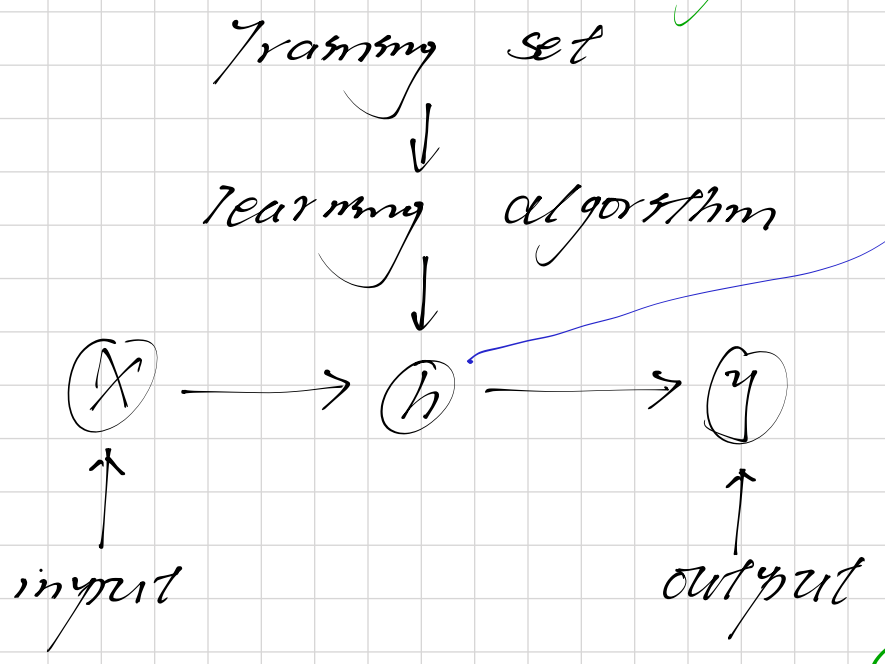
§ Machine Learning – Week 1

1. ML: A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P. If its performance at tasks in T as measured by P, improves with experience E.

ML
- Supervised Learning
  - ① regression: results within a continuous output
  - ② classification: results within a discrete output
- Unsupervised Learning
  - ① no label
  - ② A structure of data

2.



$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots$$

$$h_\theta(x) = \sum_{i=0}^{n} \theta_i x_i = \theta^T x$$

Cost function: $J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$ (m → training examples)

Goal: $\min_\theta J(\theta)$

3. $\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$

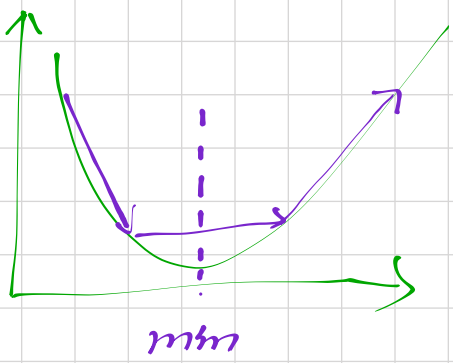
α: Learning rate

↗ small: gradient descent ↓

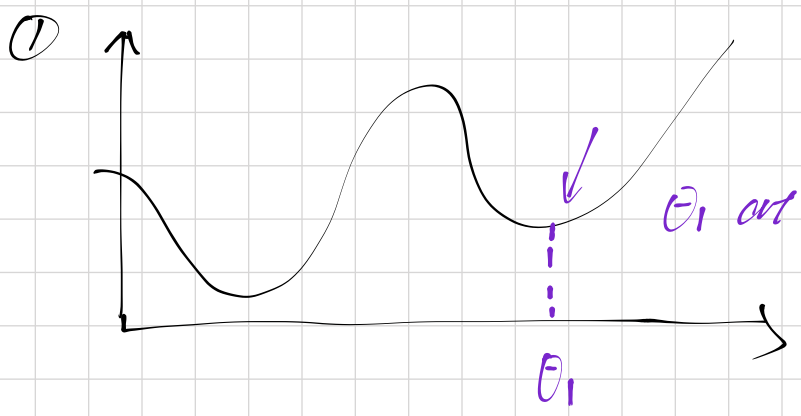
↘ large: gradient descent ↑

☆ α large: may overshoot the min

eg.



α large

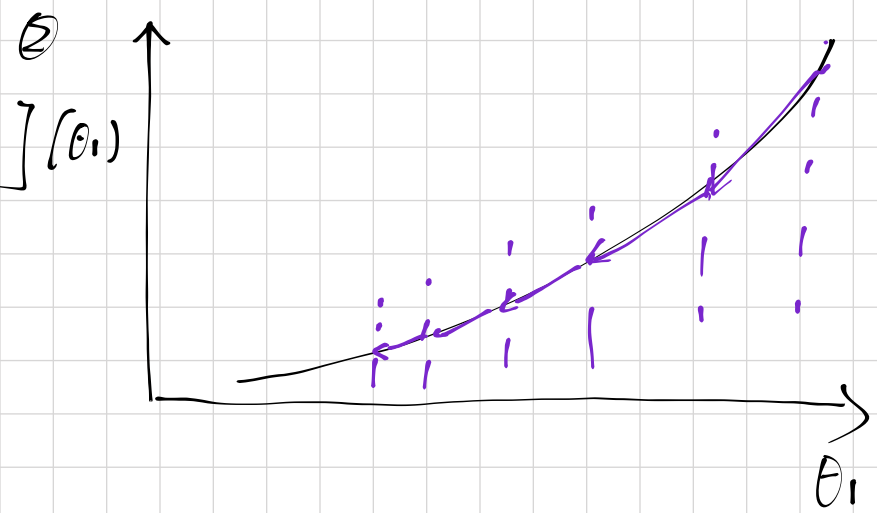


θ_1 at local optima

$\therefore \theta_1 := \theta_1 - \alpha \underbrace{\frac{d}{d\theta_1} J(\theta_1)}_{=0}$

$\therefore \theta_1 \longrightarrow$ unchanged

gradient descent can converge to a local minimum, even with the learning rate α fixed



$\theta_1 := \theta_1 - \alpha \underbrace{\frac{d}{d\theta_1} J(\theta_1)}_{\text{changed}}$

$\therefore$ approach a local minimum, gradient descent will automatically take smaller steps

$\therefore$ don't need to decrease α

4. $\begin{cases} \theta_j := \theta_j - \alpha \dfrac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \\ J(\theta) = \dfrac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 \end{cases}$

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{\partial}{\partial \theta_j} \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

$$= \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right) x_j$$

$$\Rightarrow \theta_j := \theta_j + \alpha \left(y^{(i)} - h_\theta(x^{(i)})\right) x_j^{(i)}$$

① Update $\theta_0, \theta_1, \cdots, \theta_j$ at the same time

② "Batch" gradient descent: use all training set