Linear regression with one variable

2.1) Model representation

⇒ Data set is called training set.

Notation:

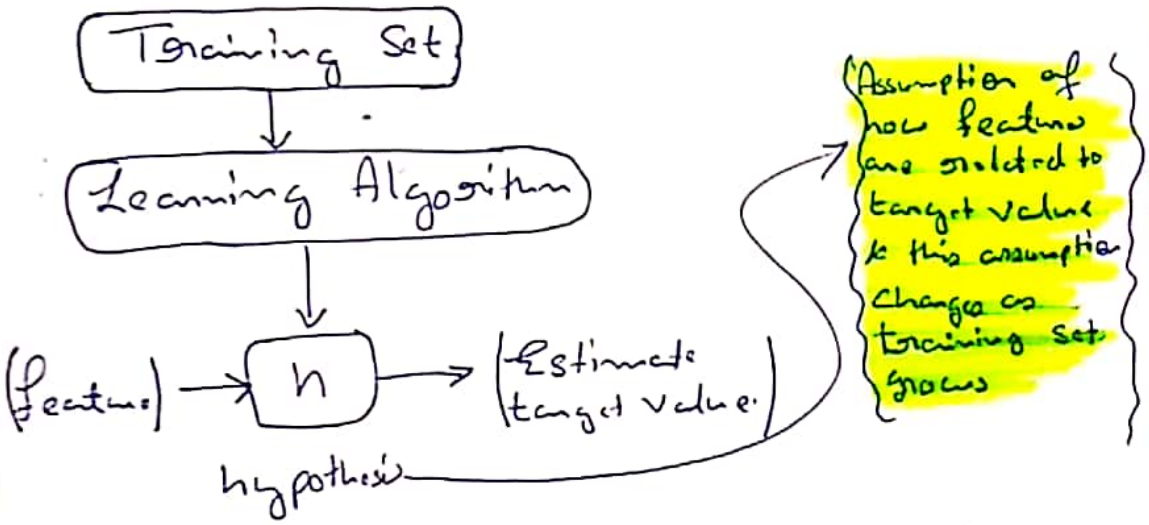
m ⇒ Number of training example

x's ⇒ "input" variable / features

y's ⇒ "output" variable / "target" variable

(x, y) ⇒ To denote single training example.

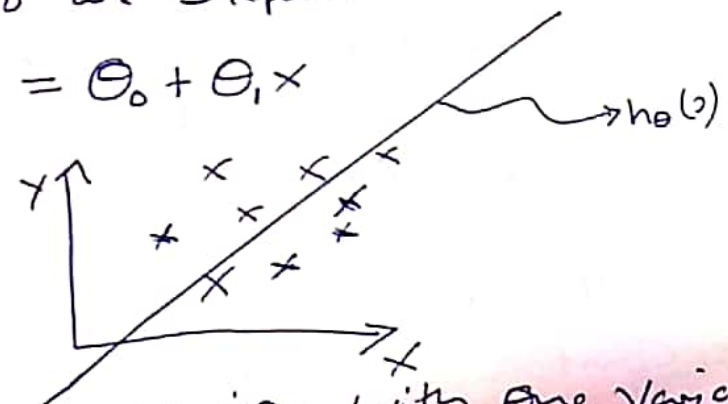
$(x^{(i)}, y^{(i)})$ ⇒ To denote i^{th} training example.



⇒ h is a function which maps x's to y's

⇒ How do we represent h?

$$h_\theta(x) = \theta_0 + \theta_1 x$$



⇒ Linear regression with one variable.
(i.e. Univariate linear regression)

2.) Cost function

Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$

{θ_i's ⇒ Parameters}

Objective: Choose θ_0, θ_1 so that $h_\theta(x)$ is close to y for our training examples (x, y).

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

↑ Cost function

↓ {Squared error cost function}

Goal: Minimize $J(\theta_0, \theta_1)$ over θ_0, θ_1

→ {Minimize $J(\theta_0, \theta_1)$ to obtain value of θ_0 & θ_1}

Contour line ⇒ A contour line of a function of two variables is a curve along which the function has a constant value, so that the curve joins points of equal value.

→ different colors are used to represent contour lines at different height.

Gradient descent

{An algorithm for minimizing the cost function J}

$$\min_{\theta_0 \cdots \theta_n} J(\theta_0, \cdots \theta_n)$$

⇒ First start from a random value

* Gradient descent algorithm

repeat until convergence {

$$\theta_j := \theta_j - \alpha \frac{\delta}{\delta \theta_j} J(\theta_0, \theta_1) \quad (\forall\ j=0 \text{ and } j=1)$$

}

⇓ {Correct simultaneous update}

$$\text{temp0} := \theta_0 - \alpha \frac{\delta}{\delta \theta_0} J(\theta_0, \theta_1)$$

$$\text{temp1} := \theta_1 - \alpha \frac{\delta}{\delta \theta_1} J(\theta_0, \theta_1)$$

$$\theta_0 := \text{temp0}$$

$$\theta_1 := \text{temp1}$$

→ {Assignment operator}

α ⇒ learning rate (i.e. how big the step is)

{= → {Truth assertion}}

Issue: It can be susceptible to local optima.

⇒ The algorithm is also called (Batch) Gradient Descent.

Each step of gradient descent uses all the training examples