 y값을 예측할 때

    Ex) $y = w_0 x + w_1$

$y = a_1 x + b_1$

$y = a_2 x + b_2$

$y = a_3 x + b_4$



어느 모델이 가장 적합한가?

[그림 1] 데이터 그래프

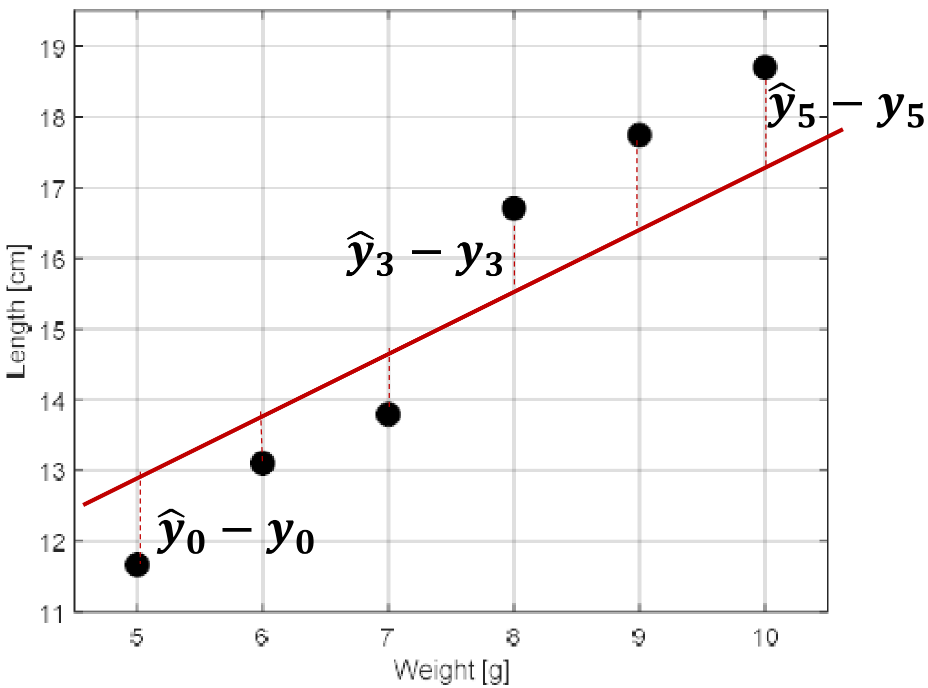
- 다중 선형 회귀 (Multiple Linear Regression)
  - y값을 예측하기 위해 여러가지 x가 사용될 때

    Ex) $y = w_0 x_0 + w_1 x_1 + w_2 x_2 + \cdots + w_{M-1} x_{M-1} + w_M$

비용 함수(Cost Function)

지능형 예측·진단 연구실
Intelligent Prognostics & Diagnostics Lab.

- **<u>모든 입력에 대해</u>** 예측값 ($\hat{y}$)과 실제값(y)의 차이를 나타낸 함수
- 머신러닝 회귀분석의 목표: Cost Function을 최소화 하는 것



[그림 1] 데이터 그래프

$$\sum_{n=0}^{5}\left(\hat{y}_n - y_n\right)$$

차이를 모두 양의 값으로 표현하고 싶다.

$$\sum_{n=0}^{N-1}\left(\hat{y}_n - y_n\right)^2$$

값이 너무 크니 평균을 취하자

$$\epsilon_{MSE} = \frac{1}{N}\sum_{n=0}^{N-1}\left(\hat{y}_n - y_n\right)^2$$

Mean Square Error
평균 제곱 오차

Cost Function의 최소화

(목표) 최대한 $\hat{y} \approx y$을 만족하는 적절한 w_0와 w_1을 찾는다.

(수정된 목표) 평균제곱오차를 최소화하는 w_0와 w_1를 찾는다.

수학적 표현

$$(w_0^*, w_1^*) = \boxed{\underset{(w_0, w_1)}{\arg\min}}\ \epsilon_{MSE}$$

Arguments of the minimum

Arguments: 함수의 입력

=> 즉, ϵ_{MSE} 를 최소로 만드는 **입력값 자체**를 의미
(w_0, w_1)

(수학적 목표)

$$(w_0^*, w_1^*) = \underset{(w_0, w_1)}{\arg\min}\ \frac{1}{N}\sum_{n=0}^{N-1}(w_0 x_n + w_1 - y_n)^2$$

Analytic Solution

지능형 예측·진단 연구실
Intelligent Prognostics & Diagnostics Lab.

$$\epsilon_{MSE}(w_0, w_1) = \frac{1}{N}\sum_{n=0}^{N-1}(w_0 x_n + w_1 - y_n)^2$$

$$\frac{\partial}{\partial w_0}\epsilon_{MSE}(w_0, w_1) = 0$$

$$\frac{\partial}{\partial w_1}\epsilon_{MSE}(w_0, w_1) = 0$$

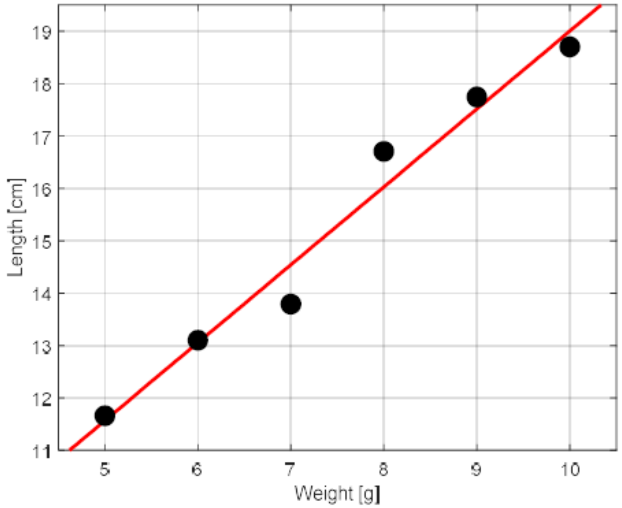
$$\Rightarrow$$

$$w_0^* = \frac{\frac{1}{N}\sum_{n=0}^{N-1} y_n\left(x_n - \frac{1}{N}\sum_{i=0}^{N-1} x_i\right)}{\frac{1}{N}\sum_{n=0}^{N-1} x_n^2 - \left(\frac{1}{N}\sum_{n=0}^{N-1} x_n\right)^2}$$

$$w_1^* = \frac{1}{N}\sum_{n=0}^{N-1}(y_n - w_0^* x_n)$$

Analytic Solution

데이터 번호	무게 (g)	늘어난 길이 (cm)
1	5	11.66
2	6	13.10
3	7	13.79
4	8	16.71
5	9	17.74
6	10	18.70



Analytic Solution

$$\hat{y} = 1.4875x + 4.1274$$

Python 실습 과제 #1

- 데이터 set “lin_regression_data01.csv” 는 추의 무게에 따른 늘어난 용수철의 길이에 대한 새로운 데이터일 때, (1열: 추의 무게 [g], 2열: 늘어난 길이 [cm])

1) 제공된 데이터 파일을 불러들여 x축은 추의 무게, y축은 늘어난 길이를 나타내는 2차원 평면에 각 데이터의 위치를 점으로 표시하라.
   - 필수요소: x축, y축 이름, grid, legend **→ 기본적으로 다음 실습에도 추가**
   - 결과물: 코드, 그래프

2) 제공된 데이터에 대한 선형회귀의 Analytic solution을 구하고, 그래프로 표시하라.
   - 필수요소: Analytic solution, x축, y축 이름, grid, legend
   - 결과물: 코드, 그래프

3) 2)에서 구한 선형회귀 모델의 평균제곱오차(MSE)를 구하라
   - 결과물: 코드, 평균제곱오차

[공통 사항] 실습 과제 수행 요령 (1)

- 각 주차에 해당하는 실습 과제에 대해 **하나의 보고서로 작성**
  - 예시) 이번 주 범위가 실습 과제 #1 & #2 => 실습과제 #1 & #2에 대해서 하나의 보고서로 작성

- 보고서 작성 형식은 **PPT** (캡쳐 도구 적극 활용), Template 신경 X
  - **업로드 파일 형식은 PDF (저장 시 PDF로 저장) => E-class 과제 Tap에 업로드**
  - **첫 페이지(표지)에 O주차, 실습과제 #O, 이름, 학번, 제출 날짜 기입**
  - 각 결과에 대한 해석은 **간단 명료**하게 서술

- 보고서: PPT로 작성 => PDF 변환 후 제출
  - **결과 그림, 결과에 대한 분석(개조식 서술)**

- 코드 : 한글/워드에 전체복사=> PDF 변환 후 제출
  - **주석 포함, 표절 검사용**

[공통 사항] 실습 과제 수행 요령 (2)

- 실습
  - 주차별 시작 점수 : 60 (만점 : 100)
  - 실습 당일 : Pass(+10), Non-pass(0), Drop(-20) **→ 조교에게 권한 부여**
  - 보고서 평가 : 1등(+30), 제출(0), 미흡(-10), 미제출(-30)

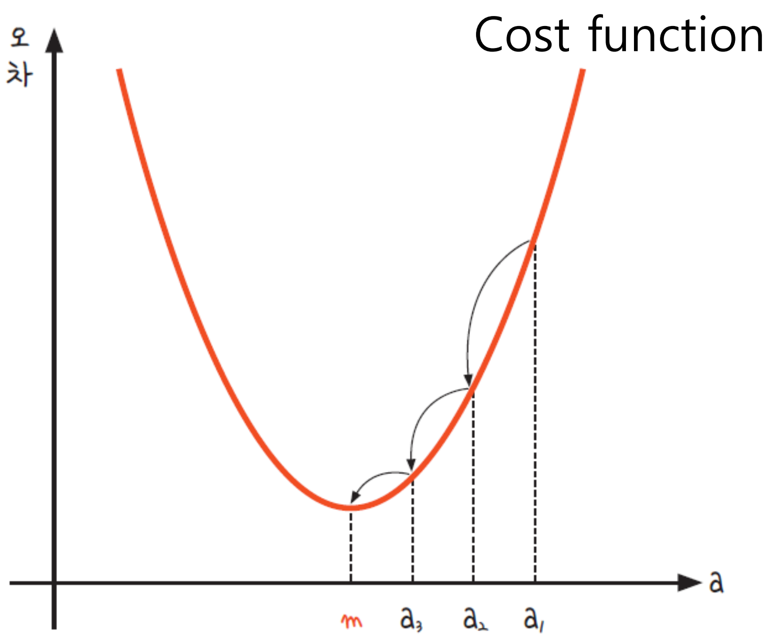
→ 주차별 1등 : 학기동안 1회(중복 수혜 불가)
→ 주차별 1등은 학기말 공개

- 보고서 평가 기준
  - 표지 작성 여부
  - 주석 처리에 따른 Code의 가독성
  - 보고서의 가독성 (Code & 그래프에 대한 해상도, X축 Y축 Label 등)
  - 수행 결과 및 결과에 대한 해석의 적절성
  - 학생 간 Code 표절 여부
    - **<u>구글링을 통한 소스코드 활용 시, 출처 반드시 표기</u>**

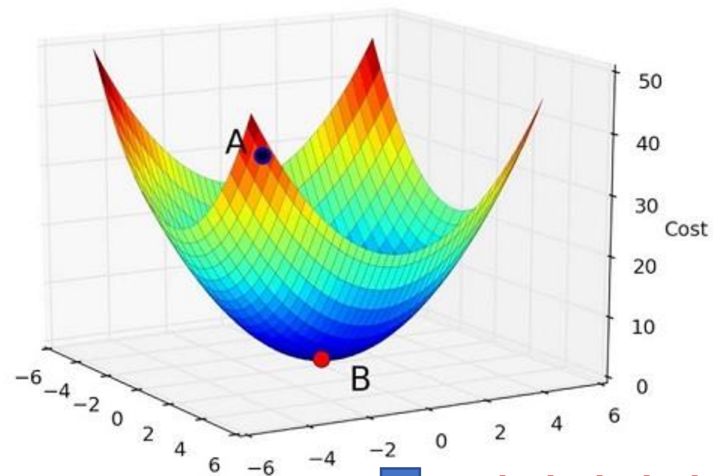
경사하강법(Gradient Decent Method)

지능형 예측·진단 연구실
Intelligent Prognostics & Diagnostics Lab.

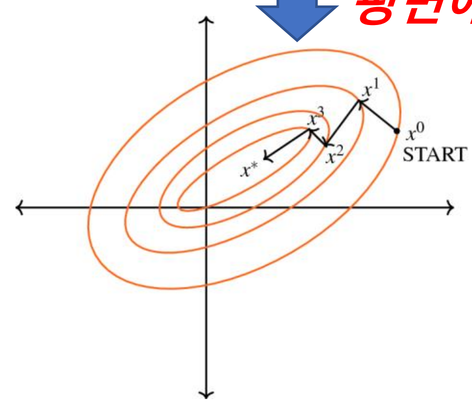
- 대부분의 Cost function은 Analytic solution을 구할 수 없음.
  - 머신러닝에서는 수치적으로 접근하는 방법을 사용



독립변수가 1개일 때



평면에서의 표현



독립변수가 2개일 때

[보충] 미분

- 미분에 의한 최적화
  - 미분의 정의

$$f'(x) = \lim_{\Delta x \to 0} \frac{f(x+\Delta x) - f(x)}{\Delta x}, \qquad f''(x) = \lim_{\Delta x \to 0} \frac{f'(x+\Delta x) - f'(x)}{\Delta x}$$

  - 1차 도함수 $f'(x)$는 함수의 기울기, 즉 값이 커지는 방향을 지시함
  - 따라서 $-f'(x)$ 방향에 목적함수의 최저점이 존재

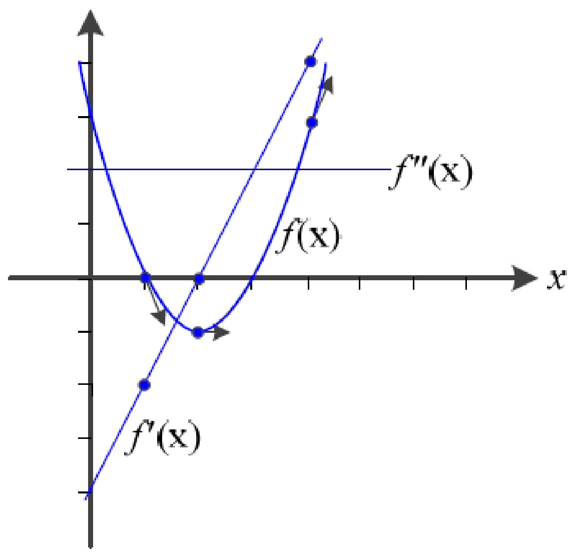


그림 2-24 간단한 미분 예제

$$y = f(x) = x^2 - 4x + 3$$

$$y' = f'(x) = 2x - 4$$

[보충] 미분

- 편미분
  - **<u>독립변수가 여러 개인 함수</u>**에 대한 미분 개념을 적용할 때 사용
  - 미분값이 이루는 벡터를 그레이디언트(Gradient)라 부름
  - 여러 가지 표기: $\nabla f, \frac{\partial f}{\partial \mathbf{x}}, \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}\right)^{\mathrm{T}}$
  - 예)

$$\left.\begin{aligned} f(\mathbf{x}) &= f(x_1, x_2) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right)x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2 \\ \nabla f &= f'(\mathbf{x}) = \frac{\partial f}{\partial \mathbf{x}} = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}\right)^{\mathrm{T}} = (2x_1^5 - 8.4x_1^3 + 8x_1 + x_2, 16x_2^3 - 8x_2 + x_1)^{\mathrm{T}} \end{aligned}\right\}$$

- 머신러닝에서의 편미분
  - 매개변수 집합 Θ에 독립변수(특징)들이 있으므로 편미분을 사용

[보충] 미분

- 연쇄법칙
  - 합성함수 $f(x) = g(h(x))$의 미분

$$\left.\begin{aligned} f'(x) &= g'\big(h(x)\big)h'(x) \\ f'(x) &= g'\big(h(i(x))\big)h'(i(x))i'(x) \end{aligned}\right\}$$

  - 예) $f(x) = 3(2x^2-1)^2 - 2(2x^2-1) + 5$일 때 $h(x) = 2x^2 - 1$로 두면,

$$f'(x) = \underbrace{(3 * 2(2x^2-1) - 2)}_{g'(h(x))}\underbrace{(2 * 2x)}_{h'(x)} = 48x^3 - 32x$$

- 다층 퍼셉트론은 합성함수
  - $\frac{\partial o_i}{\partial u_{23}^1}$를 계산할 때 연쇄법칙 적용
  - 인공신경망 파트에서 자세히 설명

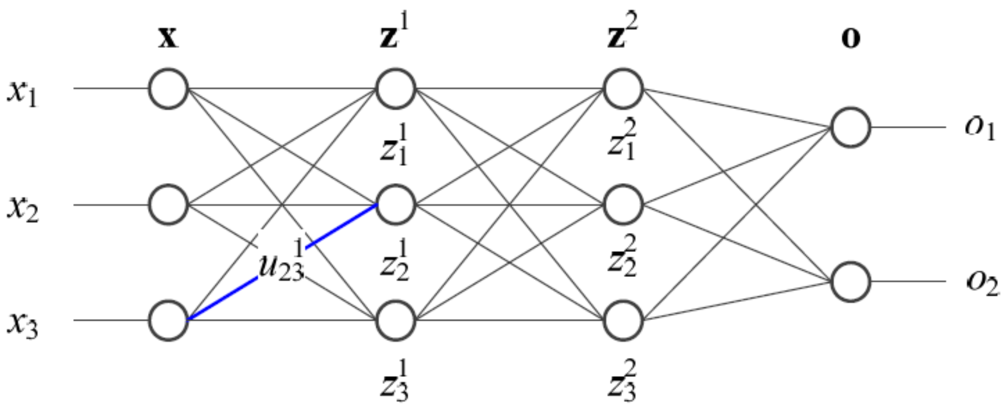
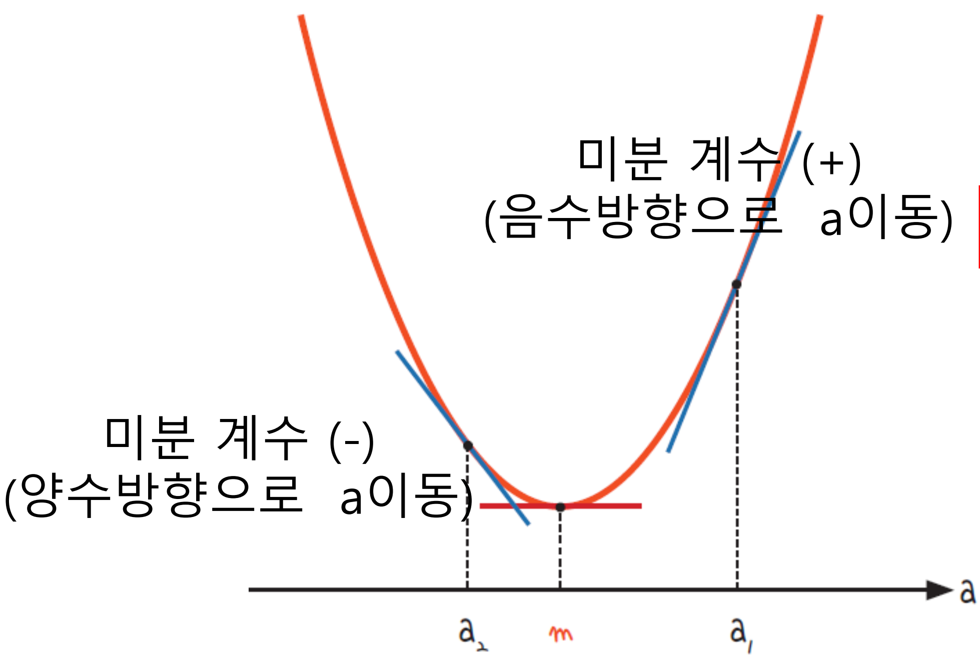


그림 2-26 다층 퍼셉트론은 합성함수

경사하강법(Gradient Decent Method)

- 최소값에서 접선의 기울기에 해당하는 미분 계수는 0
  - ➢ **<u>여러 번의 반복 (iteration)을 통해 미분 계수가 0이 되는 지점을 찾는다.</u>**

- 미분 계수(Gradient)는 언제나 현재 위치에서 함수값이 커지는 방향을 가리키므로, **미분 계수의 반대 방향으로 이동**하면 최솟값을 찾을 수 있음
  - ➢ 즉 Gradient에 마이너스(-)를 취하여 더한다.



Learning Rate

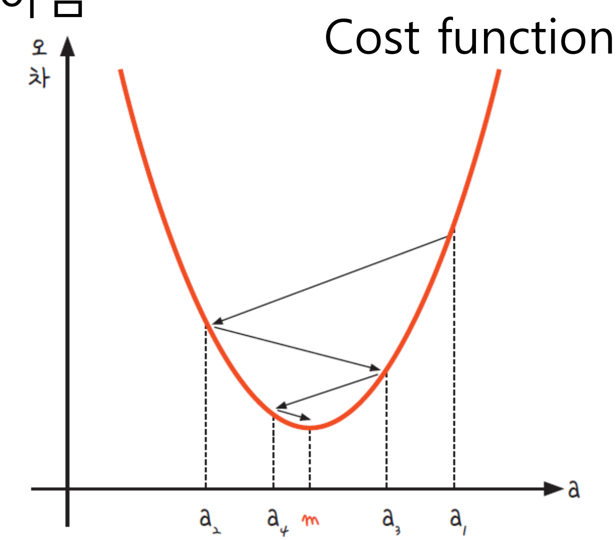
$$\overset{\text{New}}{\boxed{w_0[t+1]}} = \overset{\text{Old}}{\boxed{w_0[t]}} - \boxed{\alpha}\frac{\partial}{\partial w_0}\epsilon_{MSE}(\overset{\text{Old}}{\boxed{w_0, w_1}})$$

$$w_1[t+1] = w_1[t] - \alpha\frac{\partial}{\partial w_1}\epsilon_{MSE}(w_1, w_2)$$

경사하강법(Gradient Decent Method)

$$\overset{\text{New}}{\boxed{w_0[t+1]}} = \overset{\text{Old}}{\boxed{w_0[t]}} - \overset{\text{Learning Rate}}{\boxed{\alpha}} \frac{\partial}{\partial w_0}\epsilon_{MSE}\big(\overset{\text{Old}}{\boxed{w_0, w_1}}\big)$$

1) w_0와 w_1에 대해서 임의의 값을 설정 => Initialization(초기화)
2) t 지점에서 Cost Function에 대한 미분 계수(기울기)를 구함
3) 기울기의 반대 방향으로 point를 $\alpha \times$ 기울기의 크기 만큼 이동시킴. (t+1 로 이동)
4) Stop 조건 확인
   4-1) Stop 조건에 만족할 경우: 현재의 값을 최종 값으로 선택
   4-2) Stop 조건에 만족하지 않을 경우, step 2)로 돌아감



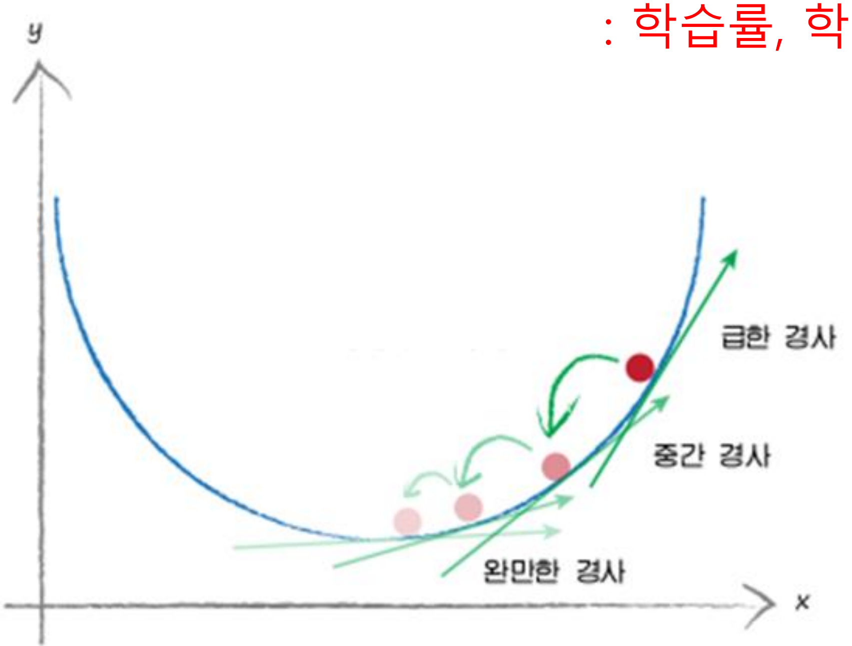
❖ Stop 조건
- 더 이상 변수가 변하지 않는다. (미분 값이 0)
- 변화량이 매우 작다. (미분 값이 매우 작다)

학습률(Learning Rate)

- 학습률 (Learning Rate)
  - 경사 하강법에서 미분 계수 앞에 곱해져 이동 거리를 결정하는 변수
  - 미분 계수의 크기는 최소값에서 멀어질수록 커지는 성질을 가짐
    - ➢ 학습률을 곱하지 않을 경우 발산 가능
  - 적절한 값을 찾아야 머신러닝의 학습 성공이 가능
    - ➢ 하이퍼 파라미터(Hyper-parameter)의 최적화

머신러닝 설계자가 직접 세팅하는 값

: 학습률, 학습 반복 횟수, Stop 조건 등



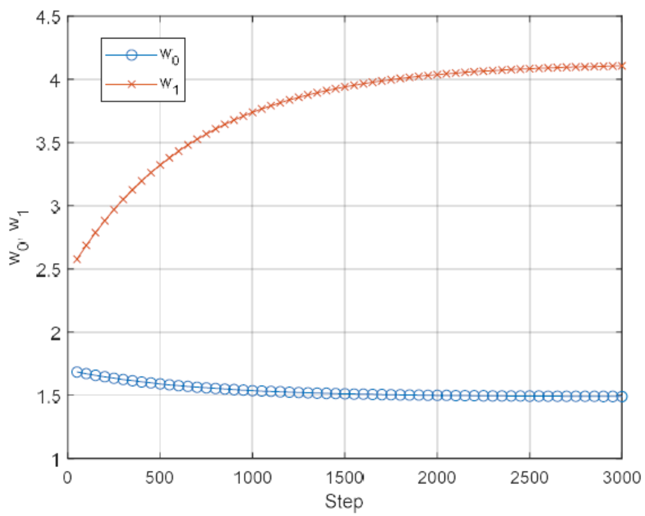
경사하강법의 선형회귀 적용

지능형 예측·진단 연구실
Intelligent Prognostics & Diagnostics Lab.

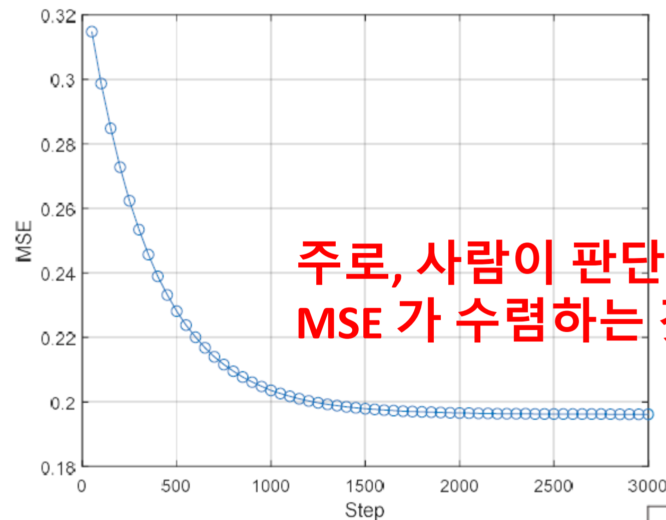
- 초기값: $w_0[0] = 2$와 $w_1[0] = 2.5$
- Learning Rate: $\alpha = 0.015$
- 3000회 반복 시: $w_0 = 1.4902, \ w_1 = 4.1063$

Optimal Solution

$$\hat{y} = 1.4875x + 4.1274$$

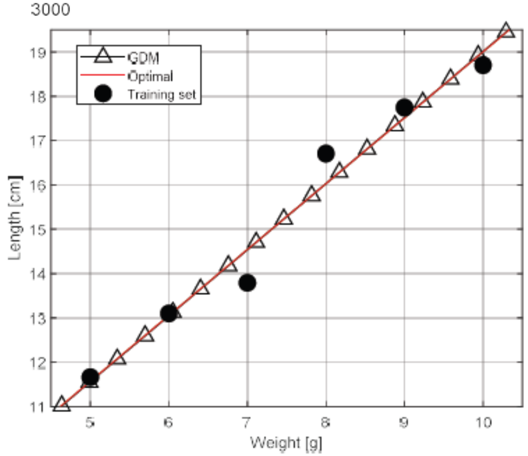


[그림 3] 경사하강법의 단계별 매개변수 값의 변화, $\alpha = 0.01$



[그림 4] 단계별 평균제곱오차의 변화

주로, 사람이 판단 할 때는
MSE 가 수렴하는 것을 확인



Python 실습 과제 #2

- 실습 과제 #1에서의 데이터 set “lin_regression_data01.csv” 에 대해서

1) 앞에서 배운 경사하강법을 사용자 지정 함수로 구현하라 (**1차 다항식 모델 한정, Stop 조건은 제외 => 반복횟수 지정**)
    - 결과물: 코드, 그래프

2) 구현한 경사하강법 함수를 이용해 Optimal Solution (weights, $\boldsymbol{w}$)을 구하고, 학습 진행(epoch)에 따른 $\boldsymbol{w}$ 와 MSE에 대한 그래프(교재 7쪽의 그림 3,4 참조)를 그려라.
    - 결과물: 코드 및 그래프
    - 필수요소: 각 그래프별 학습률, 초기값, 반복횟수 등에 따른 분석 내용

3) 선형회귀모델로 구한 Optimal Solution과 데이터 set을 하나의 그래프에 표시하고, 실습과제 #1-2)의 결과와 비교하라.
    - 결과물: 코드 및 비교 분석내용

[참고] Error에 대한 편미분

지능형 예측·진단 연구실
Intelligent Prognostics & Diagnostics Lab.

$$\epsilon_{MSE} = \frac{1}{N}\sum_{n=0}^{N-1}(\boxed{\hat{y}_n} - \boxed{y_n})^2$$

예측값 참값 => 고정값 => 상수

- **1차 다항식 모델**

$$\hat{y}_n = \mathrm{w}_0 x + \mathrm{w}_1$$

$$\frac{\partial}{\partial w_0}\epsilon_{MSE}(w_0, w_1) = \frac{1}{N}\sum_{n=0}^{N-1}(\hat{y}_n - \mathrm{y}_\mathrm{n})\, x$$

$$\frac{\partial}{\partial w_1}\epsilon_{MSE}(w_0, w_1) = \frac{1}{N}\sum_{n=0}^{N-1}(\hat{y}_n - \mathrm{y}_\mathrm{n})$$

다차원 데이터로의 확장

지능형 예측·진단 연구실
Intelligent Prognostics & Diagnostics Lab.

단일 변수 대한 Cost Function: $\epsilon_{MSE} = \frac{1}{N}\sum_{n=0}^{N-1}(w_0 \boxed{x_n} + w_1 - y_n)^2$

단일 변수 (용수철 무게)

변수가 여러 개라면?

$$\epsilon_{MSE} = \frac{1}{N}\sum_{n=0}^{N-1}(\hat{y}_n - y_n)^2$$

$$= \frac{1}{N}\sum_{n=0}^{N-1}(w_0 \underbrace{\overbrace{\boxed{x_{0,n}}}^{\text{Input}}}_{\text{속도}} + w_1 \underbrace{\overbrace{\boxed{x_{1,n}}}^{\text{Input}}}_{\text{질량}} + \cdots + w_{M-1} \underbrace{\overbrace{\boxed{x_{M-1,n}}}^{\text{Input}}}_{\text{온도}} + w_M - \underbrace{\overbrace{\boxed{y_n}}^{\text{Output}}}_{\text{거리}})^2$$

n: 측정한 데이터 수에 대한 Index
M: 입력 데이터 Type 수 (데이터 차원 수)

Notation에 대한 개념 확실하게 이해필요

데이터 번호	입력	출력
0	$x_{0,0}, x_{1,0}, \cdots, x_{m,0}, \cdots, x_{M-1,0}$	y_0
1	$x_{0,1}, x_{1,1}, \cdots, x_{m,1}, \cdots, x_{M-1,1}$	y_1
$\vdots$	$\vdots$	$\vdots$
n	$x_{0,n}, x_{1,n}, \cdots, x_{m,n}, \cdots, x_{M-1,n}$	y_n
$\vdots$	$\vdots$	$\vdots$
$N-1$	$x_{0,N-1}, x_{1,N-1}, \cdots, x_{m,N-1}, \cdots, x_{M-1,N-1}$	y_{N-1}

다차원 데이터를 위한 벡터 표현

지능형 예측·진단 연구실
Intelligent Prognostics & Diagnostics Lab.

$$\epsilon_{MSE} = \frac{1}{N}\sum_{n=0}^{N-1}\left(\hat{y}_n - y_n\right)^2$$
$$= \frac{1}{N}\sum_{n=0}^{N-1}\left(w_0 x_{0,n} + w_1 x_{1,n} + \cdots + w_{M-1}x_{M-1,n} + w_M - y_n\right)^2$$

$$\hat{y} = w_0 x_0 + w_1 x_1 + \cdots + w_{M-1}x_{M-1} + \boxed{w_M x_M}$$

핵심 트릭

복잡했던 식을 벡터의 곱으로 간단하게 표현

$$\hat{y} = \underbrace{\boxed{\begin{bmatrix} w_0 & w_2 & \cdots & w_{M-1} & w_M \end{bmatrix}}}_{\boldsymbol{w}^T} \underbrace{\boxed{\begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{M-1} \\ x_M \end{bmatrix}}}_{\boldsymbol{x}} = w^T x$$

$$\epsilon_{MSE} = \frac{1}{N}\sum_{n=0}^{N-1}\left(\hat{y}_n - y_n\right)^2 = \frac{1}{N}\sum_{n=0}^{N-1}\left(w^T x_n - y_n\right)^2$$

다차원 입력데이터에 대한 Cost Function

=> Cost Function 이 최소가 되는 $\boldsymbol{w} = [w_0\ w_1\ w_2\ \cdots w_M]$ *가 Optimal Solution*

다차원 데이터에 대한 Analytic Solution

지능형 예측·진단 연구실
Intelligent Prognostics & Diagnostics Lab.

$$\epsilon_{MSE} = \frac{1}{N}\sum_{n=0}^{N-1}\left(\hat{y}_n - y_n\right)^2 = \frac{1}{N}\sum_{n=0}^{N-1}\left(w^T x_n - y_n\right)^2$$

편미분을 통한 해석해 구하기

$$\frac{\partial}{\partial w_m}\epsilon_{MSE} = \frac{2}{N}\sum_{n=0}^{N-1}\left(w^T x_n - y_n\right)\frac{\partial}{\partial w_m}w^T x_n$$

$$= \frac{2}{N}\sum_{n=0}^{N-1}\left(w^T x_n - y_n\right)x_{m,n}$$ = 0 이 되는 해

$m=0$일 때, $\sum_{n=0}^{N-1}\left(w^T x_n - y_n\right)x_{0,n} = 0$

$m=1$일 때, $\sum_{n=0}^{N-1}\left(w^T x_n - y_n\right)x_{1,n} = 0$

$\vdots$

$m=M-1$일 때, $\sum_{n=0}^{N-1}\left(w^T x_n - y_n\right)x_{M-1,n} = 0$

$m=M$일 때, $\sum_{n=0}^{N-1}\left(w^T x_n - y_n\right)x_{M,n} = 0$

$$\sum_{n=0}^{N-1}\left(w^T x_n - y_n\right)x_n^T = [0\ 0 \cdots 0\ 0]$$

다차원 데이터에 대한 Optimal Solution

$$\sum_{n=0}^{N-1}\left(w^T x_n - y_n\right)x_n^T = [0\ 0 \cdots 0\ 0]$$

데이터 수에 대한 Summation 계산을 행렬 계산으로 한번에 계산 가능해짐

$$w^T \mathrm{X}^\mathrm{T}\mathrm{X} - y^T \mathrm{X} = [0\ 0 \cdots 0\ 0]$$

$$\begin{aligned}(w^T \mathrm{X}^\mathrm{T}\mathrm{X} - y^T \mathrm{X})^T &= (w^T \mathrm{X}^\mathrm{T}\mathrm{X})^T - (y^T \mathrm{X})^T \\ &= (\mathrm{X}^\mathrm{T}\mathrm{X})^T w - \mathrm{X}^\mathrm{T}\mathrm{y} \\ &= \mathrm{X}^\mathrm{T}\mathrm{X}w - \mathrm{X}^\mathrm{T}\mathrm{y} = [0\ 0 \cdots 0\ 0]\end{aligned}$$

$$\mathrm{X}^\mathrm{T}\mathrm{X}w = \mathrm{X}^\mathrm{T}\mathrm{y}$$

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

역행렬이 항상 존재한다는 것을 보장 할 수 없음

즉, 수치적 접근을 통한 최적해(Optimal Solution)을 구하는 것이 일반적

다차원 데이터에서의 경사하강법

지능형 예측·진단 연구실
Intelligent Prognostics & Diagnostics Lab.

- **비용 함수(Cost Function)**
  - $cost(a,b) = \epsilon_{MSE}(a,b) = \frac{1}{N}\sum_{n=0}^{N-1}(ax_i + b - y_i)^2$

  $\rightarrow \boldsymbol{cost(W) = \epsilon_{MSE}(W) = \frac{1}{N}\sum_{n=0}^{N-1}\left(W^T X^{(i)} - Y^{(i)}\right)^2}$

- **경사하강법(Gradient Decent Method)**
  - $a_{new} = a_{old} - \alpha \frac{\partial cost(a,b)}{\partial a}\Big|_{(a_{old},b_{old})}$
  - $b_{new} = b_{old} - \alpha \frac{\partial cost(a,b)}{\partial b}\Big|_{(a_{old},b_{old})}$

  $\rightarrow \boldsymbol{W_{new} = W_{old} - \alpha \frac{\partial cost(W)}{\partial W}\Big|_{W_{old}}}$

Python 실습 과제 #3

- "lin_regression_data_02.csv"는 X0, X1, y로 이루어진 3차원 데이터일 때,

1) 제공된 데이터 파일을 불러들여 x축은 X0, y축은 X1, z축은 y를 나타내는 3차원 평면에 각 데이터의 위치를 점으로 표시하라.
   - 필수요소 : x축, y축, z축 이름, grid, legend
   - 결과물: 코드, 그래프

2) 주어진 X 데이터에 더미(dummy) 데이터를 추가하여 행렬을 생성하고, 초기 가중치(weights) 값을 이용하여 예측한 $\hat{y}$을 3차원 축에 평면으로 나타내라.
   - 결과물: 코드 및 그래프 초기 가중치(weights) 지정은 random 함수 사용

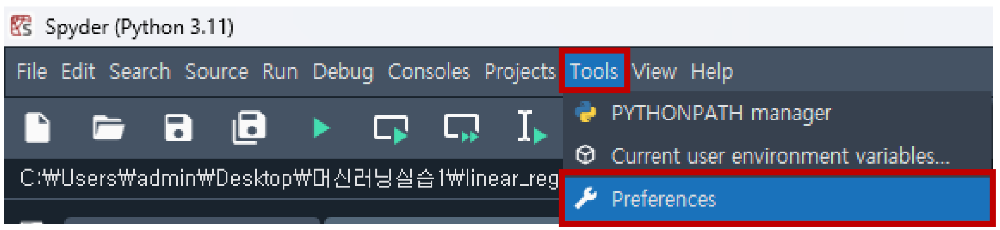
3) 실습 #2에서 구현한 경사하강법 함수를 다차원 데이터 입력이 가능하도록 변경해 Optimal Solution (weights, $\boldsymbol{w}$)을 구하고, 학습 진행(epoch)에 따른 $\boldsymbol{w}$와 MSE에 대한 그래프(본 ppt 자료 23page 참고)를 그려라.
   - 필수요소: 각 그래프 별 학습률, 초기값, 반복횟수 등에 따른 분석 내용
   - 결과물: 코드 및 그래프

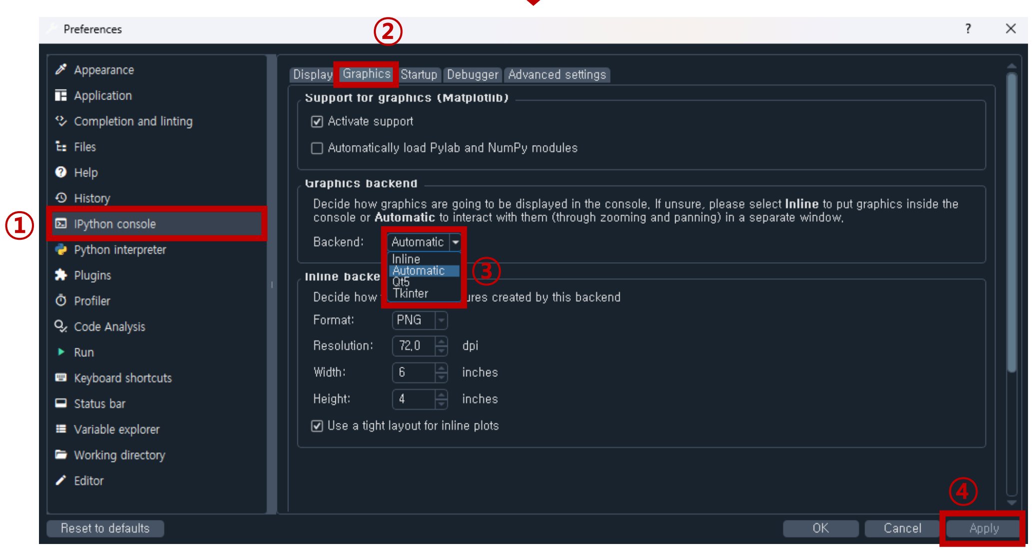
4) 3차원 축에 y와 Optimal Solution을 이용한 $\hat{y}$을 점으로 표시하고, $\hat{y}$은 3차원 축에 평면으로 나타내라. 보고서에 최소 3개이상의 각도로 그래프를 회전하여 분석할 것.
   - 결과물: 코드 및 그래프

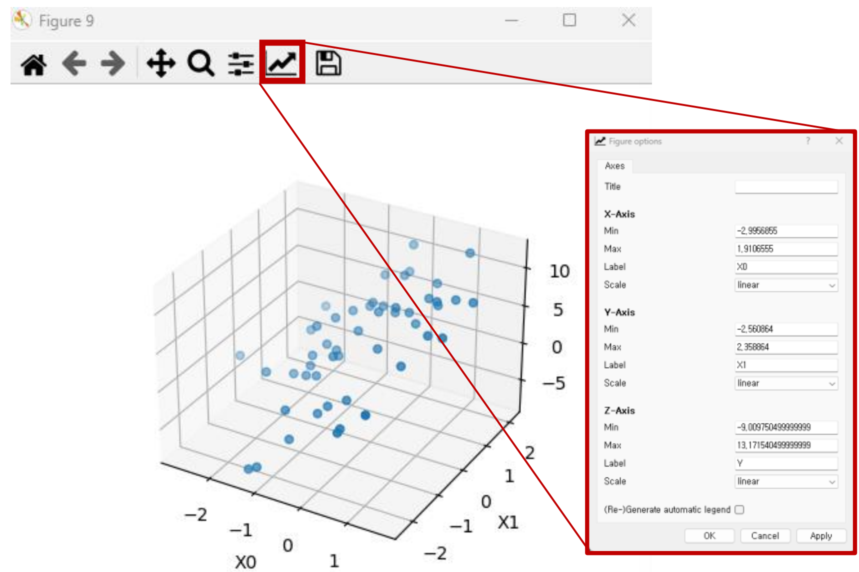
참고 : https://matplotlib.org/stable/gallery/mplot3d/scatter3d.html

[참고] 3차원 그래프 그리기 설정

- Spyder 3차원 그래프 환경 세팅 방법
  1) Tools => Preferences 클릭
  2) IPython console => Graphics => Backend를 Automatic으로 설정 후 apply
  3) Figure 옵션을 통해 그래프 제어





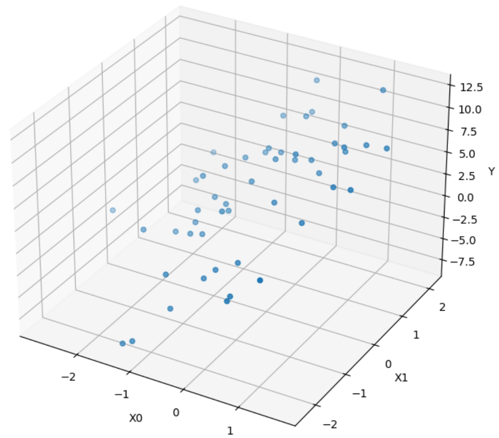


< Figure 창>　　< 옵션창 >

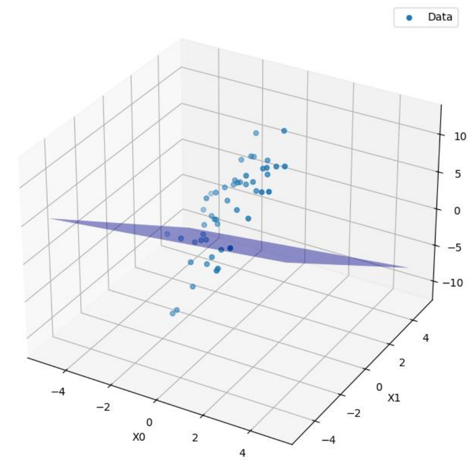
[참고] 실습 결과

지능형 예측·진단 연구실
Intelligent Prognostics & Diagnostics Lab.

- # 3-1, 3-2 결과

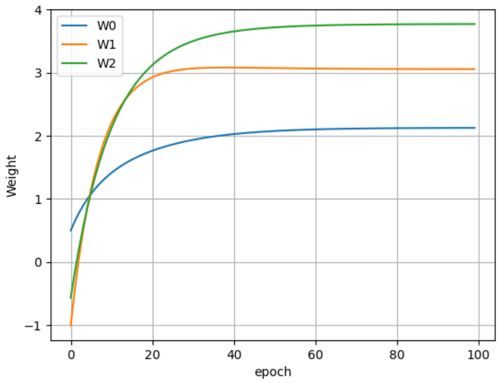


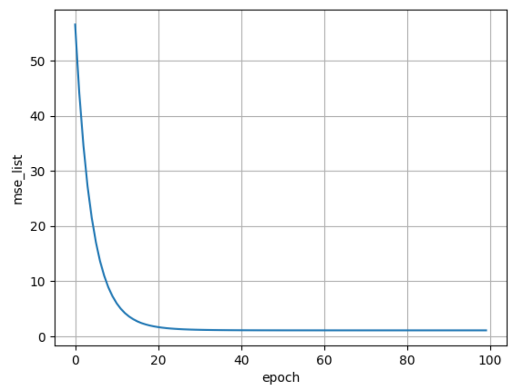
3-1(원 데이터)



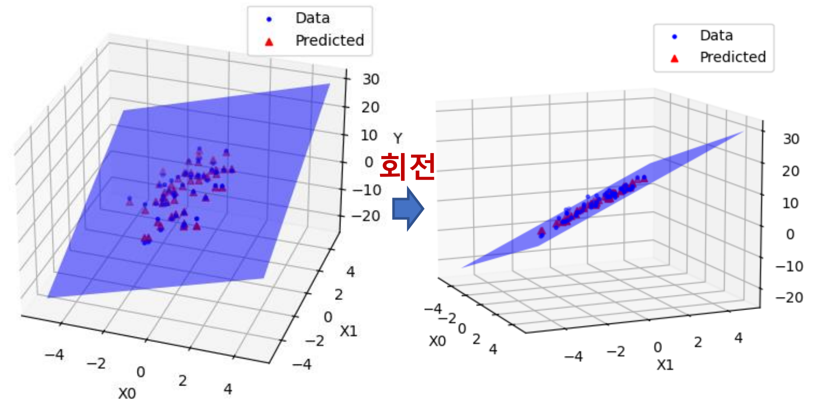
3-2 (원 데이터 + 초기화 가중치)

- # 3-3, 3-4 결과





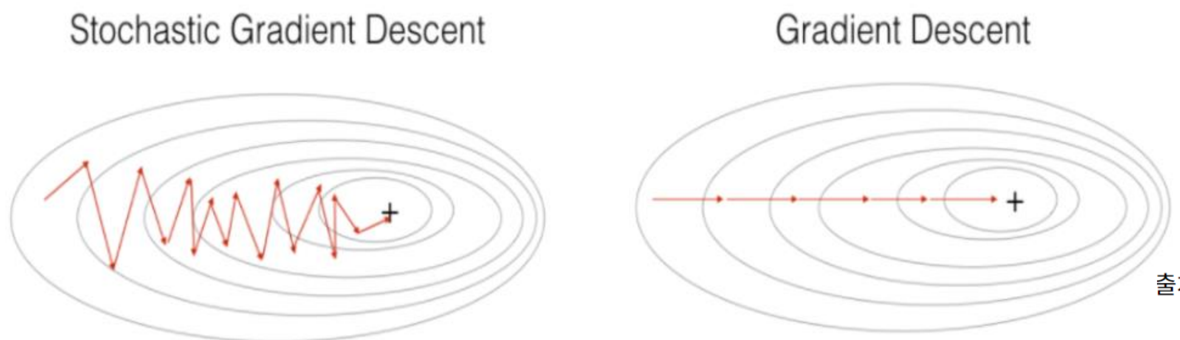
3-3 (weight, mse 그래프)



3-4 (3차원 plot)

확률적 경사하강법(SGD)

- Batch: 경사하강법에서 기울기를 계산하는데 사용하는 데이터의 갯수
- 확률적 경사하강법(Stochastic Gradient Descent, SGD)

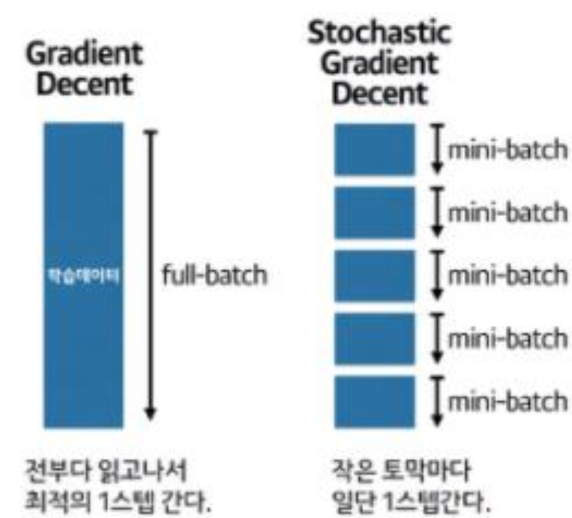


출처: https://engmrk.com/mini-batch-gd/

Batch Size: 1

Batch Size: N (데이터 전체)

- Mini Batch :
  일부 모음으로 SGD를 적용하는 방법

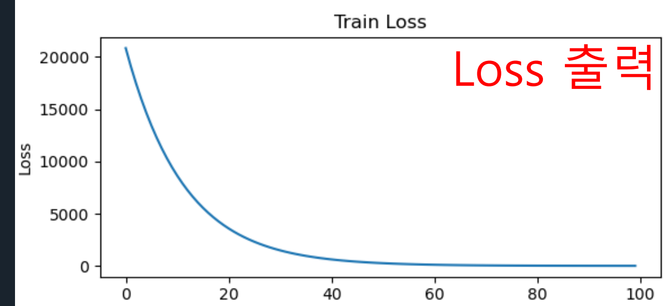


출처 : https://www.slideshare.net/yongho/ss-79607172

[참고] Keras로 다중 선형회귀 구현하기

- **Keras로 다중 선형회귀 구현하기**

```
# Python 실습 과제 #3

import numpy as np

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras import optimizers
import matplotlib.pyplot as plt

x_data = np.array([ [73,80,75],
                   [93,88,93],
                   [89,91,90],
                   [80,80,80],
                   [96,98,100],
                   [73,66,70], ])
y_data = np.array([72,88,92,81,100,71])

model = Sequential()
model.add(Dense(1, input_dim = 3, activation='linear'))
sgd = optimizers.SGD(learning_rate = )
model.compile(loss='mse', optimizer=sgd, metrics=['mse'])

history = model.fit(x_data, y_data, batch_size= , epochs= , verbose= )

x_test = np.array([[90,88,93], [70,70,70]])
```



1. Learning_rate에 따른 학습 상태 관측
2. batch_size에 따른 학습 상태 관측
3. epochs, verbose의 의미와 값에 따른 변화
4. Test set에 대한 예측 결과 출력
5. 학습 완료된 상태에서의 Weight 출력
6. 학습 과정에 대한 MSE 출력 (Loss 출력)

[참고] sklearn 활용: 다중 선형회귀 (1)

지능형 예측·진단 연구실
Intelligent Prognostics & Diagnostics Lab.

- **맨해튼 집값 예측하기**

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```python
url = "https://raw.githubusercontent.com/Codecademy/datasets/master/streeteasy/manhattan.csv"
manhattan = pd.read_csv(url)
```

맨해튼 집값 데이터 load

Name	Type	Size
manhattan	DataFrame	(3539, 18)

Index	rental_id	rent	bedrooms	bathrooms	size_sqft	n_to_subw	floor	lding_age	no_fee	as_roofdec	washer_d	as_doorma	ias_elevato	s_dishwast	has_patio	has_gym	neighborhood	borough
0	1545	2550	0	1	480	9	2	17	1	1	0	0	1	1	0	1	Upper East Side	Manhattan
1	2472	11500	2	2	2000	4	1	96	0	0	0	0	0	0	0	0	Greenwich Village	Manhattan
2	2919	4500	1	1	916	2	51	29	0	1	0	1	1	1	0	0	Midtown	Manhattan
3	2790	4795	1	1	975	3	8	31	0	0	0	1	1	1	0	1	Greenwich Village	Manhattan
4	3946	17500	2	2	4800	3	4	136	0	0	0	1	1	1	0	1	Soho	Manhattan
5	10817	3800	3	2	1100	3	5	101	0	0	0	0	0	0	0	0	Central Harlem	Manhattan
6	9077	1995	0	0	600	6	1	115	0	0	0	0	0	0	0	0	Midtown East	Manhattan
7	5150	2995	0	1	579	6	21	33	0	0	0	0	0	0	0	0	Battery Park City	Manhattan
8	9507	15000	2	2	1715	0	30	2	0	0	0	0	0	0	0	0	Flatiron	Manhattan
9	1437	4650	1	1	915	5	5	106	0	0	0	0	0	0	0	0	Upper East Side	Manhattan
10	404	2950	1	1	550	43	17	14	1	1	0	1	1	0	0	0	Upper East Side	Manhattan
11	8293	6920	3	2	1439	7	9	39	1	0	0	0	0	0	0	0	Midtown East	Manhattan
12	6594	4875	1	1	900	1	14	52	1	0	1	1	1	1	0	1	East Village	Manhattan

[참고] sklearn 활용: 다중 선형회귀 (2)

지능형 예측·진단 연구실
Intelligent Prognostics & Diagnostics Lab.

- **맨해튼 집값 예측하기**

```
x = manhattan[['bedrooms', 'bathrooms', 'size_sqft', 'min_to_subway', 'floor',
'building_age_yrs', 'no_fee', 'has_roofdeck', 'has_washer_dryer', 'has_doorman',
'has_elevator', 'has_dishwasher', 'has_patio', 'has_gym']]
y = manhattan[['rent']]
```

다차원 입력

출력: 집값

사이킷런 라이브러리 활용

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x,y, train_size = 0.7, test_size = 0.3)
```

Train data : Test data = 70 : 30 으로 분할

```
from sklearn.linear_model import LinearRegression
mlr = LinearRegression()
mlr.fit(x_train, y_train) # learning 실행
mlr.coef_ # weight 확인
mlr.intercept_ # bias 확인
```

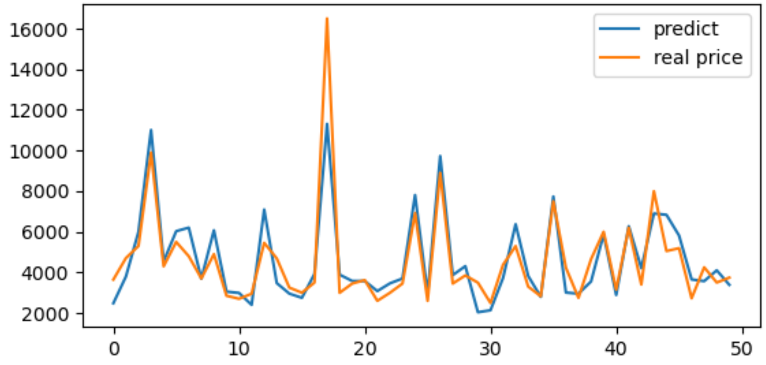
SVD(특이값 분해) 기법으로 추정

[참고] sklearn 활용: 다중 선형회귀 (3)

- **맨해튼 집값 예측하기**

Test 데이터 50개에 대해서 예측값 출력

```
plt.plot(mlr.predict(x_test[:50]))
plt.plot(y_test[:50].values.reshape(-1, 1))
plt.legend(["predict","real price"])
```




```
my_apt = [[1,1,600,16,1,8,1,0,1,0,0,1,1,0]]
mlr.predict(my_apt)  # 임의의 값에 대한 예측
```

```
array([[2951.70009882]])
```

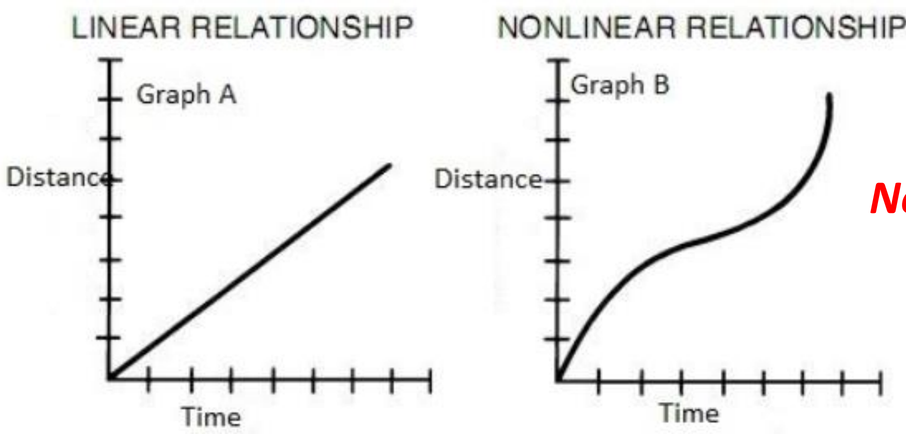
```
mlr.score(x_train, y_train) # 성능 확인
```

```
0.7881931858162815
```

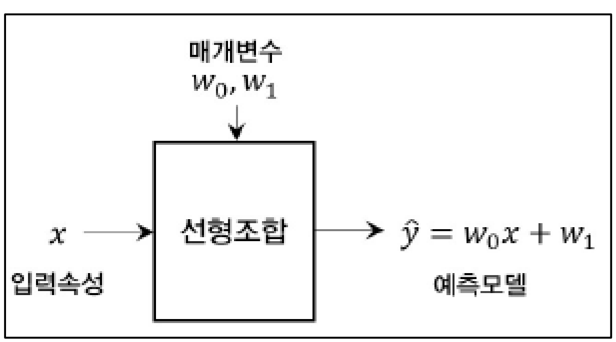
1에 가까울 수록 완벽하게 예측

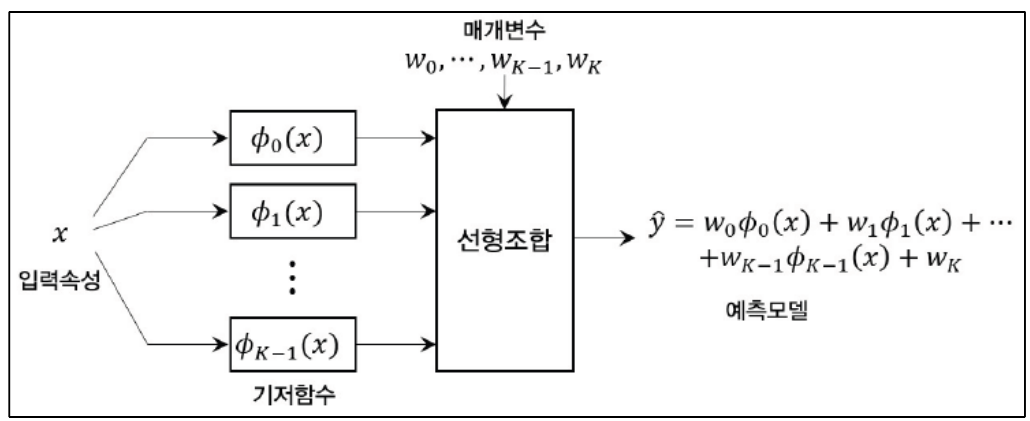
선형 기저함수 모델

- **선형 기저함수 모델(Linear Basis Function Model)**



Non-Linear한 특성을 가지는 함수에 대한 표현을 선형함수의 조합으로 표현이 가능한가?

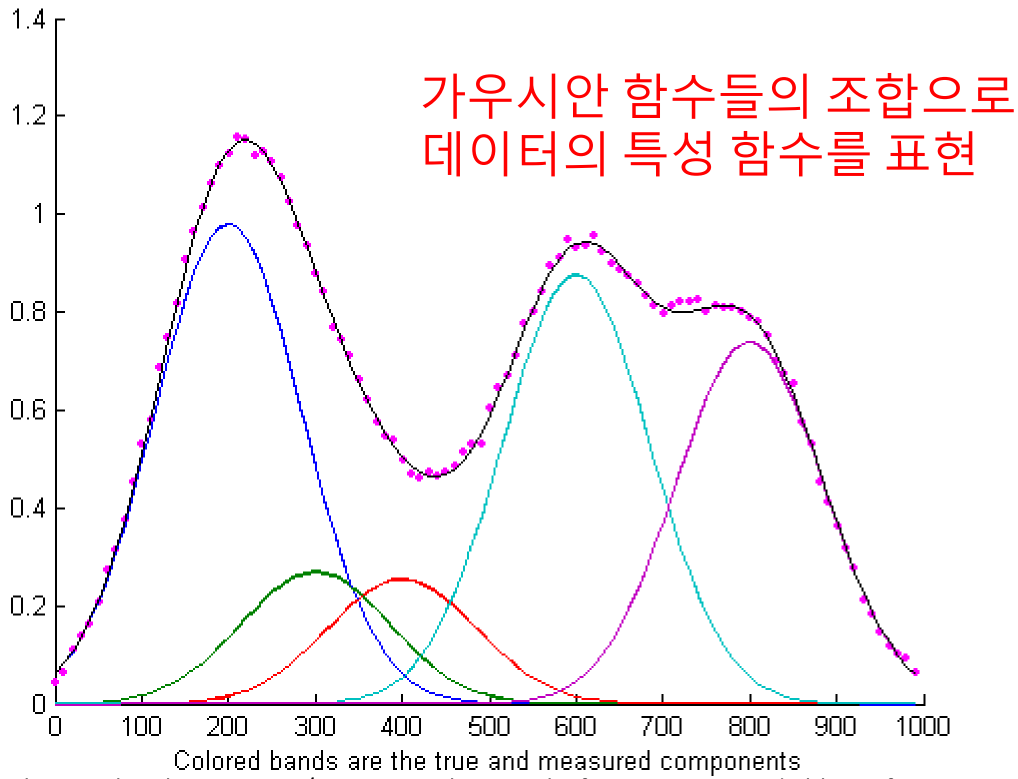




Non-Linear 함수들의 조합으로 표현하자!

대표적 기저함수 모델

- **Polynomial Basis Function(다항식 기저함수)** $$\phi_k(x) = x^k$$
- **Gaussian Basis Function(가우시안 기저함수)** $$\phi_k(x) = e^{-\frac{1}{2}\left(\frac{x-\mu_k}{\sigma}\right)^2}$$



출처: https://pythonmachinelearning.pro/using-neural-networks-for-regression-radial-basis-function-networks/

기저함수 개수 K에 따른 파라미터 설정

지능형 예측·진단 연구실
Intelligent Prognostics & Diagnostics Lab.

- **Gaussian Basis Function(가우시안 기저함수)**

$$\phi_k(x) = e^{-\frac{1}{2}\left(\frac{x-\mu_k}{\sigma}\right)^2}$$

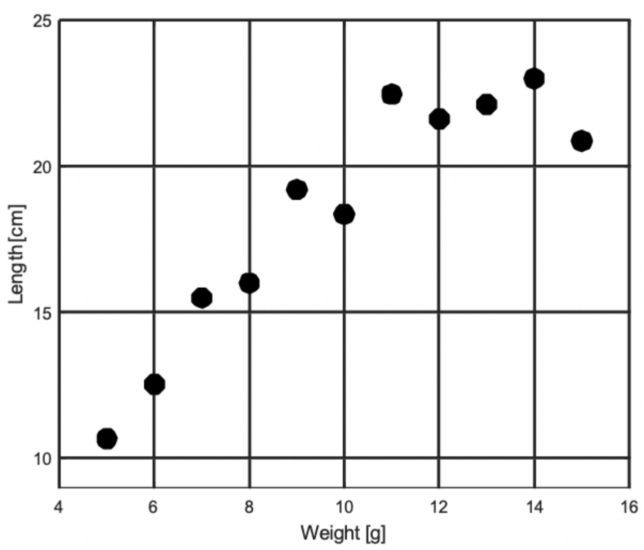
  - k번째 가우스 함수의 평균

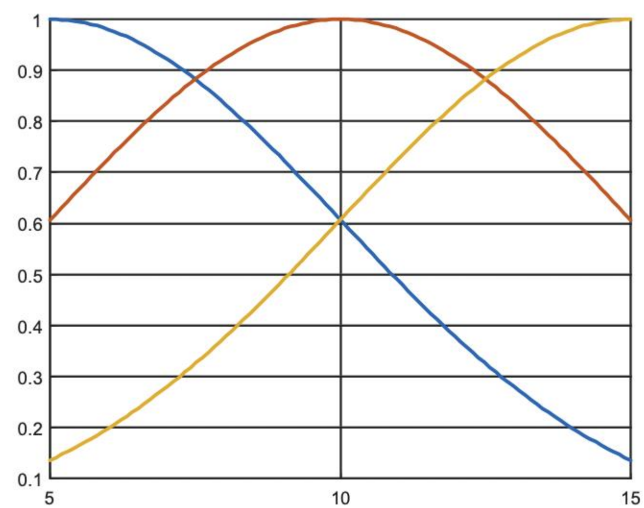
$$\mu_k = x_{\min} + \frac{x_{\max} - x_{\min}}{K-1}k, \quad k = 0, 1, \cdots, K-1$$

  - 모든 가우스 함수의 분산

$$\sigma = \frac{x_{\max} - x_{\min}}{K-1}$$

기저함수	매개변수	
	μ_k	σ
$\phi_0(x)$	5	5
$\phi_1(x)$	10	
$\phi_2(x)$	15	





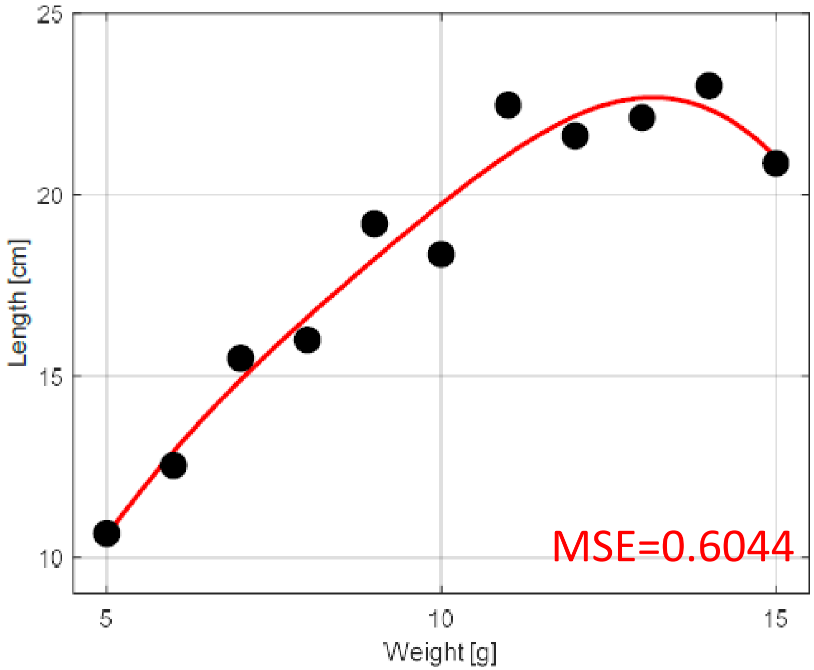
기저함수 수(K)에 따른 결과 비교

지능형 예측·진단 연구실
Intelligent Prognostics & Diagnostics Lab.

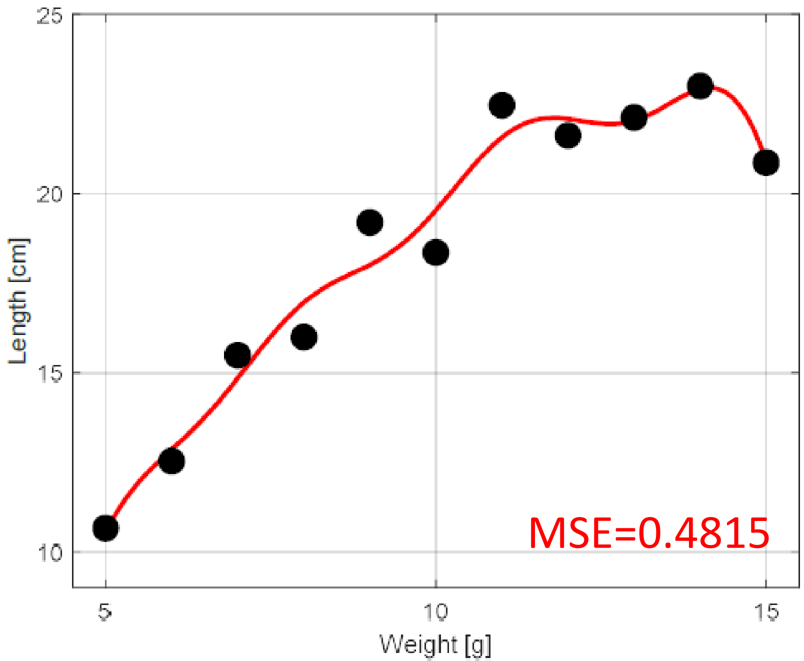
$$\hat{y} = 27.02\underbrace{\boxed{e^{-\frac{1}{2}\left(\frac{x-5}{5}\right)^2}}}_{\phi_0(x)} + 3.46\underbrace{\boxed{e^{-\frac{1}{2}\left(\frac{x-10}{5}\right)^2}}}_{\phi_1(x)} + 39.08\underbrace{\boxed{e^{-\frac{1}{2}\left(\frac{x-15}{5}\right)^2}}}_{\phi_2(x)} - 23.82$$

$$w = \begin{bmatrix} 27.02 \\ 3.46 \\ 39.08 \\ -23.82 \end{bmatrix}$$

$K = 3$



$K = 8$



MSE가 작다고 무조건 좋은 것일까?

선형 기저함수 모델의 Analytic Solution

– 비용함수: 평균제곱오차

$$\epsilon_{MSE} = \frac{1}{N}\sum_{n=0}^{N-1}\left(\hat{y}_n - y_n\right)^2 = \frac{1}{N}\sum_{n=0}^{N-1}\left(\boldsymbol{\phi}(x_n) - y_n\right)^2$$

$$\boldsymbol{\phi}(x_n) = \begin{bmatrix} \phi_0(x_n) \\ \phi_1(x_n) \\ \vdots \\ \phi_{K-1}(x_n) \\ \phi_K(x_n) \end{bmatrix}$$

– N개의 훈련 데이터 입력에 대한 행렬

$$\boldsymbol{\Phi} = \begin{bmatrix} \phi_0(x_0) & \phi_1(x_0) & \cdots & \phi_{K-1}(x_0) & \phi_K(x_0) \\ \phi_0(x_1) & \phi_1(x_1) & \cdots & \phi_{K-1}(x_1) & \phi_K(x_1) \\ \vdots & \vdots & & \vdots & \vdots \\ \phi_0(x_{N-1}) & \phi_1(x_{N-1}) & \cdots & \phi_{K-1}(x_{N-1}) & \phi_K(x_{N-1}) \end{bmatrix}$$

– N개의 훈련 데이터 출력에 대한 벡터

$$\boldsymbol{y} = \begin{bmatrix} y_0 \\ y_2 \\ \vdots \\ y_{N-1} \end{bmatrix}$$

– Analytic Solution

$$\boldsymbol{w} = \left(\boldsymbol{\Phi}^T\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^T\boldsymbol{y} \quad \Longleftrightarrow$$

다중 선형회귀 모델과 비교

$$w = \left(\mathrm{X}^{\mathrm{T}}\mathrm{X}\right)^{-1}\mathrm{X}^{\mathrm{T}}\mathrm{y}$$

Python 실습 과제 #4

지능형 예측·진단 연구실
Intelligent Prognostics & Diagnostics Lab.

- 실습 과제 #1에서의 데이터 set "lin_regression_data01.csv" 에 대해서

1) K개의 가우스 함수를 이용한 선형 기저함수 모델의 Analytic Solution을 구하는 사용자 지정함수를 구현하라.
    - 결과물: 코드

2) 1)에서 구현한 기저함수 모델을 활용하여, K=3, 6, 8 일 때에 대한 weight를 표로 나타내고, raw 데이터와 회귀곡선을 그래프로 나타내라.
    - 결과물: 표, 그래프, 분석내용

3) K= 3~10에 대한 MSE 값을 그래프로 나타내라. (x축: K, y축:MSE)
    - 결과물: 그래프 및 분석내용