derive new features.

$$h_\theta(x) = \theta_0 + \theta_1 \times \underbrace{\text{frontage}}_{x_1} + \theta_2 \times \underbrace{\text{depth}}_{x_2}$$

$$\text{Area} = x_1 \times x_2$$

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

might be a better feature.

---

poly regregression.

~~x is size~~

$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$$

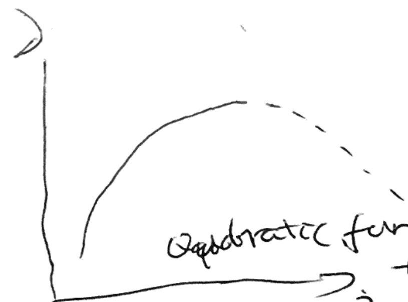
$$= \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$$

size = 1000

$size^2$ = 1000^2

$\sim^3$ = 1000^3

scale feature !!!



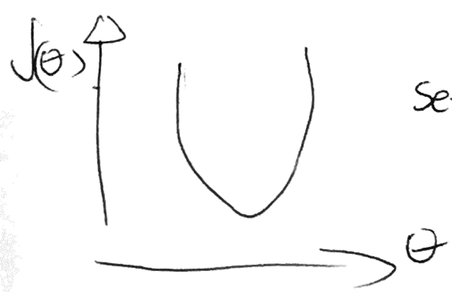
Quadratic function might decrease the price.

$\theta_0 + \theta_1 x_1 + \theta_2 x^2$.

$\theta_0 + \theta_1 x + \theta_2 \sqrt{x}$ ← could be better.

Normal Equation.

$$J(\theta) = a\theta^2 + b\theta + c$$



Set $\frac{dJ(\theta)}{d\theta} = 0$

solve for θ.

$$\frac{\partial J(\theta)}{\partial \theta_j} = 0 \text{ for every } j.$$

solve for $\theta_0, \theta_1 \cdots \theta_n$

x_0	x_1	x_2	x_3	x_4	y

$$X = m\underbrace{\begin{bmatrix} x_0^1 & x_1^1 & x_2^1 & x_3^1 & x_4^1 \\ x & - & - & - & - \\ \vdots & & & & \\ x_0^m & x_1^m & & & x_4^m \end{bmatrix}}_{n+1}$$

$$Y = \begin{bmatrix} y^1 \\ y^2 \\ \vdots \\ y^m \end{bmatrix} \Big\} m.$$

$$\theta = (X^T X)^{-1} X^T y.$$

No need for feature scaling

* compare GD & NE.

GD	NE.
choose α.	no α.
iterate a lot	no iteration
works when n is large. $O(n^2)$	slow when n is large $> 10{,}000$

↑ $(X^T X)^{-1}$ — $O(n^3)$.

Non invertible $X^T X$?

① redundant feature.
eg. linearly dependant feature.

② ~~delete~~ m is much less than n. (delete or regularize)