Artificial Intelligence Foundations and Applications

Machine Learning – Part 2

Linear Models

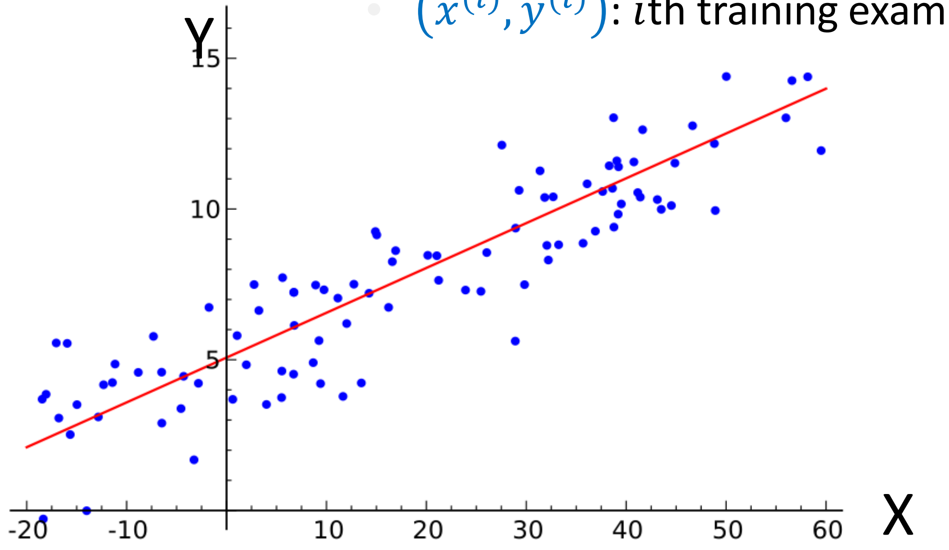
Sudeshna Sarkar

Nov 7 2022

Centre of Excellence in Artificial Intelligence, IIT Kharagpur

Linear Regression

- n features
- m training examples
- $(x^{(i)}, y^{(i)})$: ith training example

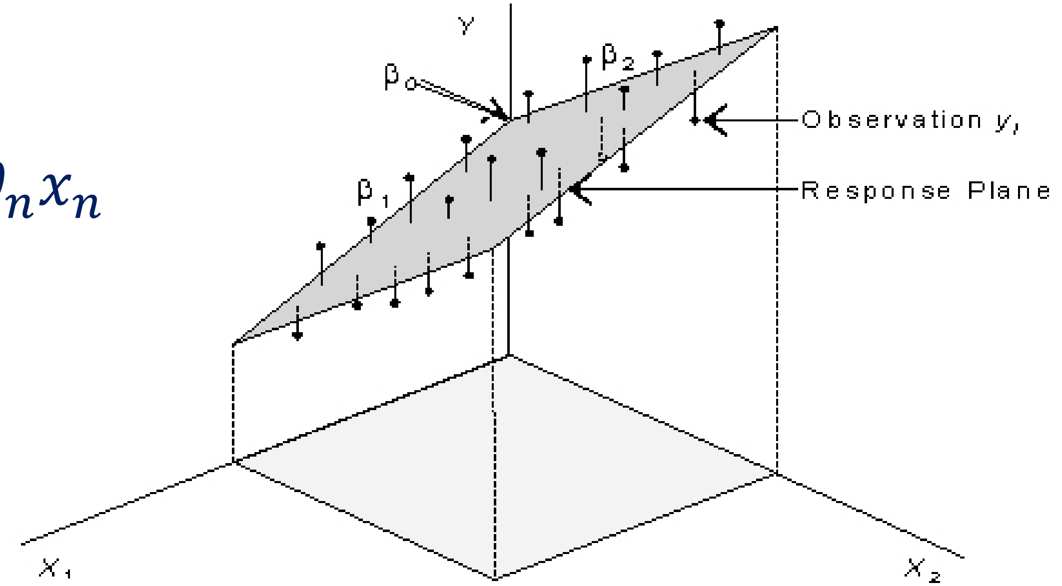


Understand the relationship between dependent variable x and explanatory variable y

predict y from x

Linear Model:

$$y = h_\theta(\mathrm{x}) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

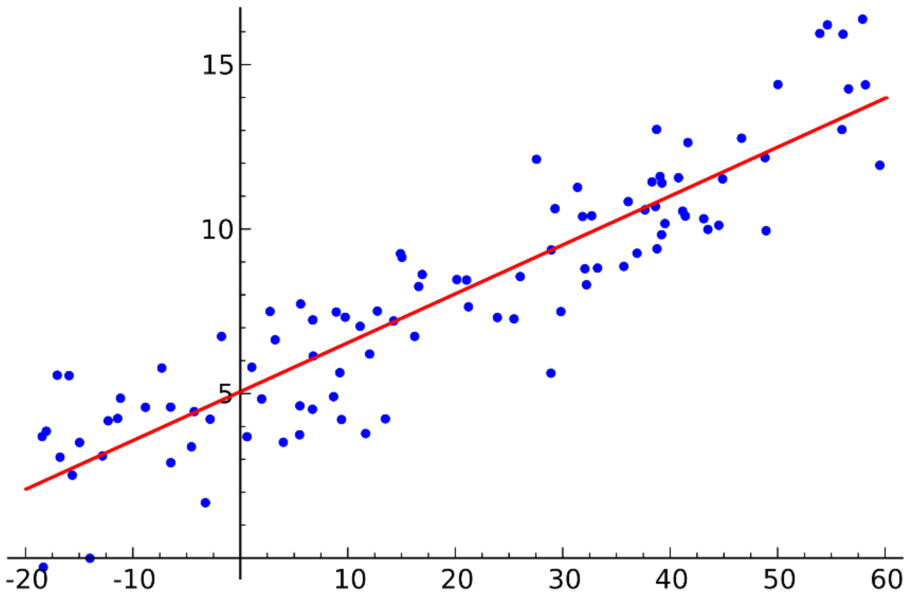


Linear hypothesis function: Intuition

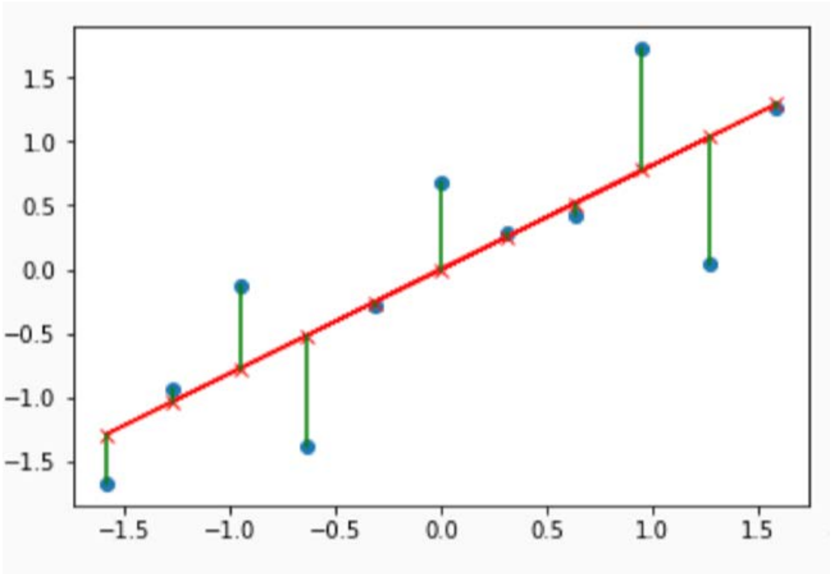
$$h_\theta(\mathrm{x}) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

Two equivalent questions:

1. Which is the best straight line to fit the data?
2. How to learn the values of the parameters θ_i ?



Cost function



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

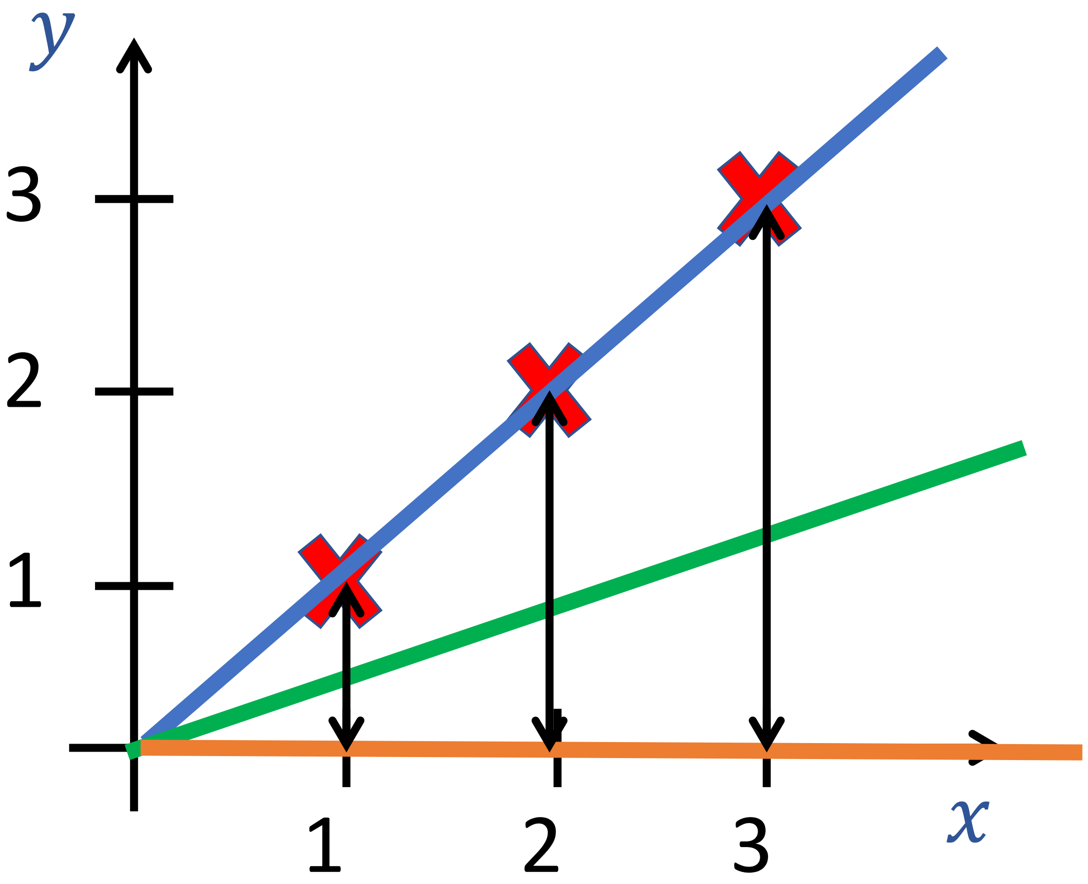
$$e^{(i)} = \widehat{y^{(i)}} - y^{(i)} = h_\theta\left(x^{(i)}\right) - y^{(i)}$$

prediction error for ith training example

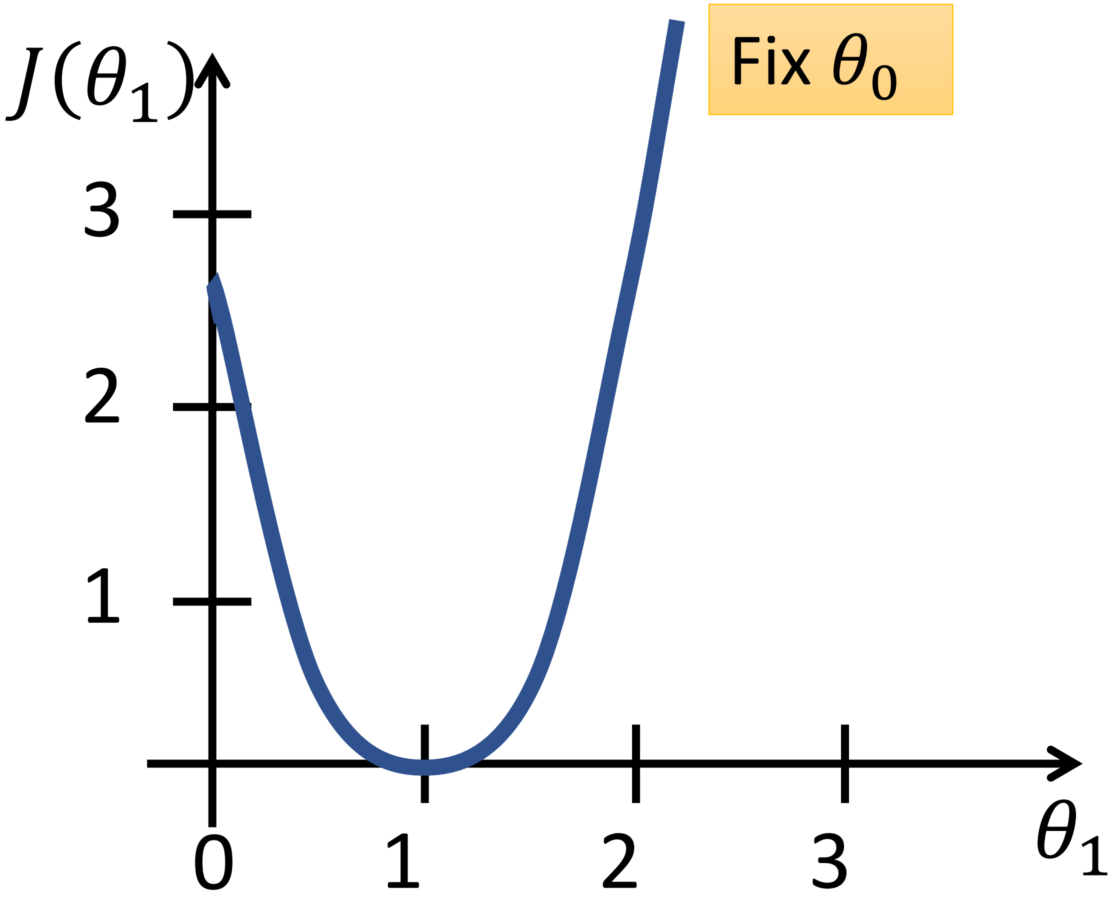
Choose parameters $\bar{\theta}$ so that

$J(\bar{\theta})$ is minimized

$h_\theta(x)$: function of x for fixed θ

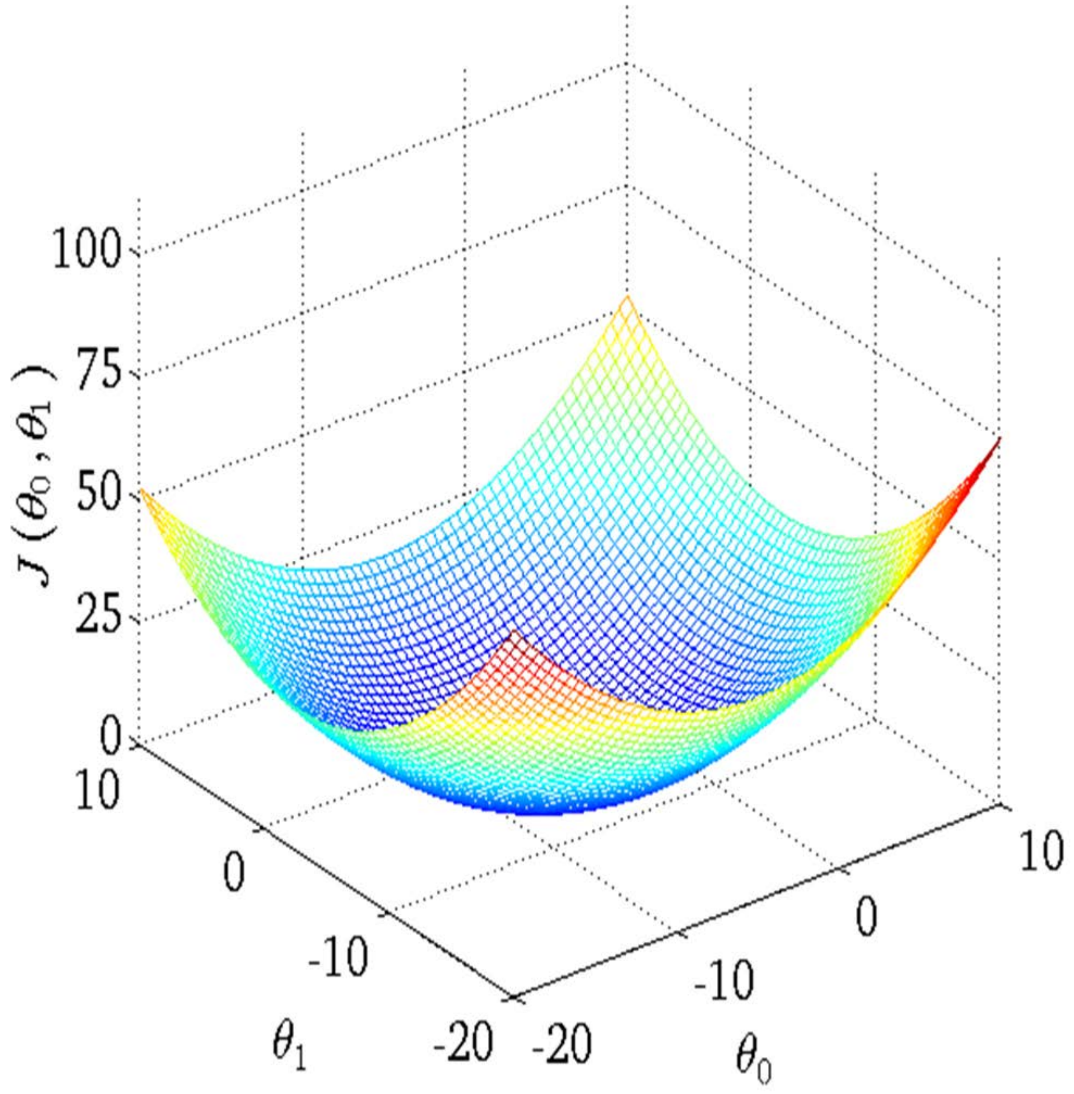


$J(\theta)$, function of θ_0, θ_1



Cost Function

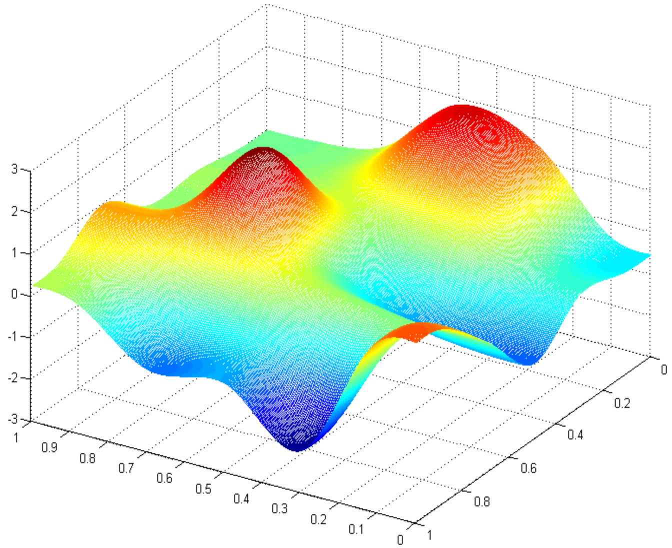
When J is a function of both θ_0 and θ_1

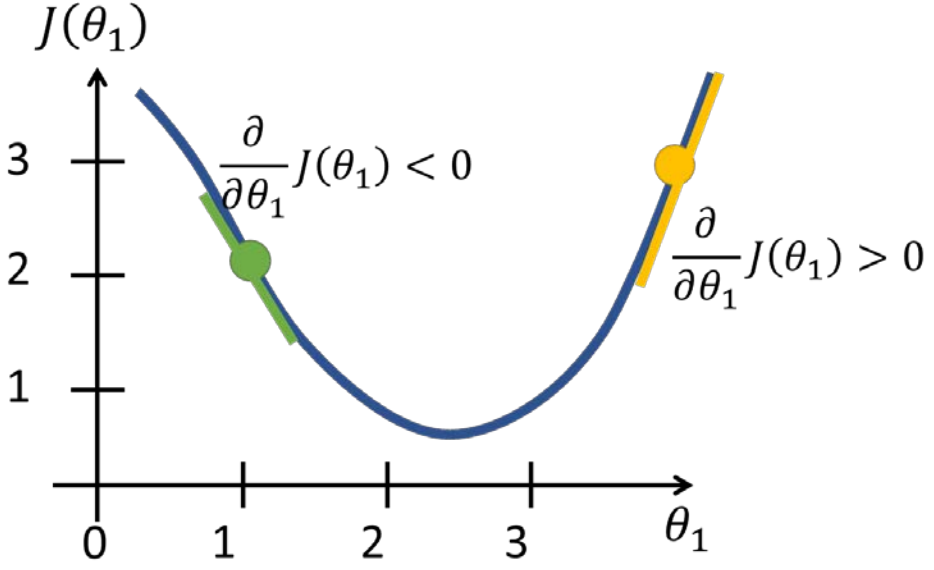


Minimizing cost function & Gradient Descent

Minimizing function $J(\theta_0, \theta_1)$

- Start with some θ_0, θ_1
- Keep changing θ_0, θ_1 to reduce $J(\theta_0, \theta_1)$
- until we end up at a minimum





$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)$$

Computing partial derivatives

Repeat until convergence{

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\bar{\theta})$$

Equivalently

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

}

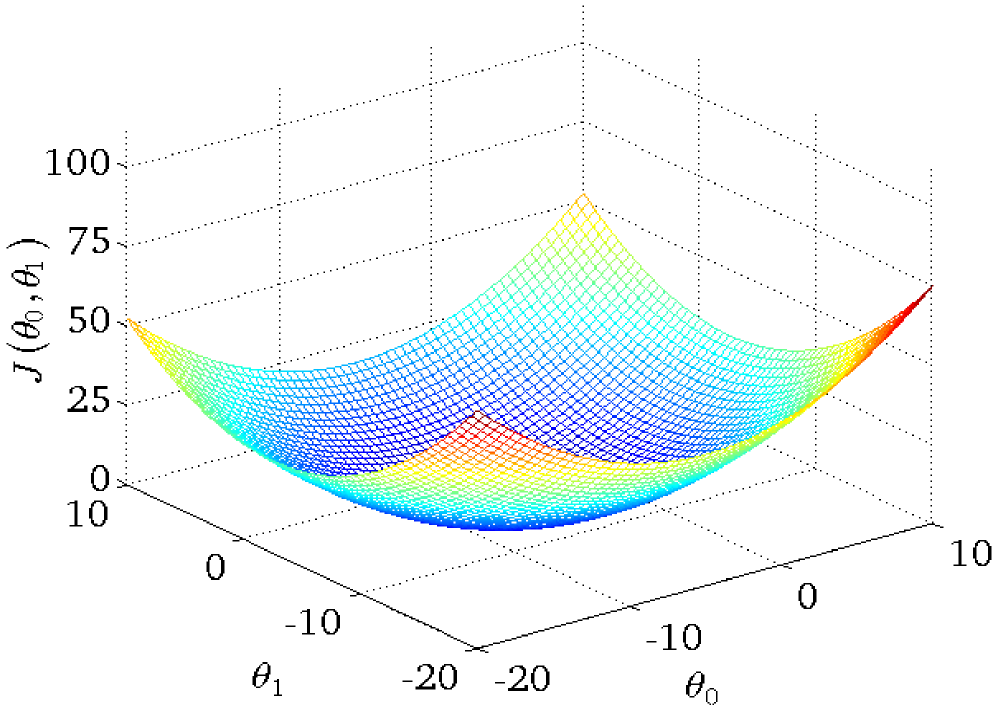
$$J(\bar{\theta}) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

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

Convergence

- The cost function in linear regression is always a convex function – always has a single global minimum

- So, gradient descent will always converge



Batch gradient descent

"Batch": Each step of gradient descent uses all the training examples

Repeat until convergence{

m: Number of training examples

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{\boxed{m}} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{\boxed{m}} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) x^{(i)}$$

}

Logistic Regression

for Classification

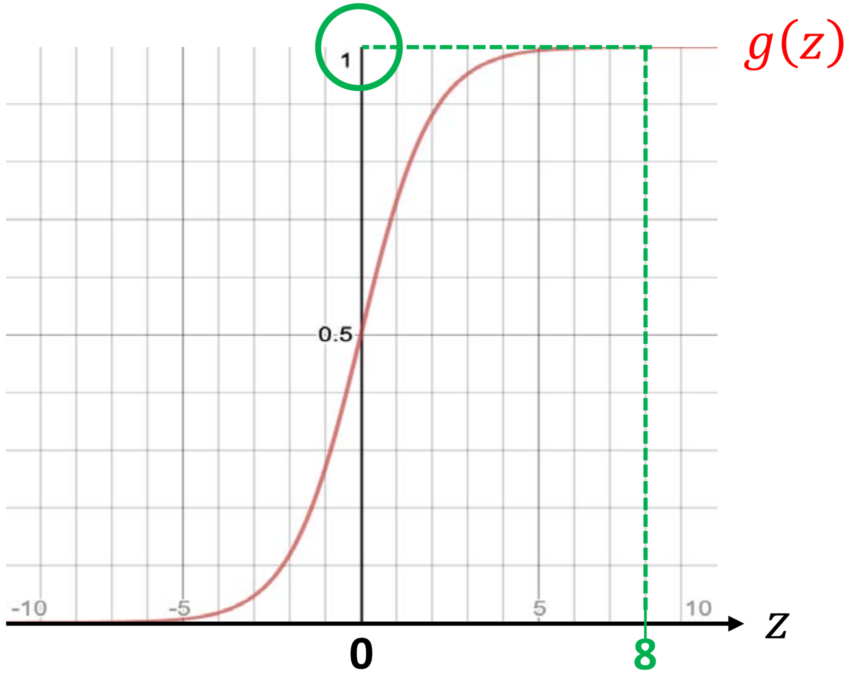
Regression vs. Classification

We want the possible outputs of $\boldsymbol{h_\theta(x) = \theta^T x}$ to be discrete-valued

Use an ***activation function*** (e.g., ***sigmoid or logistic function***)

$$g(z) = \frac{1}{1 + e^{-z}}$$

$z \in \mathbb{R}, but$
$g(z) \in [0,1]$



If y = **1**, we want $g(z) \approx 1$ (i.e., we want a correct prediction)
For this to happen, $\boldsymbol{z \gg 0}$

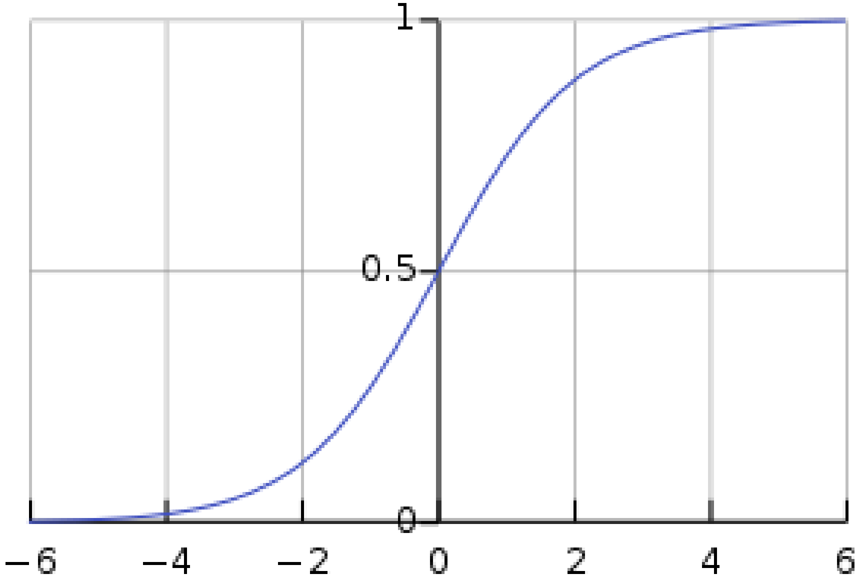
If y = **0**, we want $g(z) \approx 0$ (i.e., we want a correct prediction)
For this to happen, $\boldsymbol{z \ll 0}$

Classification

$$\boldsymbol{x} = [\boldsymbol{x}_0, \dots, \boldsymbol{x}_m] \rightarrow g(\boldsymbol{\theta}^T \boldsymbol{x}) \rightarrow \boldsymbol{h}_{\boldsymbol{\theta}}(\boldsymbol{x}) \in [0,1]$$

$$h_\theta(x) = g(\theta^\top x)$$

$$g(z) = \frac{1}{1 + e^{-z}}$$



Thresholding:
predict “y = 1” if $h_\theta(x) \geq 0.5$

predict “y = 0” if $h_\theta(x) < 0.5$

Classification

$$\boldsymbol{x} = [\boldsymbol{x}_0, \ldots, \boldsymbol{x}_m] \rightarrow \boldsymbol{g(\theta^T x)} \rightarrow \boldsymbol{h_\theta(x)} \in [0,1]$$

$$h_\theta(x) = g(\theta^\top x)$$

$$g(z) = \frac{1}{1 + e^{-z}}$$

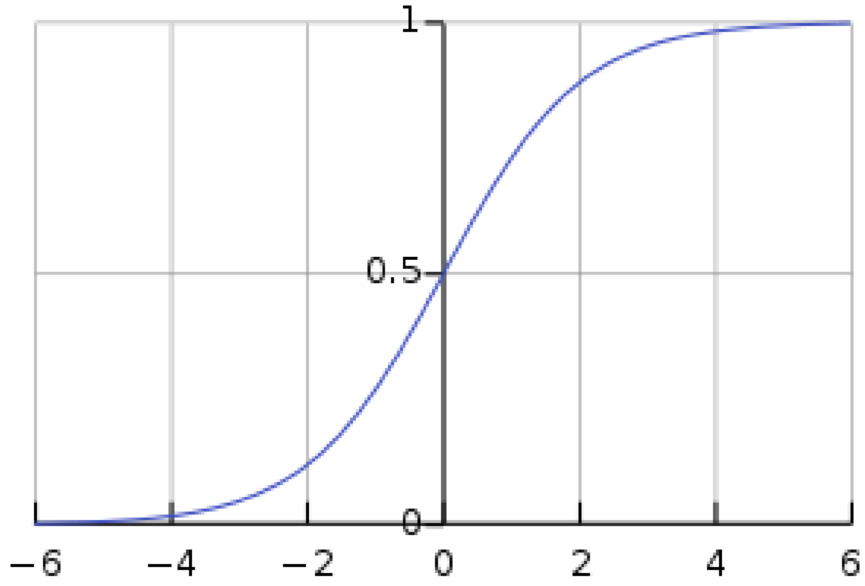
Thresholding:

predict "y = 1" if $h_\theta(x) \geq 0.5$

$$\boldsymbol{z = \theta^\top x \geq 0}$$

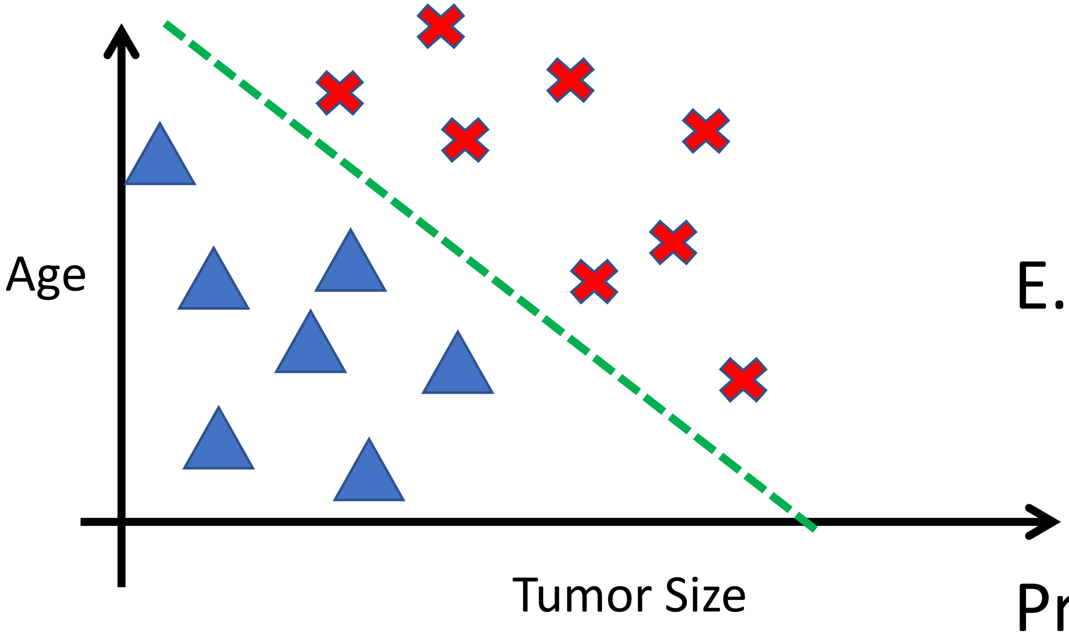
predict "y = 0" if $h_\theta(x) < 0.5$

$$\boldsymbol{z = \theta^\top x < 0}$$



Alternative Interpretation: $h_\theta(x) =$ estimated probability that $y = 1$ on input x

Decision boundary



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

E.g., $\theta_0 = -3, \theta_1 = 1,\ \theta_2 = 1$

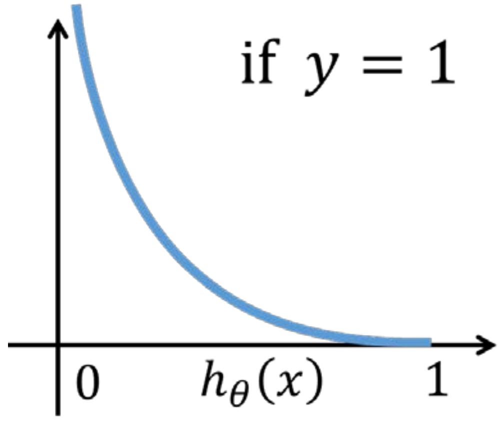
Predict "$y = 1$" if $-3 + x_1 + x_2 \geq 0$

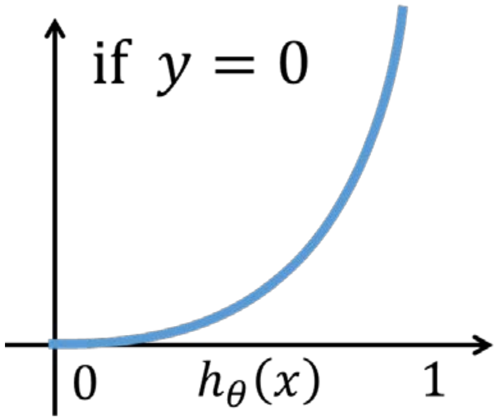
Cost function for Logistic Regression

Logistic Regression

$$\mathrm{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

$$= \boldsymbol{-y\ \log(h_\theta(x)) - (1 - y)\log(1 - h_\theta(x))}$$





$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} \mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)}))$$

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

Gradient descent

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)}\ \log\left(h_\theta\left(x^{(i)}\right)\right) + (1 - y^{(i)}) \log\left(1 - h_\theta\left(x^{(i)}\right)\right)\right]$$

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

Good news: Convex function!
Bad news: No analytical solution

Gradient descent

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)}\ \log\left(h_\theta\left(x^{(i)}\right)\right) + (1-y^{(i)})\log\left(1-h_\theta\left(x^{(i)}\right)\right)\right]$$

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

Gradient descent

(Simultaneously update all θ_j)

Repeat {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

}

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

Gradient descent for Linear Regression

Repeat {

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

}

$$\boxed{h_\theta(x) = \theta^\top x}$$

Gradient descent for Logistic Regression

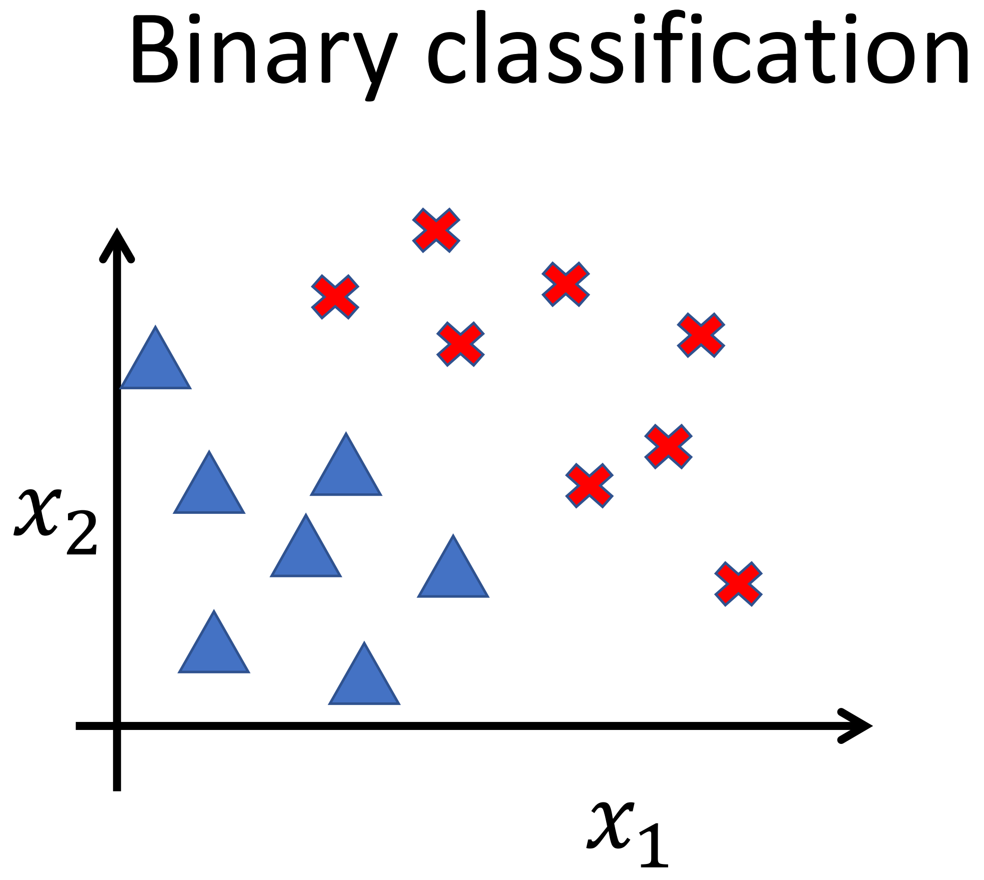
Repeat {

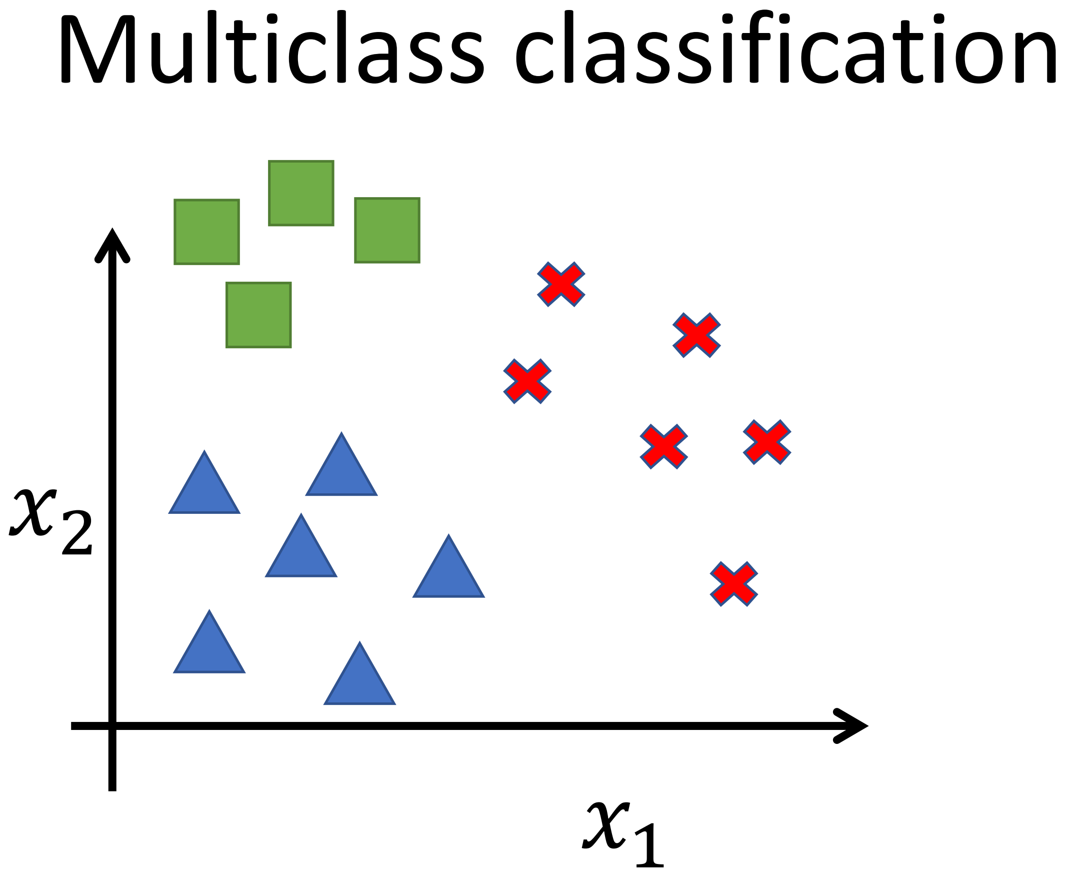
$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)} \right) x_j^{(i)}$$

}

$$\boxed{h_\theta(x) = \frac{1}{1 + e^{-\theta^\top x}}}$$

Multiclass classification





Multi-class Classification

- Multi-class Classification: y can take on K different values $\{1,2,\ldots,k\}$
- $h_\theta(x)$ estimates the probability of belonging to each class

$$P(y=k|x,\theta) \propto \exp\left(\theta_k^T x\right)$$

$$\theta = \begin{bmatrix} \vdots & \vdots & \vdots \\ \theta_1 & \theta_2 & \theta_k \\ \vdots & \vdots & \vdots \end{bmatrix}$$

$$P(y=k|x,\theta) = \frac{\exp\left(\theta_k^T x\right)}{\sum_{j=1}^{K} \exp\left(\theta_j^T x\right)}$$

$$\boxed{\mathrm{J}(\theta) = -\left[\sum_{i=1}^{m} \sum_{j=1}^{K} 1\{y^{(i)} = k\} \log \frac{\exp\left(\theta_k^T x^{(i)}\right)}{\sum_{j=1}^{K} \exp\left(\theta_j^T x^{(i)}\right)}\right]}$$