인공지능 개론
L02 Regression

국민대학교
소프트웨어융합대학원
박하명

지도학습 Supervised Learning

훈련 데이터(Training Data)로부터 하나의 함수를 유추해내기 위한 기계 학습(Machine Learning)의 한 방법

Training Data

```
[1.2,  3.8, -1.4, …,  4.1] →  1.1
[3.2, -1.2, -0.2, …,  2.1] →  2.7
[2.8, -1.4, -0.3, …,  2.3] →  2.8
[1.2,  3.4, -1.5, …,  4.2] →  0.9
[4.2,  2.1,  2.8, …, -0.5] → -0.1
                    …
[3.2,  2.2,  2.2, …, -0.4] → -0.2
```

Test

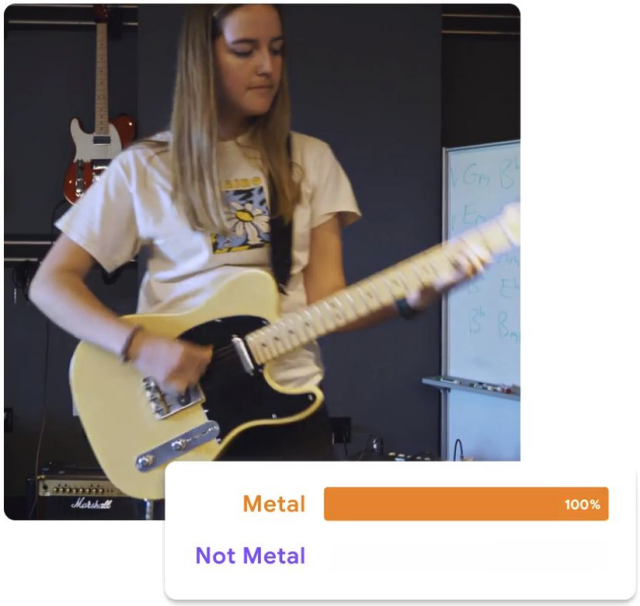
```
[1.3,  3.2, -1.5, …,  4.1] → ?
```

https://teachablemachine.withgoogle.com/

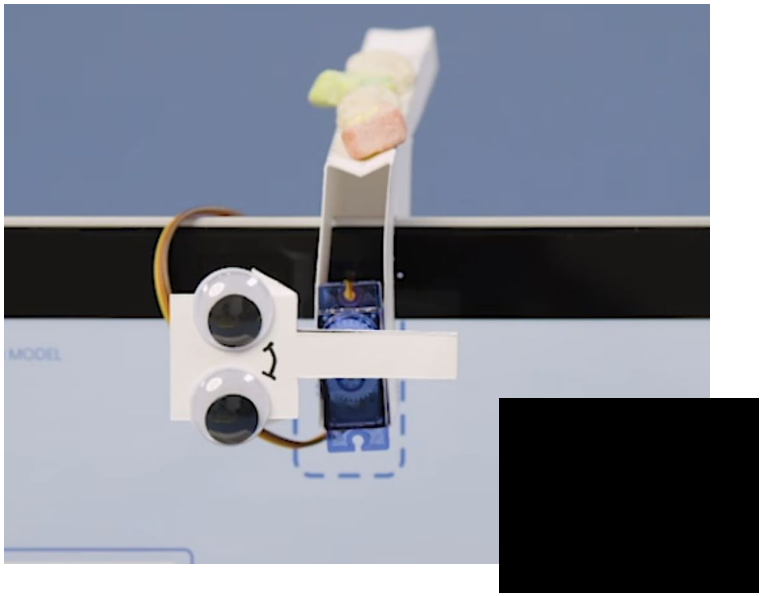
지도학습 예제: Teachable Machine

누구나 쉽게 **지도학습**을 할 수 있도록 구글에서 만든 온라인 서비스

이미지, 소리, 자세 등을 입력으로 받아서 분류작업Classification 수행



https://experiments.withgoogle.com/tiny-sorter/view



회귀분석 (Regression)

관찰된 연속형 변수들에 대해 **두 변수 사이의 모형**을 구한뒤 **적합도를 측정**해 내는 분석 방법 (출처: 위키피디아)

→ 지도 학습 방법 중 하나

선형회귀분석 (Linear Regression): **종속 변수 y**와 한 개 이상의 **독립 변수 (또는 설명 변수) X**와의 **선형 상관 관계를 모델링**하는 회귀분석 기법 (출처: 위키피디아)

Linear Regression

입력이 복잡한 지도학습…

Training Data

```
[1.2,  3.8, -1.4, …,  4.1] →  1.1
[3.2, -1.2, -0.2, …,  2.1] →  2.7
[2.8, -1.4, -0.3, …,  2.3] →  2.8
[1.2,  3.4, -1.5, …,  4.2] →  0.9
[4.2,  2.1,  2.8, …, -0.5] → -0.1
                   …
[3.2,  2.2,  2.2, …, -0.4] → -0.2
```

Test

```
[1.3,  3.2, -1.5, …,  4.1] → ?
```

쉬운 설명을 위해…
단순한 예제.

Training Data

```
[1.2] →  1.1
[3.2] →  2.7
[2.8] →  2.8
[1.2] →  0.9
[4.2] → -0.1
   …
[3.2] → -0.2
```

Test

```
[1.3] → ?
```

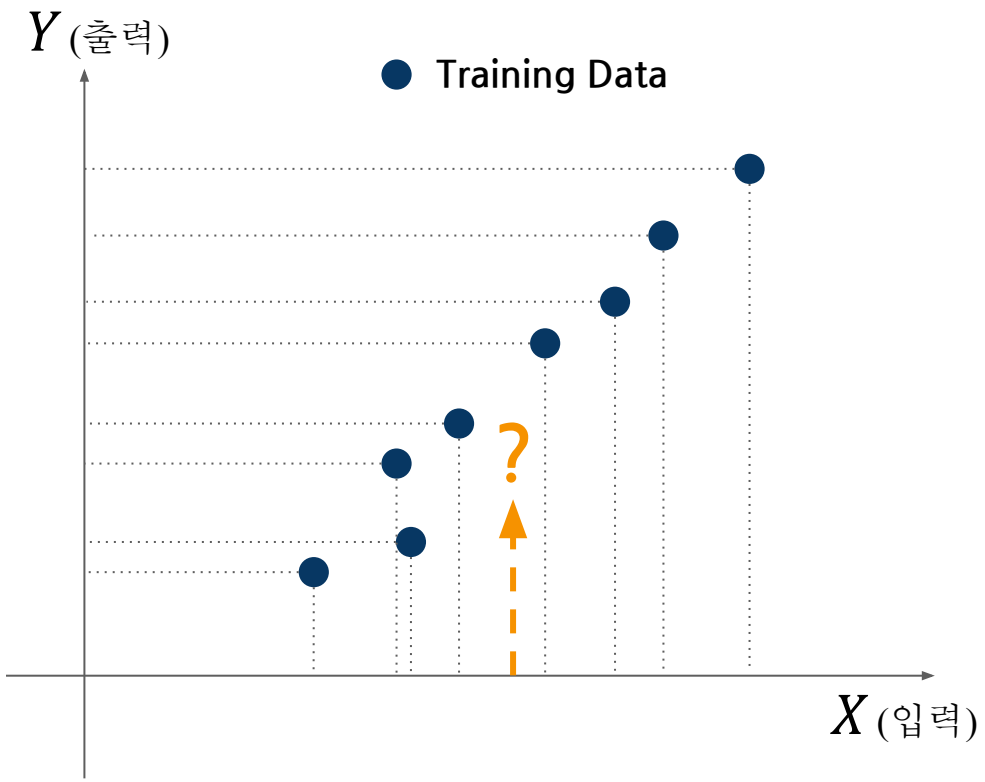
Linear Regression

Training Data

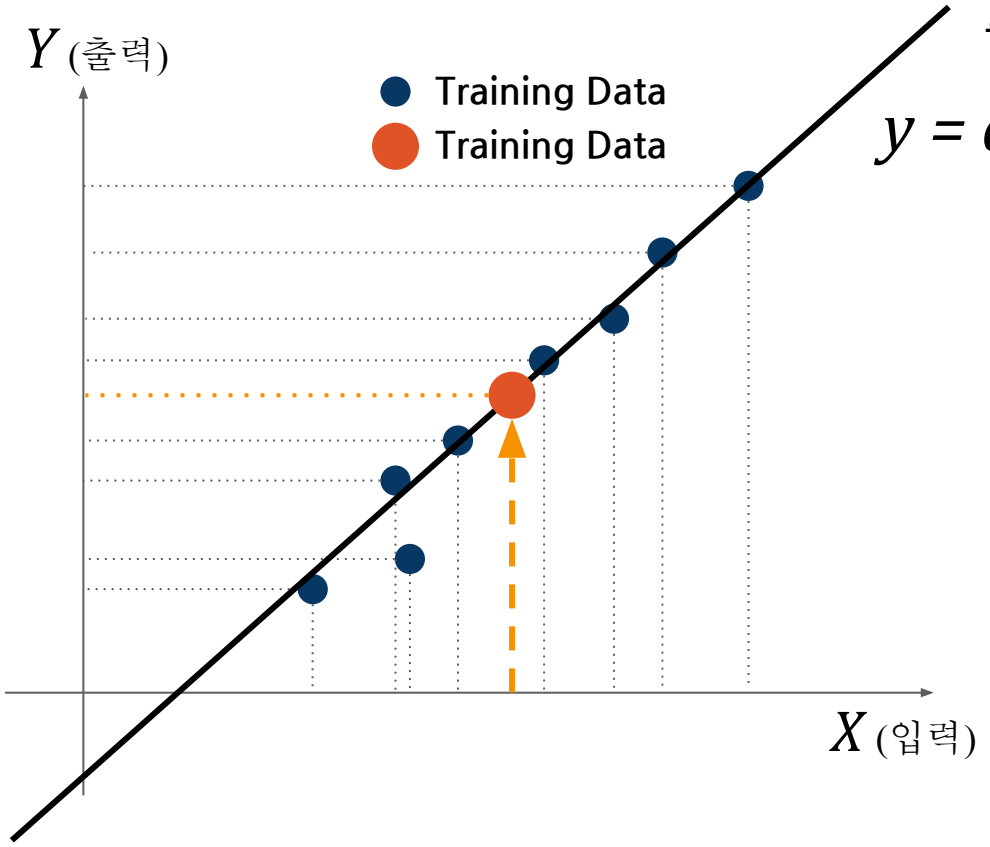
```
[1.2] →  1.1
[3.2] →  2.7
[2.8] →  2.8
[1.2] →  0.9
[4.2] → -0.1
   …
[3.2] → -0.2
```

Test

```
[1.3] →  ?
```



Linear Regression



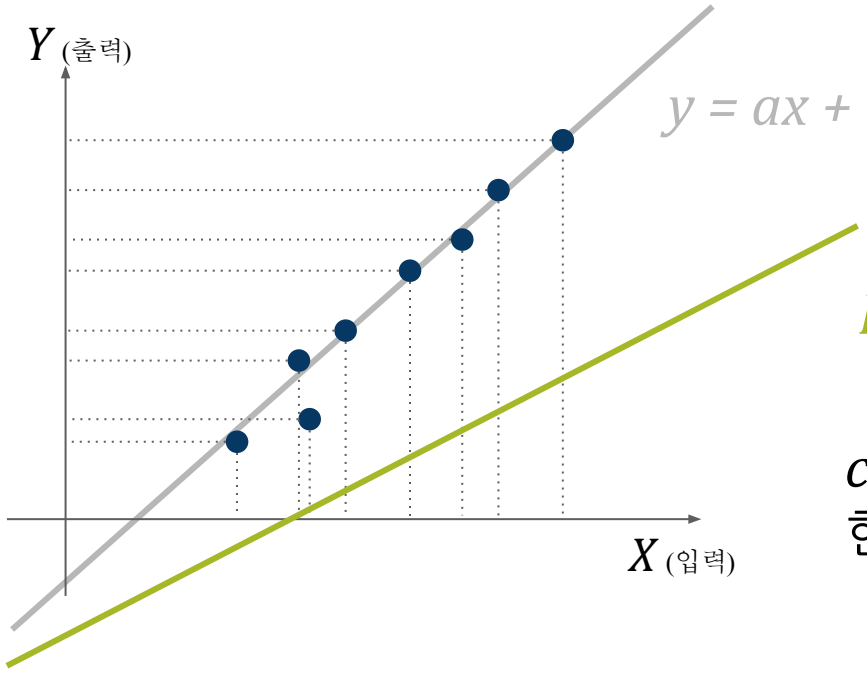
$y = ax+c$

이 **선**만 구할 수 있으면,
어떤 **입력**이 들어와도
출력을 **예측**할 수 있다!

선을 구한다
= a 값과 c 값을 구한다

어떻게 구하지..?

가설과 비용 Hypothesis and Cost function



가설Hypothesis

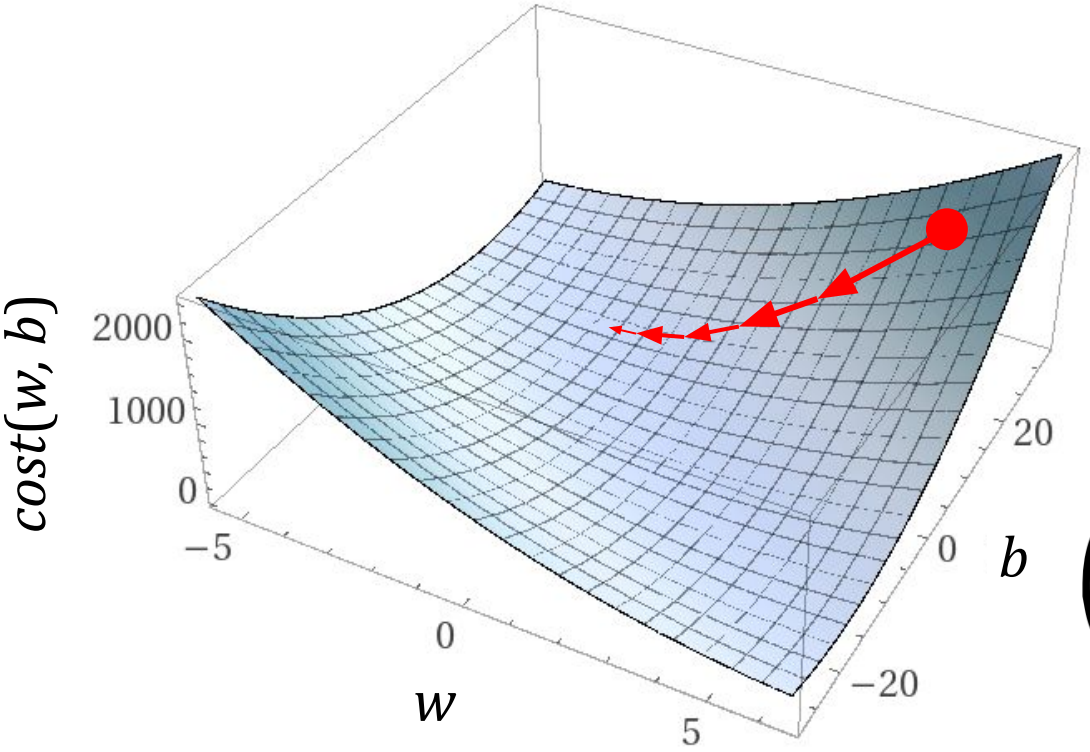
$H(x) = wx + b$

$cost(w,b)$:
현재 가설은 얼마나 잘못되었는가?

경사 하강법 Gradient Descent

우리의 목표: cost를 최소화 하자! = cost를 최소로 만드는 w, b 값을 찾자!

$$\arg\min_{w,b} cost(w,b)$$



경사 따라 내려가야하는데,
경사는 어떻게 구하지?
⇒ **편미분** 이용

경사:

$$\left(\frac{\partial cost(w,b)}{\partial w}, \frac{\partial cost(w,b)}{\partial b} \right)$$

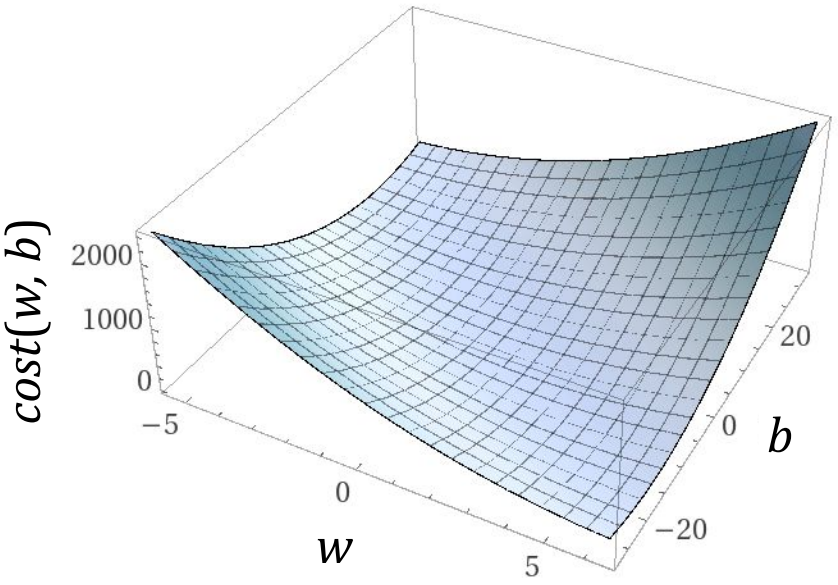
경사 하강법 Gradient Descent

경사: $\left(\frac{\partial cost(w,b)}{\partial w}, \frac{\partial cost(w,b)}{\partial b} \right)$

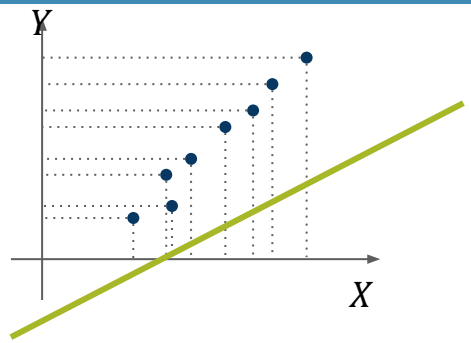
업데이트: Learning Rate

$$w = w - \alpha \frac{\partial cost(w,b)}{\partial w}$$

$$b = b - \alpha \frac{\partial cost(w,b)}{\partial b}$$



Linear Regression (2)



입력이 조금 더 복잡할 때?

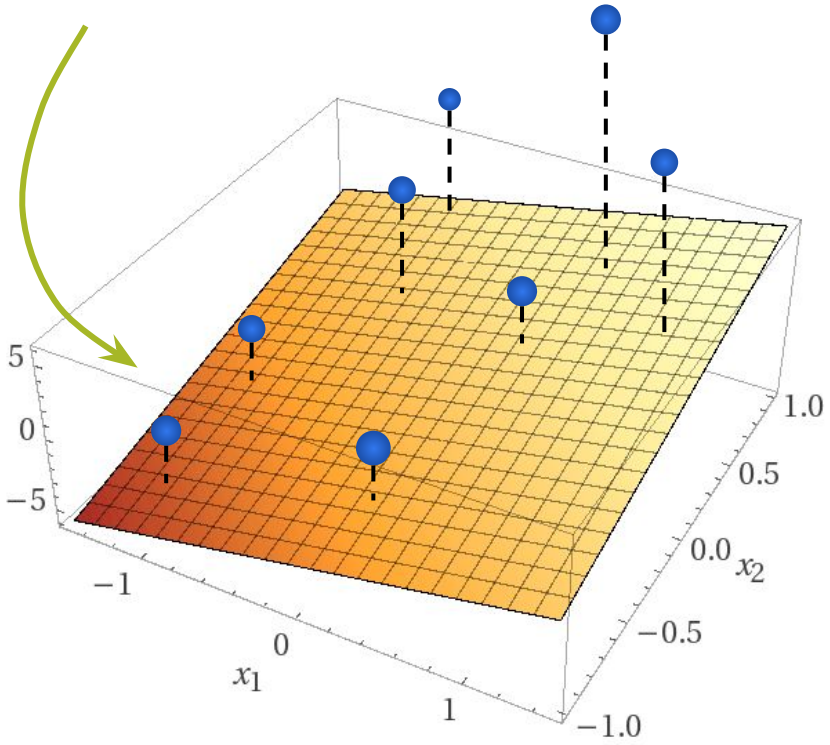
Training Data

```
[1.2,  3.8] →  1.1
[3.2, -1.2] →  2.7
[2.8, -1.4] →  2.8
[1.2,  3.4] →  0.9
[4.2,  2.1] → -0.1
          …
[3.2,  2.2] → -0.2
```

Test

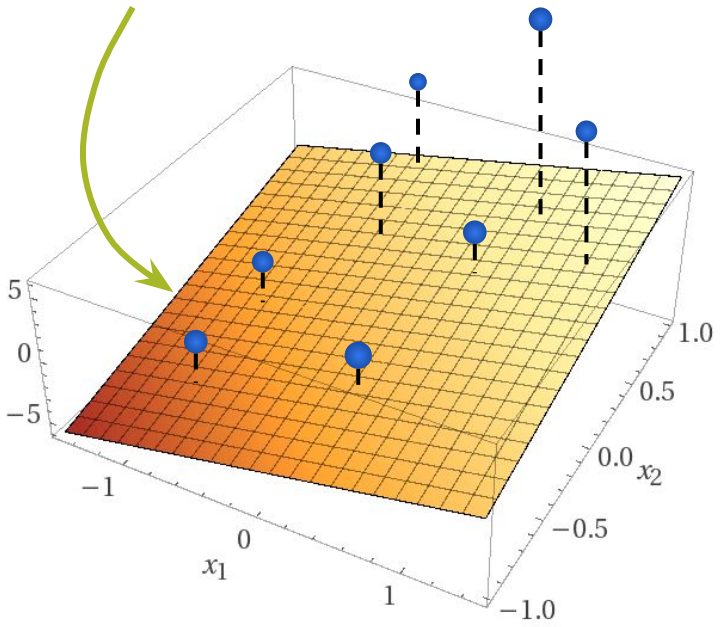
```
[1.3,  3.2] → ?
```

$$H(x_1, x_2) = w_1 x_1 + w_2 x_2 + b$$



가설과 비용 (2) Hypothesis and Cost function (2)

$H(x_1, x_2) = w_1x_1 + w_2x_2 + b$



$$\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$H(x_1, x_2) = H(\mathbf{x}) = \mathbf{w}^\mathsf{T}\mathbf{x} + b$$

$$cost(\mathbf{w}, b) = \frac{1}{n}\sum_{i=0}^{n}\left(y_i - H(\mathbf{x}_i)\right)^2$$

경사 하강법 (2) Gradient Descent (2)

우리의 목표: cost를 최소화 하자! = cost를 최소로 만드는 **w**, b 값을 찾자!

$$\arg\min_{\mathbf{w},b} cost(\mathbf{w}, b)$$

업데이트:

$$w_1 = w_1 - \alpha \frac{\partial cost(\mathbf{w},b)}{\partial w_1}$$

$$w_2 = w_2 - \alpha \frac{\partial cost(\mathbf{w},b)}{\partial w_2}$$

$$b = b - \alpha \frac{\partial cost(\mathbf{w},b)}{\partial b}$$

$$\mathbf{w} = \mathbf{w} - \alpha \frac{\partial cost(\mathbf{w},b)}{\partial \mathbf{w}}$$

Learning Rate

경사 (Gradient)

Linear Regression (3)

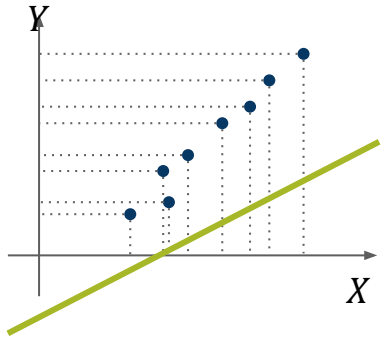
입력이 조금 더 더 복잡할 때?

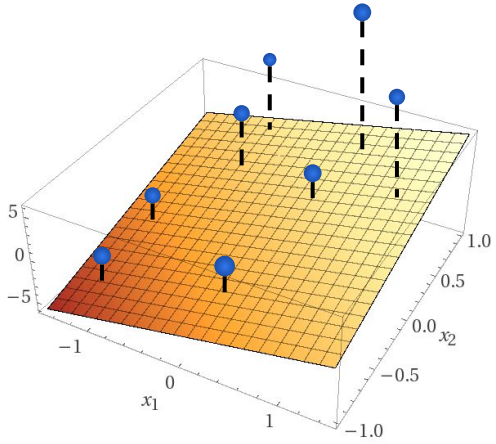
Training Data

```
[1.2,  3.8, -1.4, …,  4.1] →  1.1
[3.2, -1.2, -0.2, …,  2.1] →  2.7
[2.8, -1.4, -0.3, …,  2.3] →  2.8
[1.2,  3.4, -1.5, …,  4.2] →  0.9
[4.2,  2.1,  2.8, …, -0.5] → -0.1
                   …
[3.2,  2.2,  2.2, …, -0.4] → -0.2
```

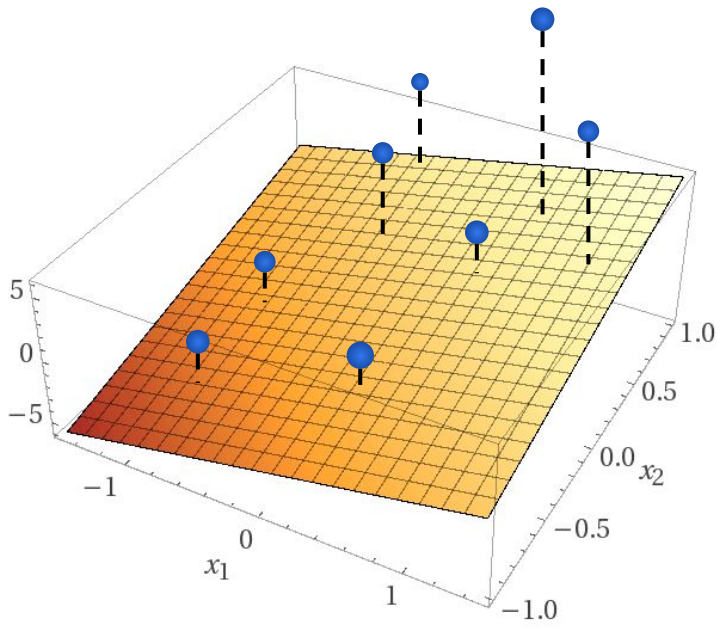
Test

```
[1.3,  3.2, -1.5, …,  4.1] → ?
```





가설과 비용 (3) Hypothesis and Cost function (3)



$$\mathbf{w} = \begin{bmatrix} w_1 \\ \vdots \\ w_m \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix}$$

$$H(\mathbf{x}) = \mathbf{w}^{\mathsf{T}}\mathbf{x} + b$$

$$cost(\mathbf{w}, b) = \frac{1}{n}\sum_{i=0}^{n}\left(y_i - H(\mathbf{x}_i)\right)^2$$

경사 하강법 (3) Gradient Descent (3)

우리의 목표: cost를 최소화 하자! = cost를 최소로 만드는 **w**, b 값을 찾자!

$$\arg\min_{\mathbf{w},b} cost(\mathbf{w}, b)$$

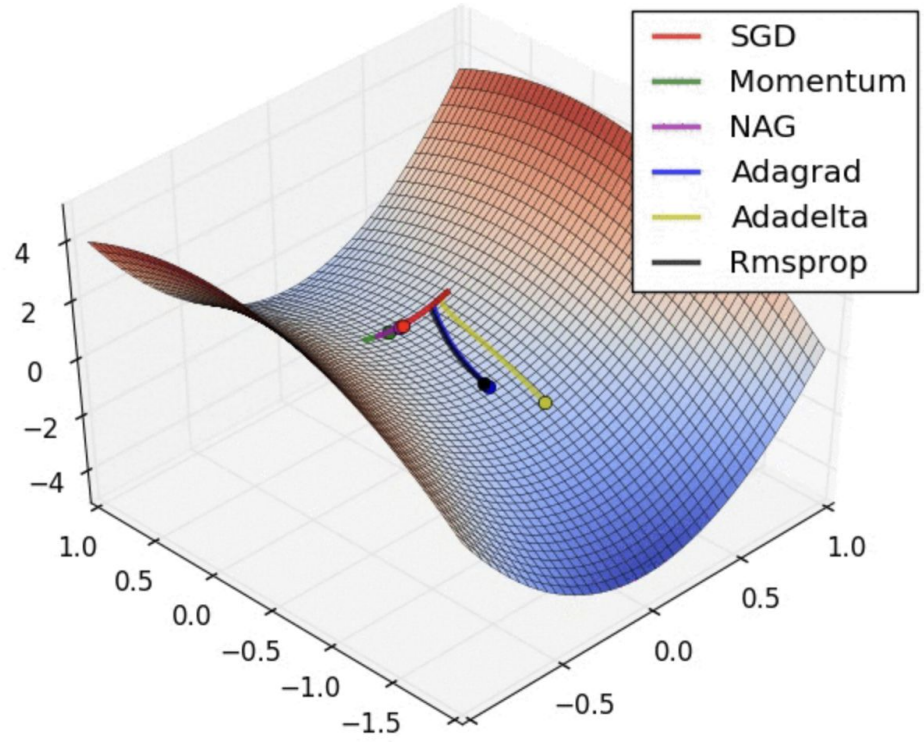
업데이트:

$$\mathbf{w} = \mathbf{w} - \alpha \frac{\partial cost(\mathbf{w}, b)}{\partial \mathbf{w}}$$

$$b = b - \alpha \frac{\partial cost(\mathbf{w}, b)}{\partial b}$$

그밖의 학습 방법들

경사하강법은 최적해를 찾기위한 한 가지 방법일 뿐, 다양한 업데이트 방법이 존재: [링크] Gradient Descent Optimization Algorithms 정리



Question?