香港科技大學
THE HONG KONG
UNIVERSITY OF SCIENCE
AND TECHNOLOGY

COMP 5212
Machine Learning
Lecture 2

Supervised Learning: Regression

Junxian He
Feb 7, 2024

Announcement

Slides are now available BEFORE each lecture at our course website

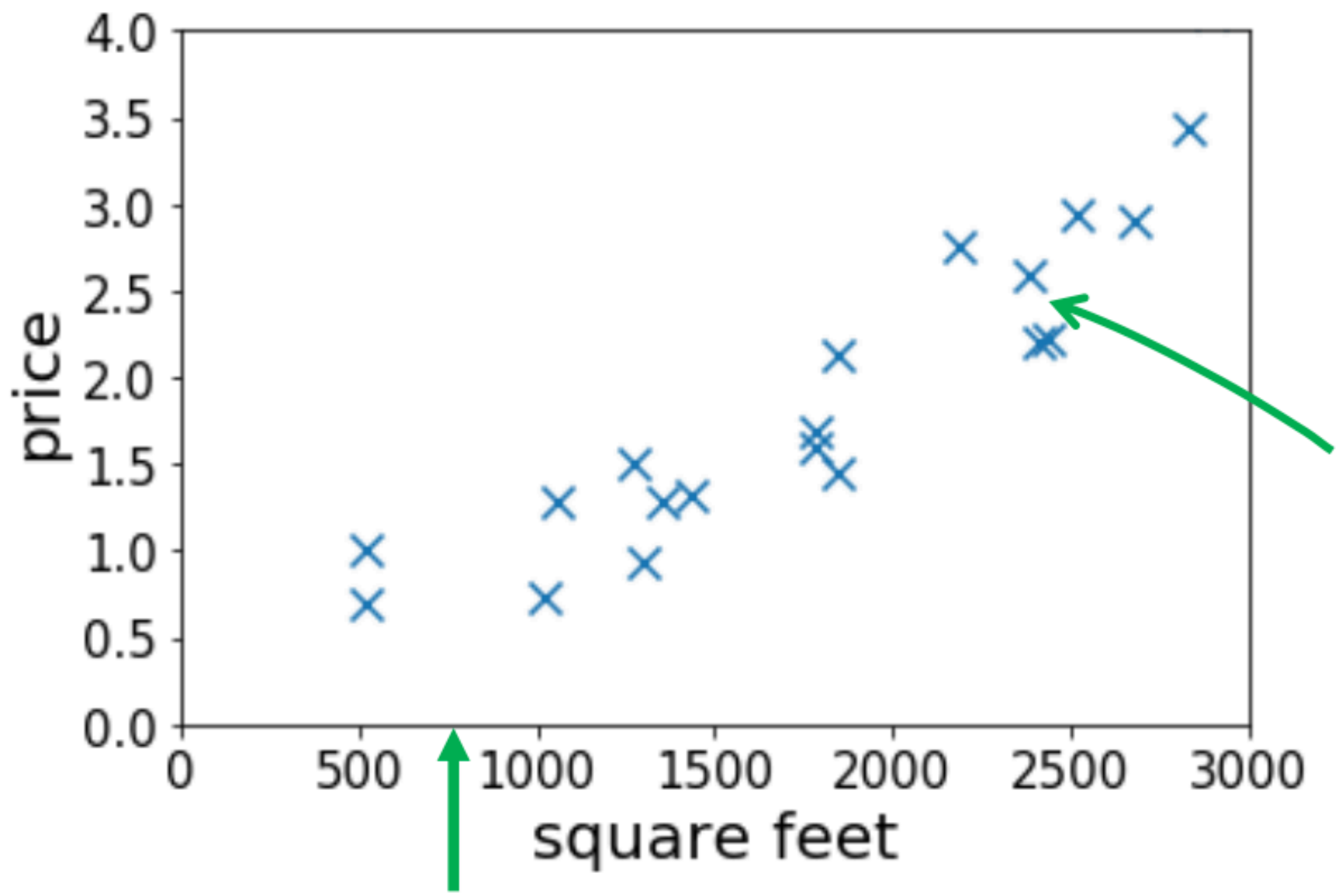
https://jxhe.github.io/teaching/comp5212s24

Supervised Learning

- A hypothesis or a prediction function is function $h : \mathcal{X} \to \mathcal{Y}$

Supervised Learning

- A hypothesis or a prediction function is function $h : \mathcal{X} \rightarrow \mathcal{Y}$

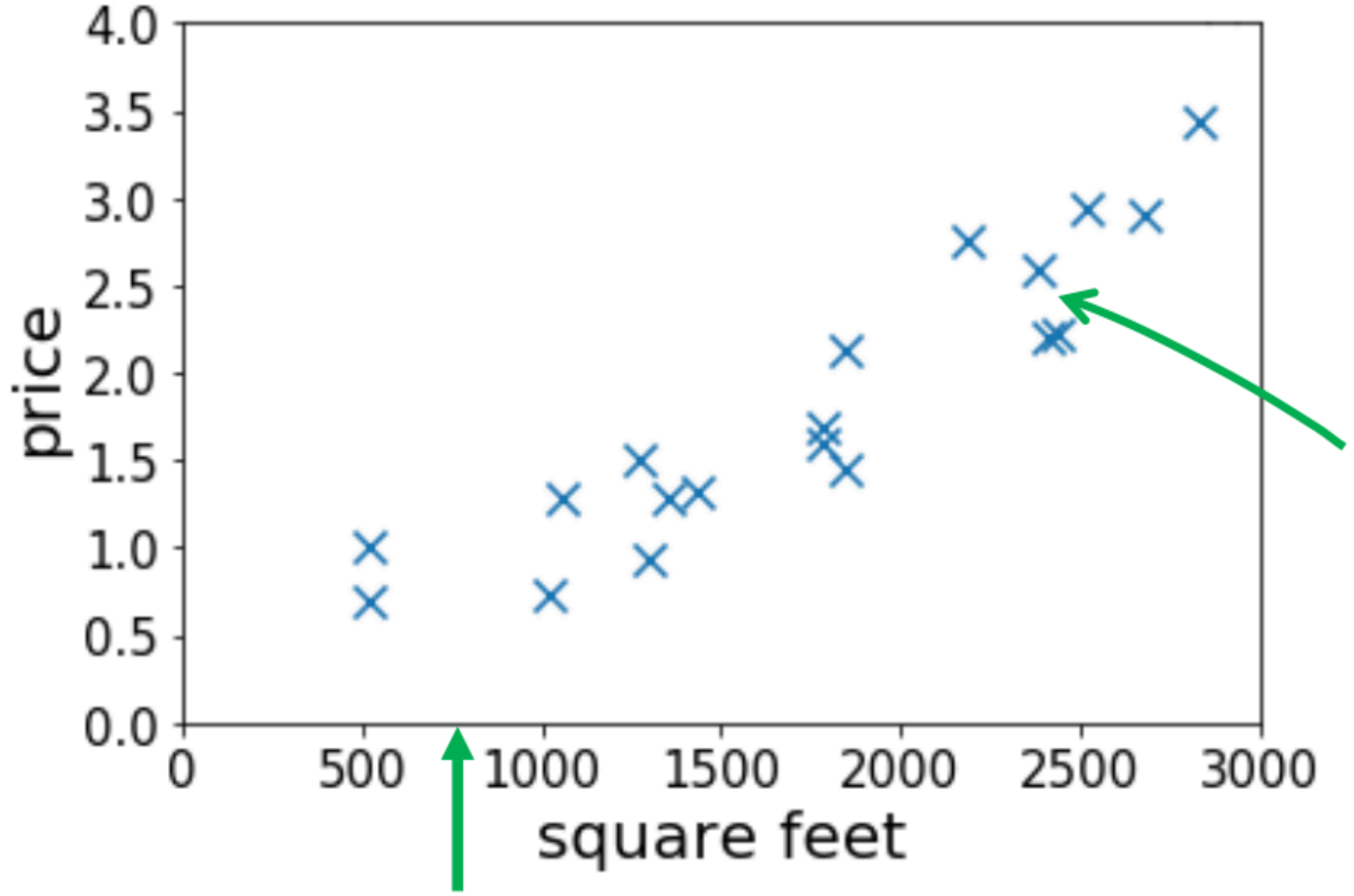


15th sample
$(x^{(15)}, y^{(15)})$

$x = 800$
$y = ?$

Supervised Learning

- A hypothesis or a prediction function is function $h : \mathcal{X} \to \mathcal{Y}$



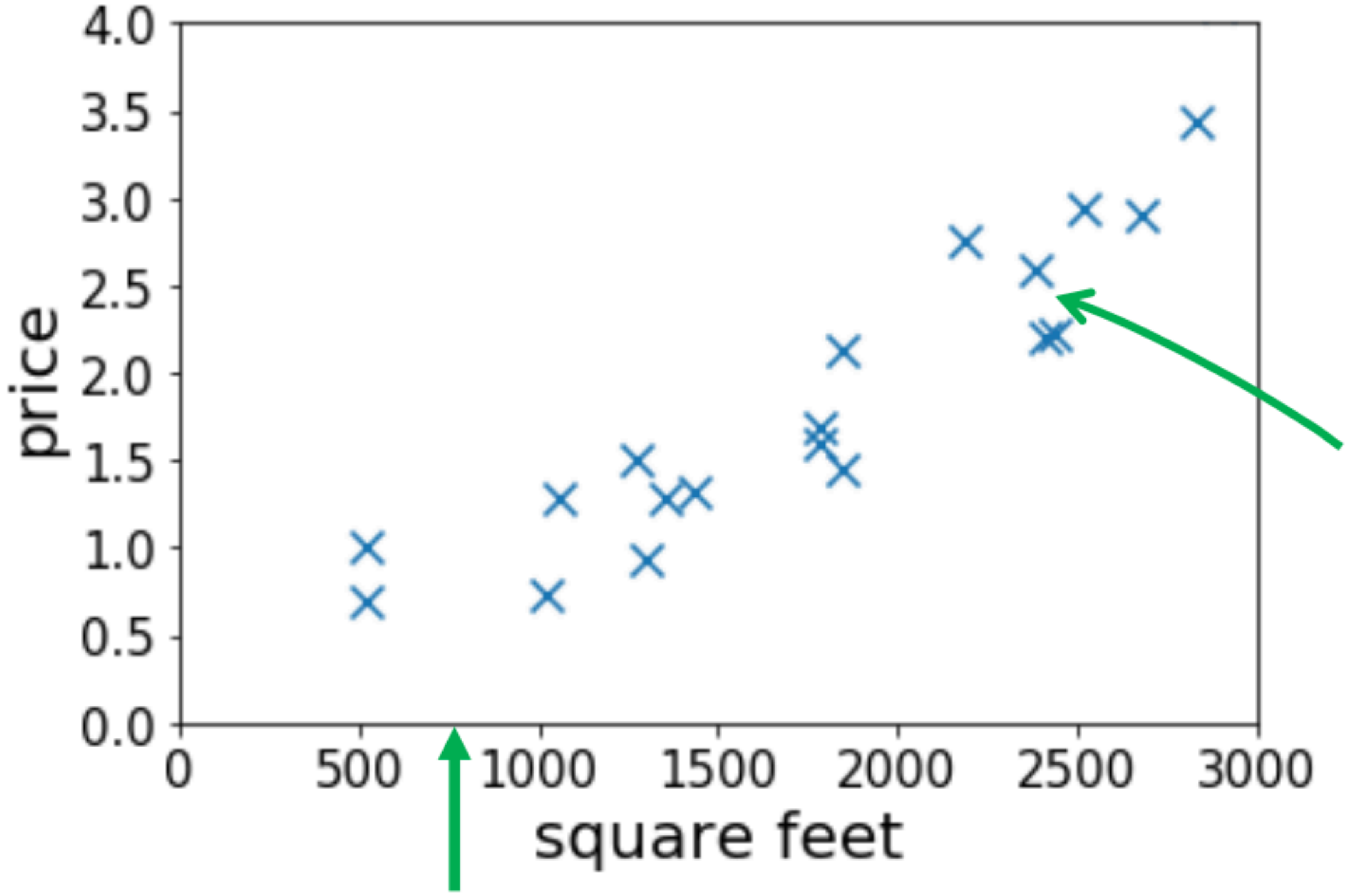
15th sample
$(x^{(15)}, y^{(15)})$

$x = 800$
$y = ?$

CAT

Supervised Learning

- A hypothesis or a prediction function is function $h : \mathcal{X} \to \mathcal{Y}$



15th sample
$(x^{(15)}, y^{(15)})$

$x = 800$
$y = ?$

X

CAT Y

Supervised Learning

- A hypothesis or a prediction function is function $h : \mathcal{X} \to \mathcal{Y}$

Supervised Learning

- A hypothesis or a prediction function is function $h : \mathcal{X} \to \mathcal{Y}$

- A training set is set of pairs $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$

  s.t. $x^{(i)} \in \mathcal{X}$ and $y^{(i)} \in \mathcal{Y}$ for $i = 1, \ldots, n$.

Supervised Learning

- A hypothesis or a prediction function is function $h : \mathcal{X} \rightarrow \mathcal{Y}$

- A training set is set of pairs $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$

  s.t. $x^{(i)} \in \mathcal{X}$ and $y^{(i)} \in \mathcal{Y}$ for $i = 1, \ldots, n$.

- Given a training set our goal is to produce a good prediction function h

Supervised Learning

- A hypothesis or a prediction function is function $h : \mathcal{X} \rightarrow \mathcal{Y}$

- A training set is set of pairs $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$
  s.t. $x^{(i)} \in \mathcal{X}$ and $y^{(i)} \in \mathcal{Y}$ for $i = 1, \ldots, n$.

- Given a training set our goal is to produce a good prediction function h

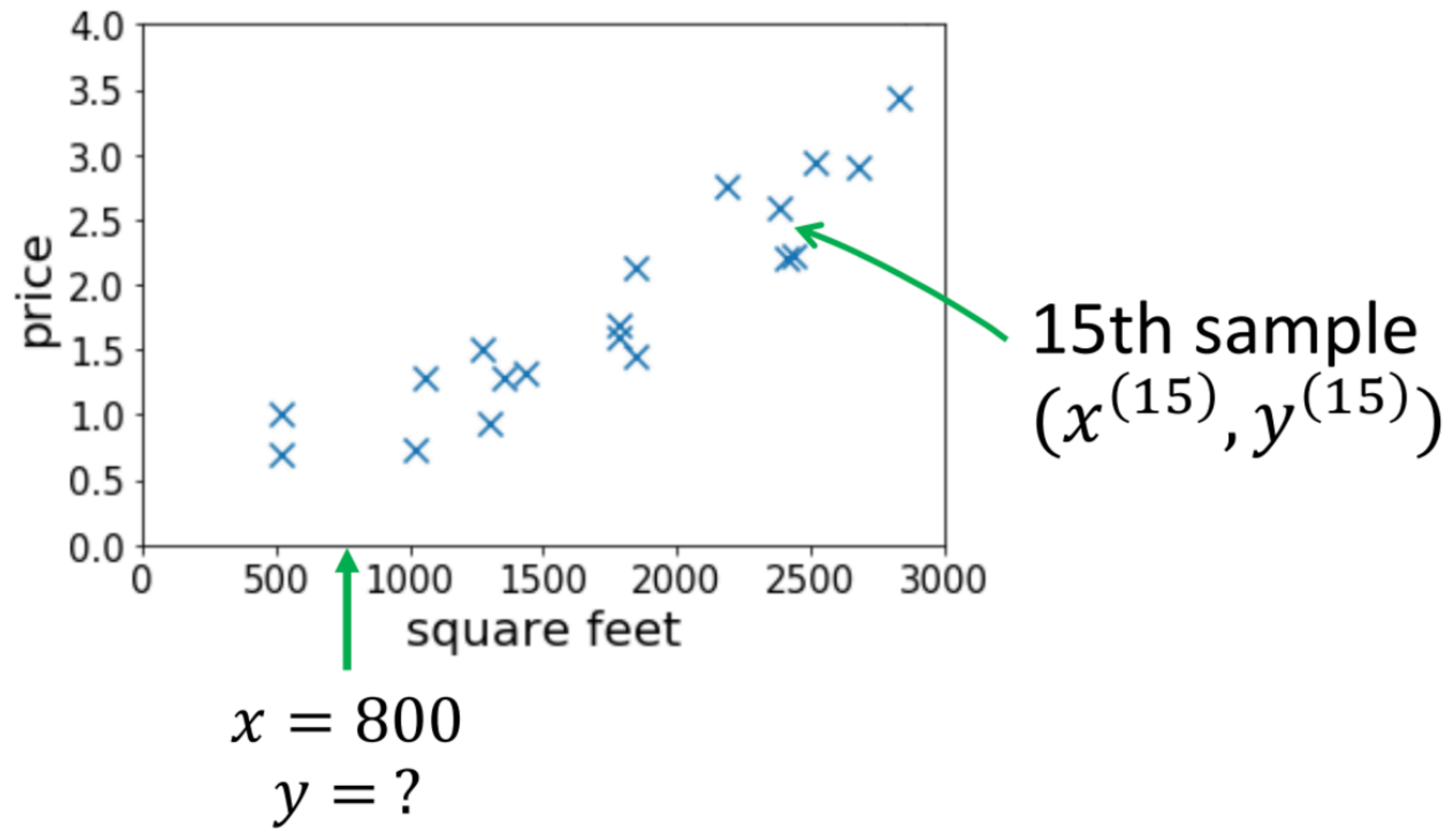
- If $\mathcal{Y}$ is continuous, then called a regression problem
- If $\mathcal{Y}$ is discrete, then called a classification problem

Supervised Learning

- How to define "good" for a prediction function?
  - Metrics / performance

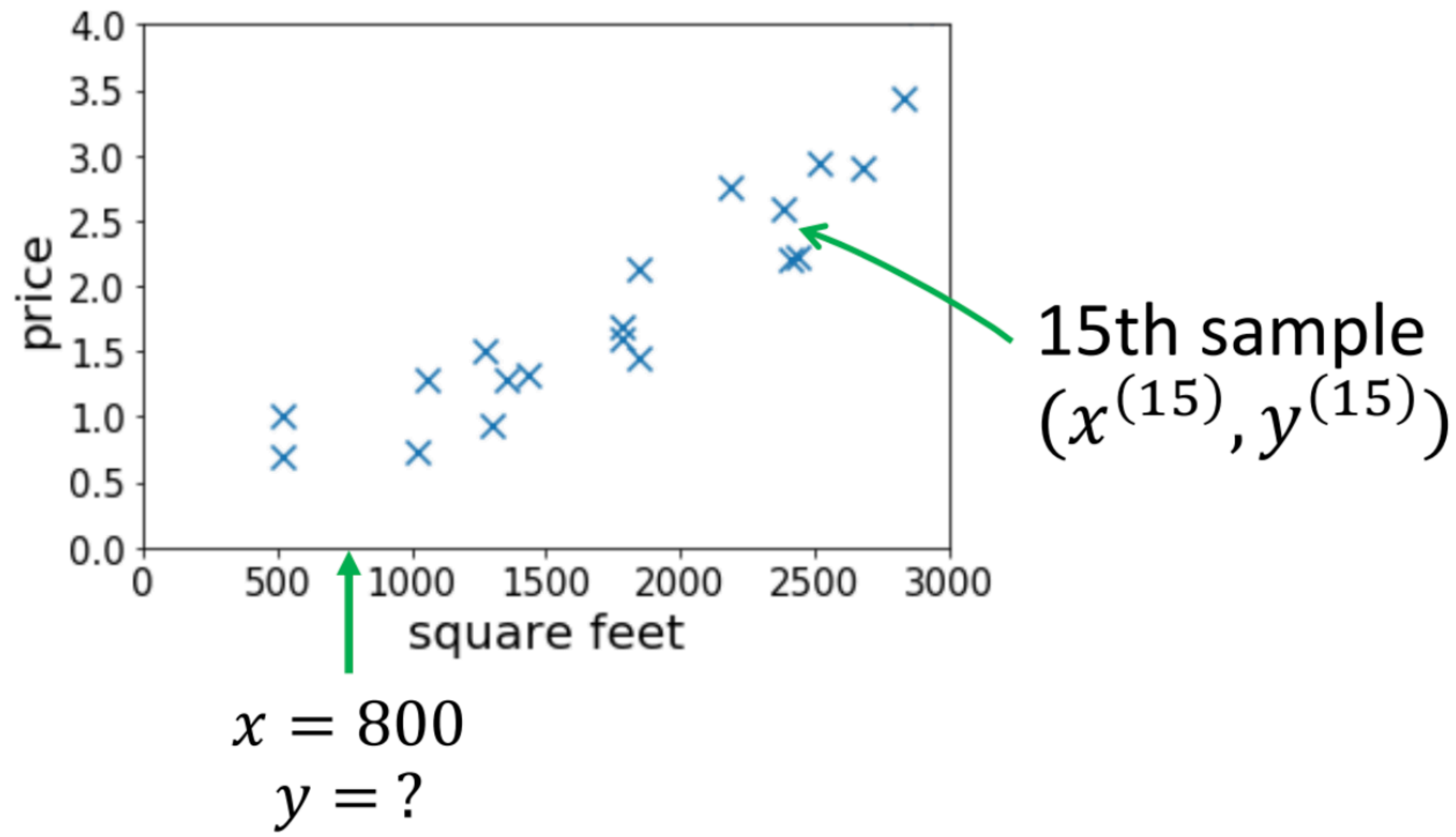
Supervised Learning

- How to define "good" for a prediction function?
  - Metrics / performance



Supervised Learning

- How to define "good" for a prediction function?
  - Metrics / performance

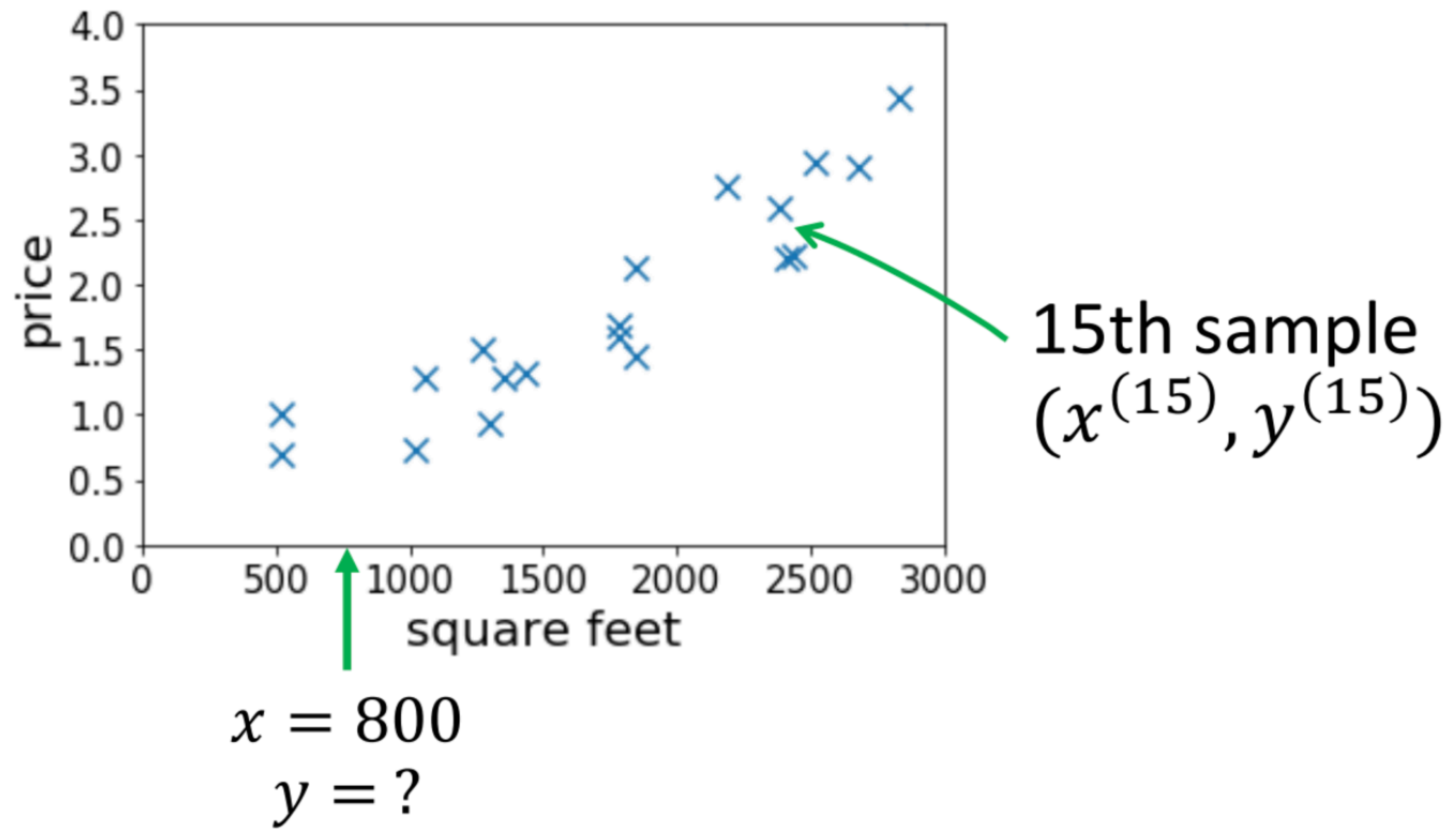


$$|\hat{y} - y*|$$

$\hat{y}$ is the prediction, $y*$ is the truth

Supervised Learning

- How to define “good” for a prediction function?
  - Metrics / performance

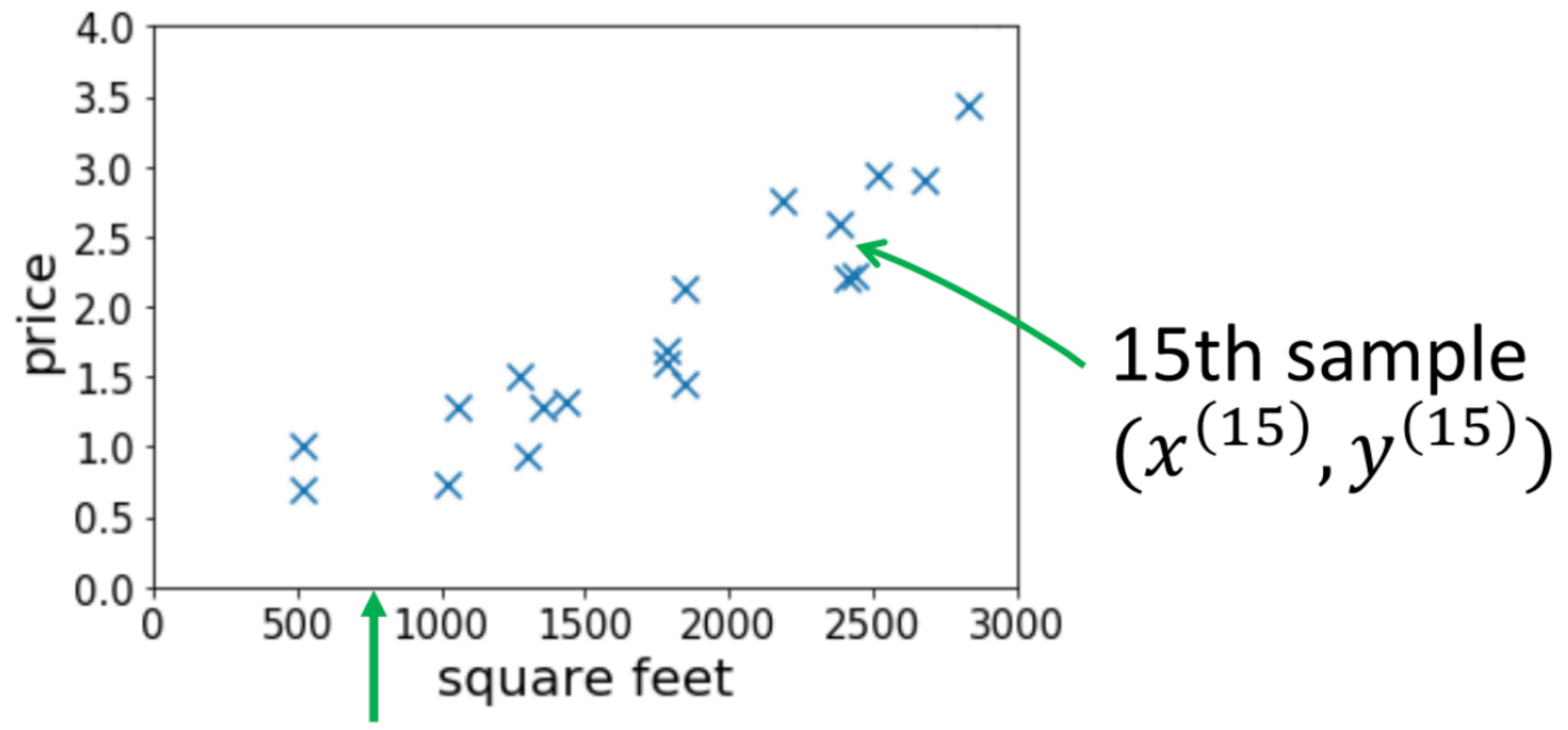


$$|\hat{y} - y*|$$

$\hat{y}$ is the prediction, $y*$ is the truth

Supervised Learning

- How to define “good” for a prediction function?
  - Metrics / performance



$x = 800$
$y = ?$

CAT

$$|\hat{y} - y*|$$

$\hat{y}$ is the prediction, $y*$ is the truth

$$\mathbb{I}(\hat{y} = y*) = \begin{cases} 1, & \hat{y} = y* \\ 0 & otherwise \end{cases}$$

Supervised Learning

- How to define “good” for a prediction function?
  - Metrics / performance
  - Good on unseen data

Supervised Learning

- How to define "good" for a prediction function?
  - Metrics / performance
  - Good on unseen data

Validation dataset is another set of pairs $\{(\hat{x}^{(1)}, \hat{y}^{(1)}), \cdots, (\hat{x}^{(m)}, \hat{y}^{(m)})\}$

Supervised Learning

- How to define “good” for a prediction function?
  - Metrics / performance
  - Good on unseen data

Validation dataset is another set of pairs $\{(\hat{x}^{(1)}, \hat{y}^{(1)}), \cdots, (\hat{x}^{(m)}, \hat{y}^{(m)})\}$

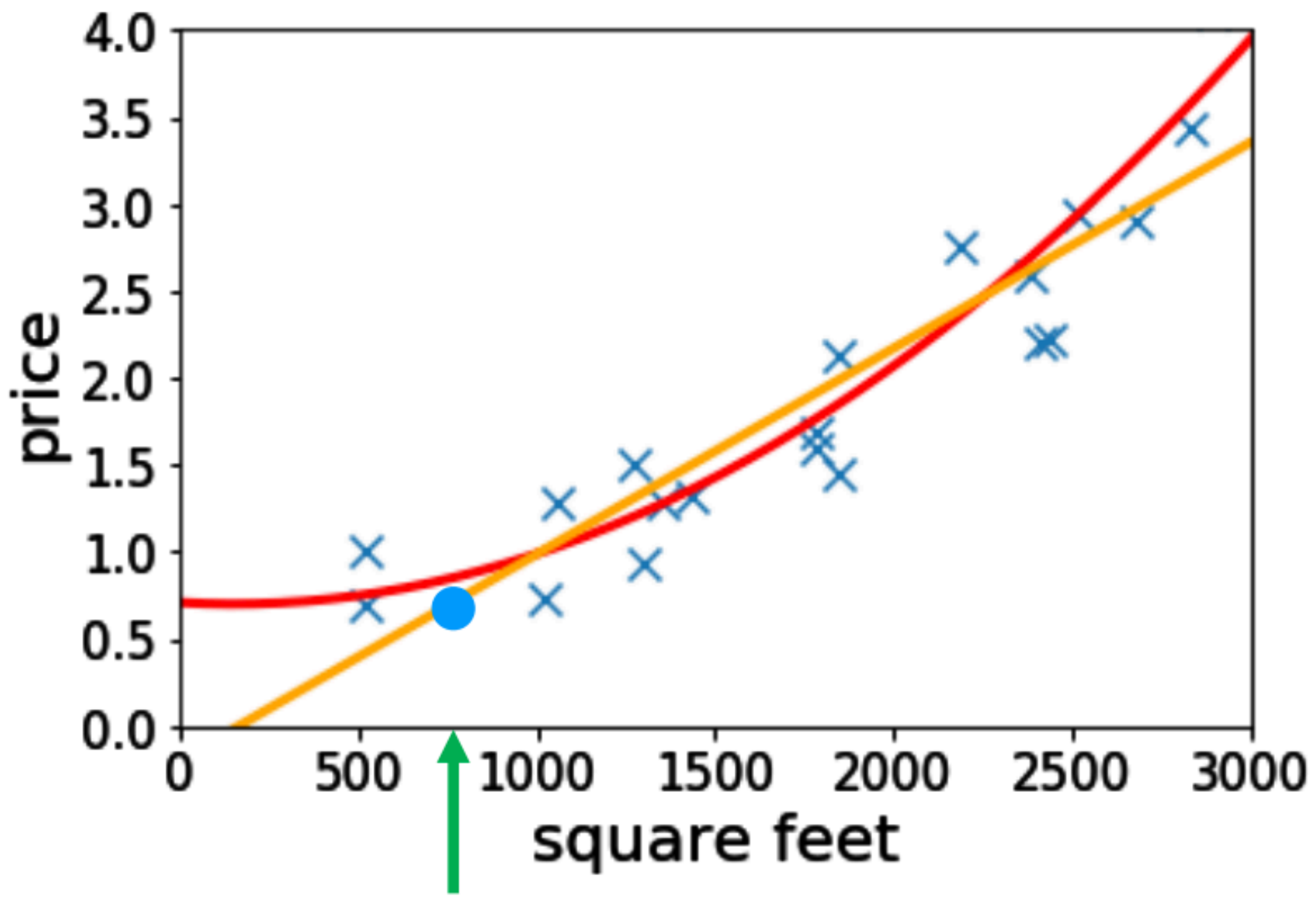
Does not overlap with training dataset

Supervised Learning

- How to define "good" for a prediction function?
  - Metrics / performance
  - Good on unseen data

Validation dataset is another set of pairs $\{(\hat{x}^{(1)}, \hat{y}^{(1)}), \cdots, (\hat{x}^{(m)}, \hat{y}^{(m)})\}$

Does not overlap with training dataset



Which curve to choose?

Supervised Learning

- How to define “good” for a prediction function?
  - Metrics / performance
  - Good on unseen data

Validation dataset is another set of pairs $\{(\hat{x}^{(1)}, \hat{y}^{(1)}), \cdots, (\hat{x}^{(m)}, \hat{y}^{(m)})\}$

Does not overlap with training dataset

Supervised Learning

- How to define “good” for a prediction function?
  - Metrics / performance
  - Good on unseen data

Validation dataset is another set of pairs $\{(\hat{x}^{(1)}, \hat{y}^{(1)}), \cdots, (\hat{x}^{(m)}, \hat{y}^{(m)})\}$

Does not overlap with training dataset

Test dataset is another set of pairs $\{(\tilde{x}^{(1)}, \tilde{y}^{(1)}), \cdots, (\tilde{x}^{(L)}, \tilde{y}^{(L)})\}$

Does not overlap with training and validation dataset

Supervised Learning

- How to define "good" for a prediction function?
  - Metrics / performance
  - Good on unseen data

Validation dataset is another set of pairs $\{(\hat{x}^{(1)}, \hat{y}^{(1)}), \cdots, (\hat{x}^{(m)}, \hat{y}^{(m)})\}$

Does not overlap with training dataset

Test dataset is another set of pairs $\{(\tilde{x}^{(1)}, \tilde{y}^{(1)}), \cdots, (\tilde{x}^{(L)}, \tilde{y}^{(L)})\}$

Does not overlap with training and validation dataset

Completely unseen before deployment

Realistic setting

Supervised Learning

- How to define “good” for a prediction function?
  - Metrics / performance
  - Good on unseen data

Validation dataset is another set of pairs $\{(\hat{x}^{(1)}, \hat{y}^{(1)}), \cdots, (\hat{x}^{(m)}, \hat{y}^{(m)})\}$

Does not overlap with training dataset

Test dataset is another set of pairs $\{(\tilde{x}^{(1)}, \tilde{y}^{(1)}), \cdots, (\tilde{x}^{(L)}, \tilde{y}^{(L)})\}$

Does not overlap with training and validation dataset

Completely unseen before deployment

Realistic setting

Hyperparameter tuning is a form of training

Supervised Training

Train

Validation

Test

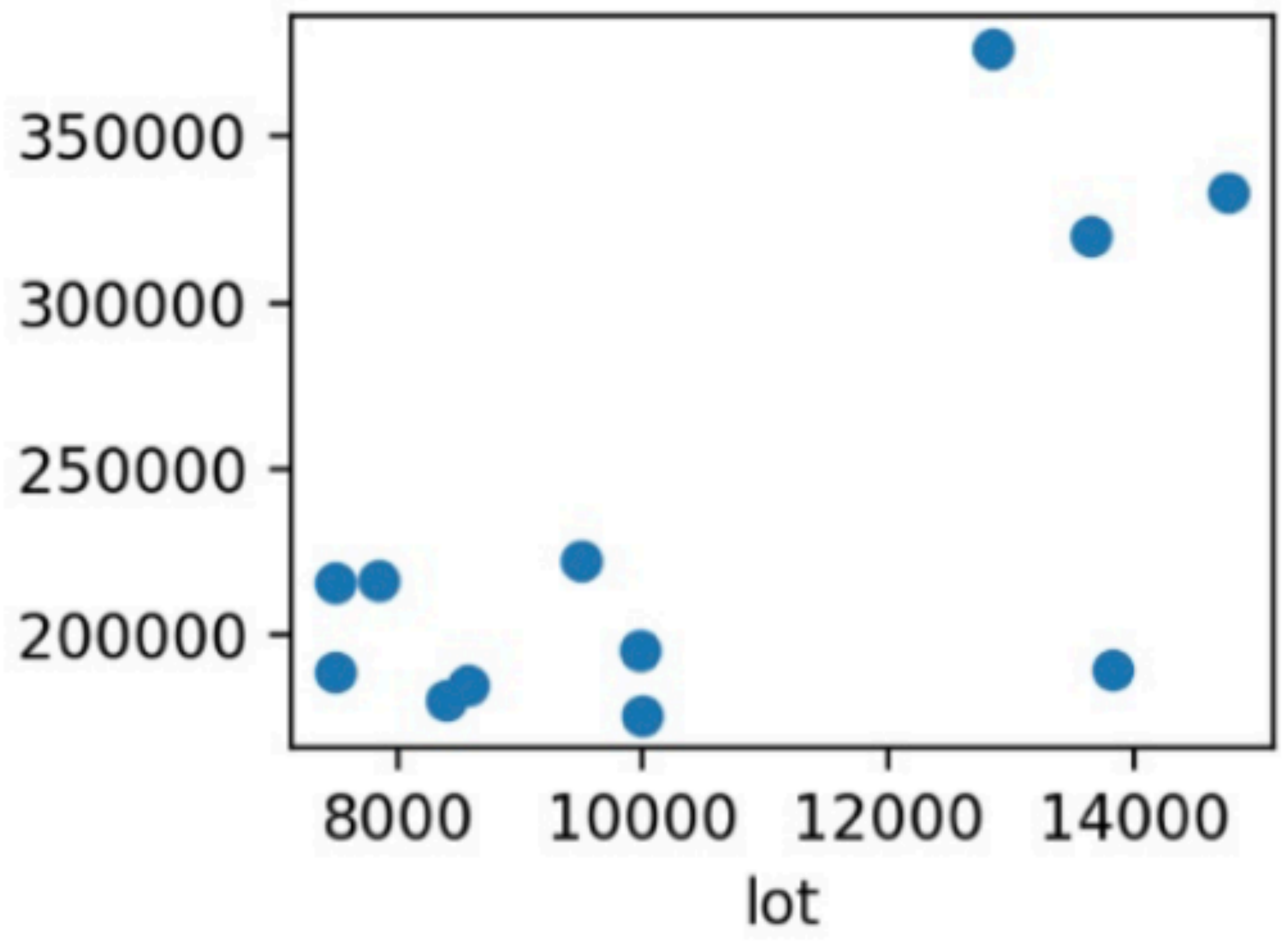
Not only for supervised learning

Example: Regression using Housing Data

Example from Stanford CS229

Example Housing Data

	SalePrice	Lot.Area
4	189900	13830
5	195500	9978
9	189000	7500
10	175900	10000
12	180400	8402
22	216000	7500
36	376162	12858
47	320000	13650
55	216500	7851
56	185088	8577



Represent h as a Linear Function

$h(x) = \theta_0 + \theta_1 x_1$ is an *affine function*

Popular choice

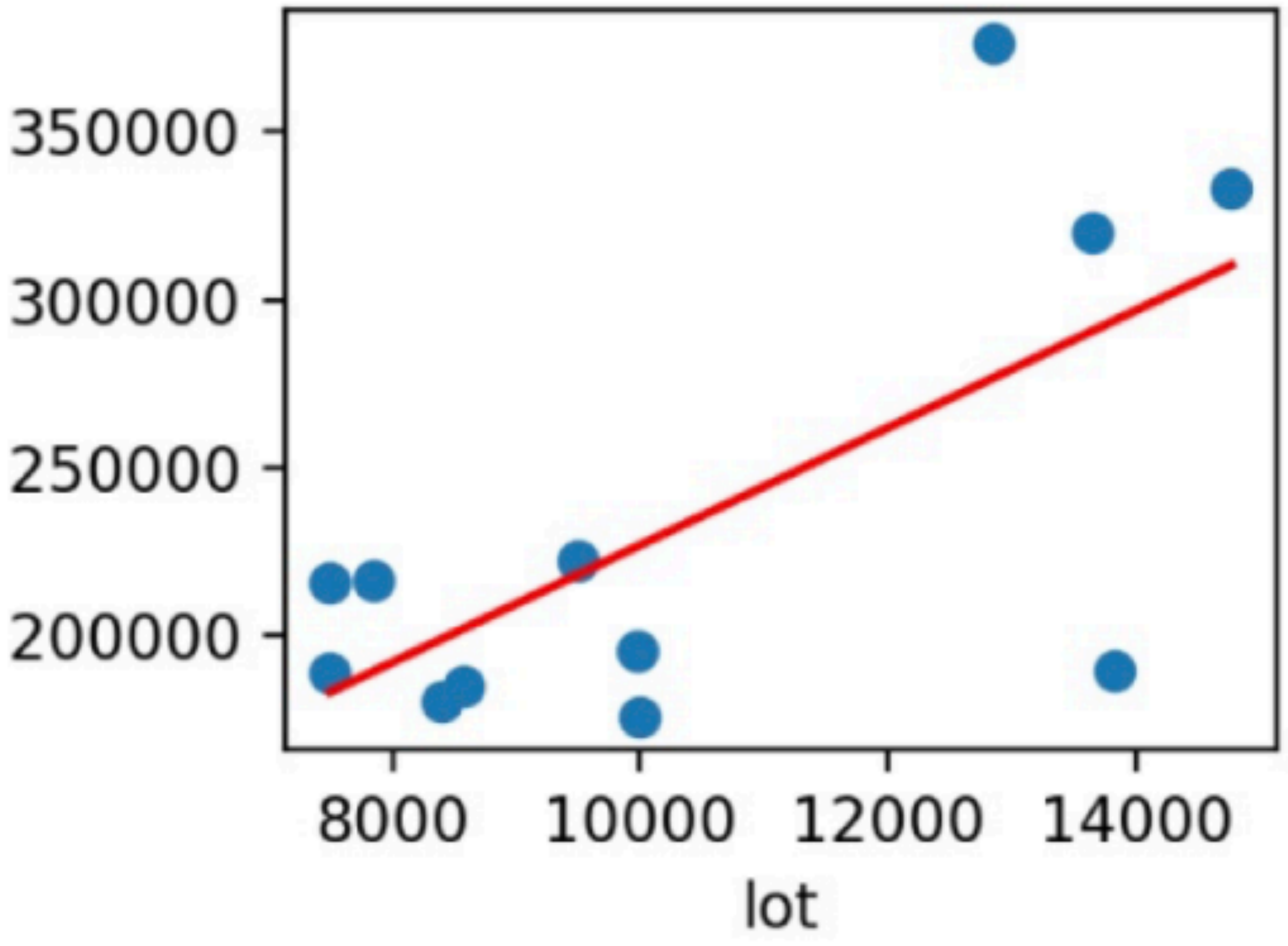
Represent h as a Linear Function

$h(x) = \theta_0 + \theta_1 x_1$ is an *affine function*

Popular choice

The function is defined by **parameters** θ_0 and θ_1, the function space is greatly reduced

Simple Line Fit



More Features

	size	bedrooms	lot size
$x^{(1)}$	2104	4	45k
$x^{(2)}$	2500	3	30k

	Price
$y^{(1)}$	400
$y^{(2)}$	900

More Features

	size	bedrooms	lot size
$x^{(1)}$	2104	4	45k
$x^{(2)}$	2500	3	30k

	Price
$y^{(1)}$	400
$y^{(2)}$	900

What's a prediction here?

$$h(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3.$$

More Features

	size	bedrooms	lot size
$x^{(1)}$	2104	4	45k
$x^{(2)}$	2500	3	30k

	Price
$y^{(1)}$	400
$y^{(2)}$	900

What's a prediction here?

$$h(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3.$$

With the convention that $x_0 = 1$ we can write:

$$h(x) = \sum_{j=0}^{3} \theta_j x_j$$

Vector Notations

Vector Notations

	size	bedrooms	lot size		Price
$x^{(1)}$	2104	4	45k	$y^{(1)}$	400
$x^{(2)}$	2500	3	30k	$y^{(2)}$	900

We write the vectors as (important notation)

$$\theta = \begin{pmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \end{pmatrix} \text{ and } x^{(1)} = \begin{pmatrix} x_0^{(1)} \\ x_1^{(1)} \\ x_2^{(1)} \\ x_3^{(1)} \end{pmatrix} = \begin{pmatrix} 1 \\ 2104 \\ 4 \\ 45 \end{pmatrix} \text{ and } y^{(1)} = 400$$

Vector Notations

	size	bedrooms	lot size		Price
$x^{(1)}$	2104	4	45k	$y^{(1)}$	400
$x^{(2)}$	2500	3	30k	$y^{(2)}$	900

We write the vectors as (important notation)

$$\theta = \begin{pmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \end{pmatrix} \text{ and } x^{(1)} = \begin{pmatrix} x_0^{(1)} \\ x_1^{(1)} \\ x_2^{(1)} \\ x_3^{(1)} \end{pmatrix} = \begin{pmatrix} 1 \\ 2104 \\ 4 \\ 45 \end{pmatrix} \text{ and } y^{(1)} = 400$$

We call θ **parameters**, $x^{(i)}$ is the input or the **features**, and the output or **target** is $y^{(i)}$. To be clear,

(x, y) is a training example and $(x^{(i)}, y^{(i)})$ is the i^{th} example.

Vector Notations

	size	bedrooms	lot size		Price
$x^{(1)}$	2104	4	45k	$y^{(1)}$	400
$x^{(2)}$	2500	3	30k	$y^{(2)}$	900

We write the vectors as (important notation)

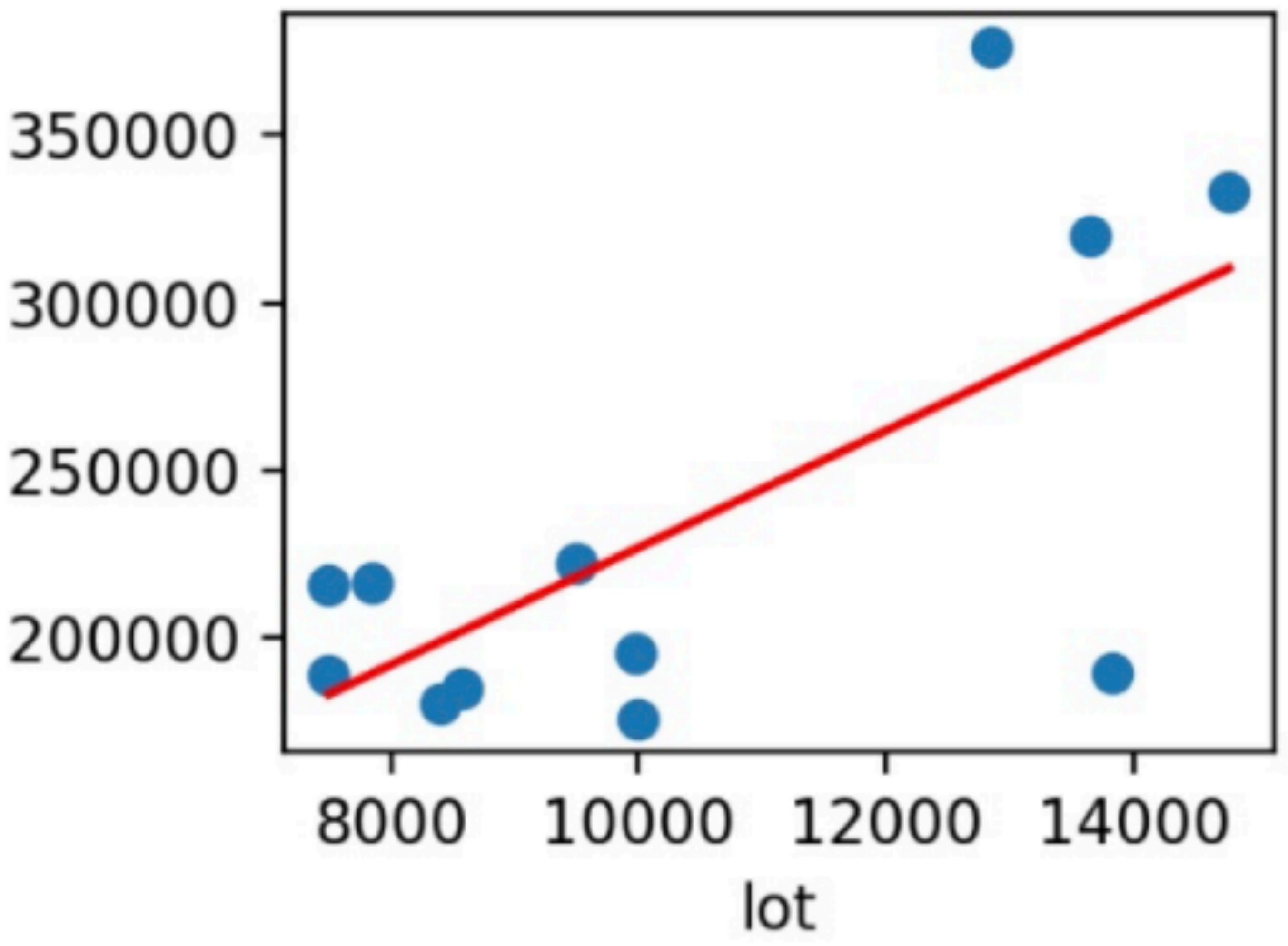
$$\theta = \begin{pmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \end{pmatrix} \text{ and } x^{(1)} = \begin{pmatrix} x_0^{(1)} \\ x_1^{(1)} \\ x_2^{(1)} \\ x_3^{(1)} \end{pmatrix} = \begin{pmatrix} 1 \\ 2104 \\ 4 \\ 45 \end{pmatrix} \text{ and } y^{(1)} = 400$$

We call θ **parameters**, $x^{(i)}$ is the input or the **features**, and the output or **target** is $y^{(i)}$. To be clear,

(x, y) is a training example and $(x^{(i)}, y^{(i)})$ is the i^{th} example.

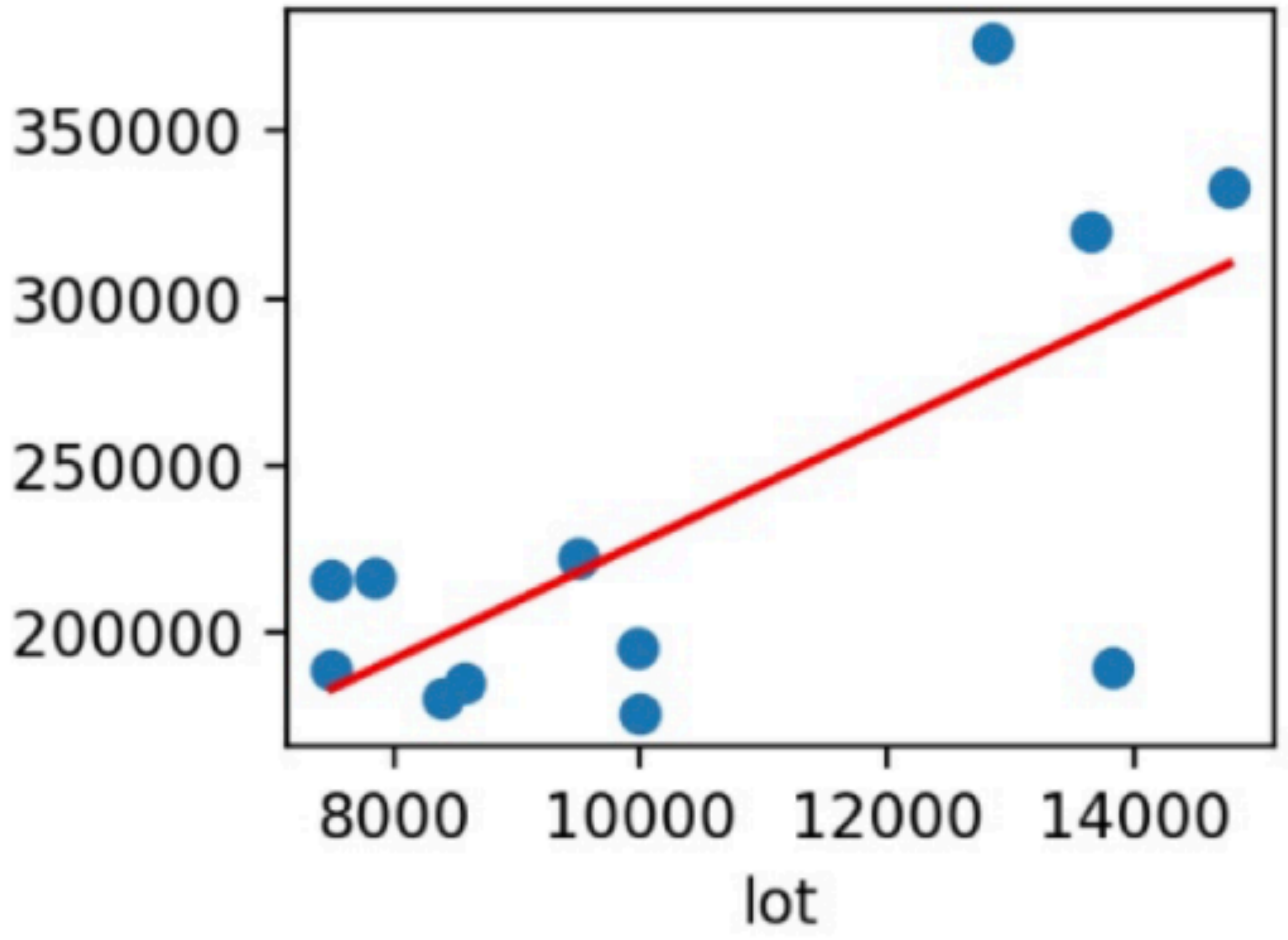
We have n examples. There are d features. $x^{(i)}$ and θ are d+1 dimensional (since $x_0 = 1$)

Vector Notation of Prediction



$$h_\theta(x) = \sum_{j=0}^{d} \theta_j x_j = x^T \theta$$

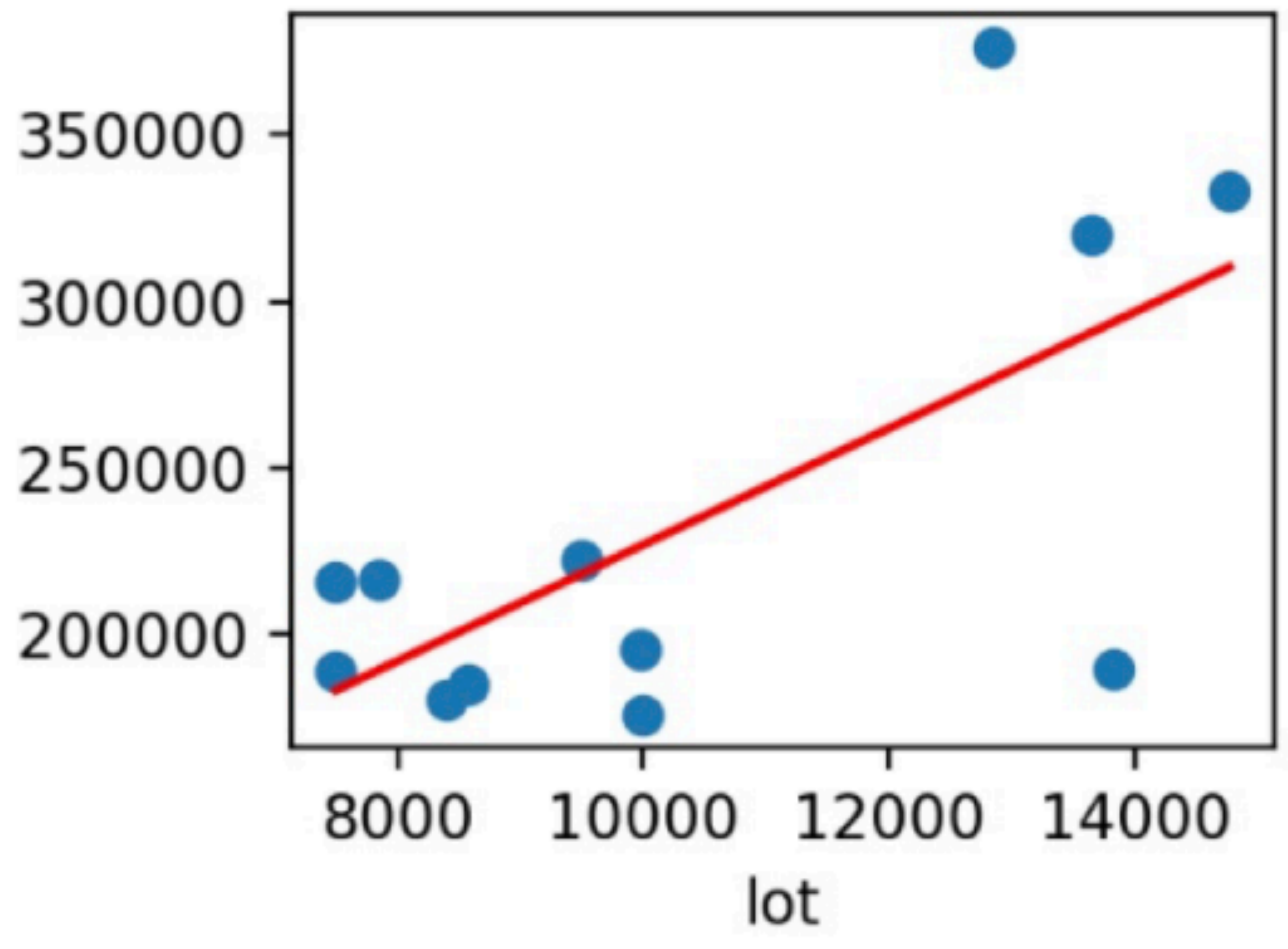
Vector Notation of Prediction



$$h_\theta(x) = \sum_{j=0}^{d} \theta_j x_j = x^T \theta$$

We want to choose θ so that $h_\theta(x) \approx y$

Loss Function

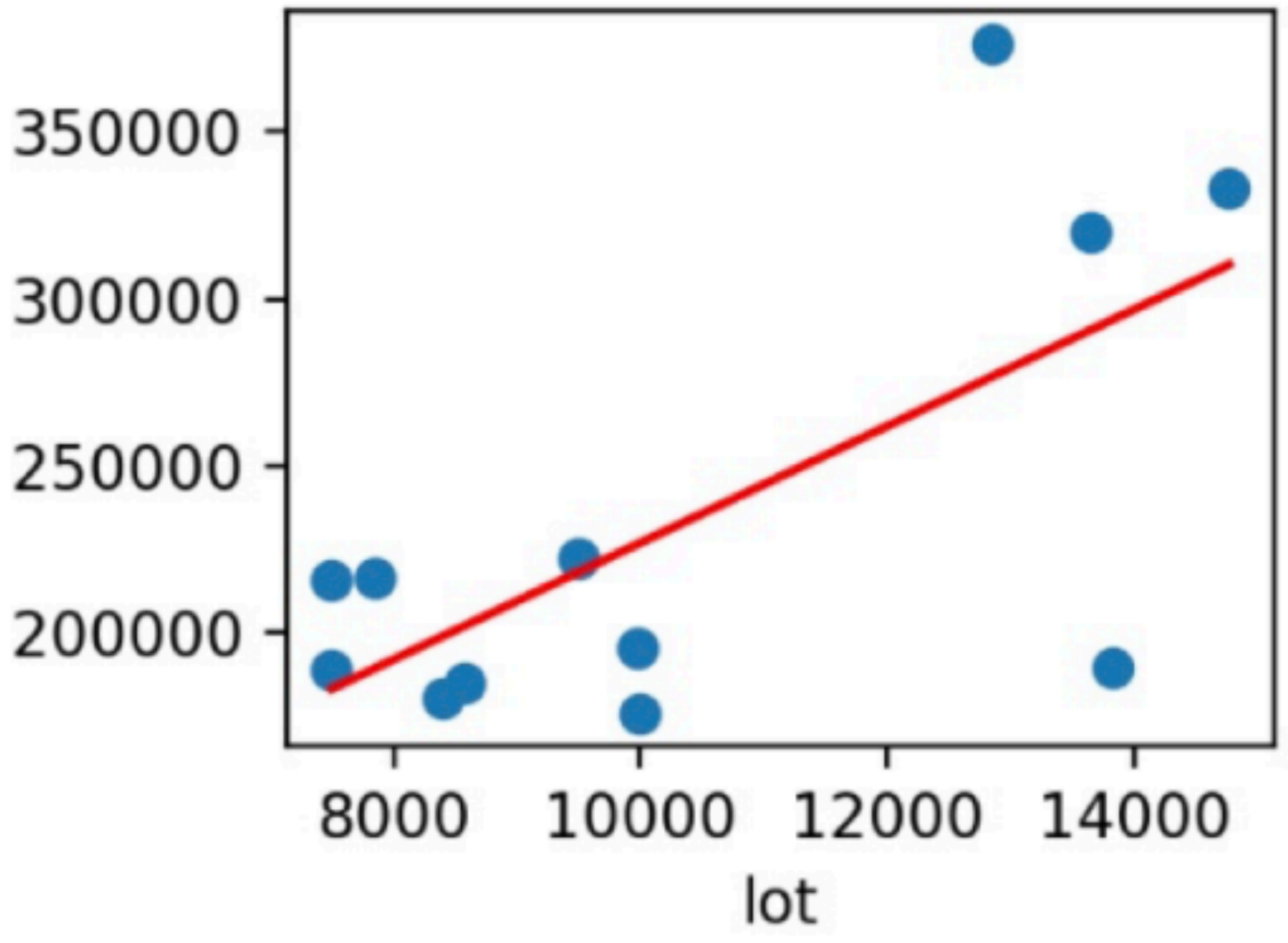


$$h_\theta(x) = \sum_{j=0}^{d} \theta_j x_j = x^T \theta$$

We want to choose θ so that $h_\theta(x) \approx y$

How to quantify the deviation of $h_\theta(x)$ from y

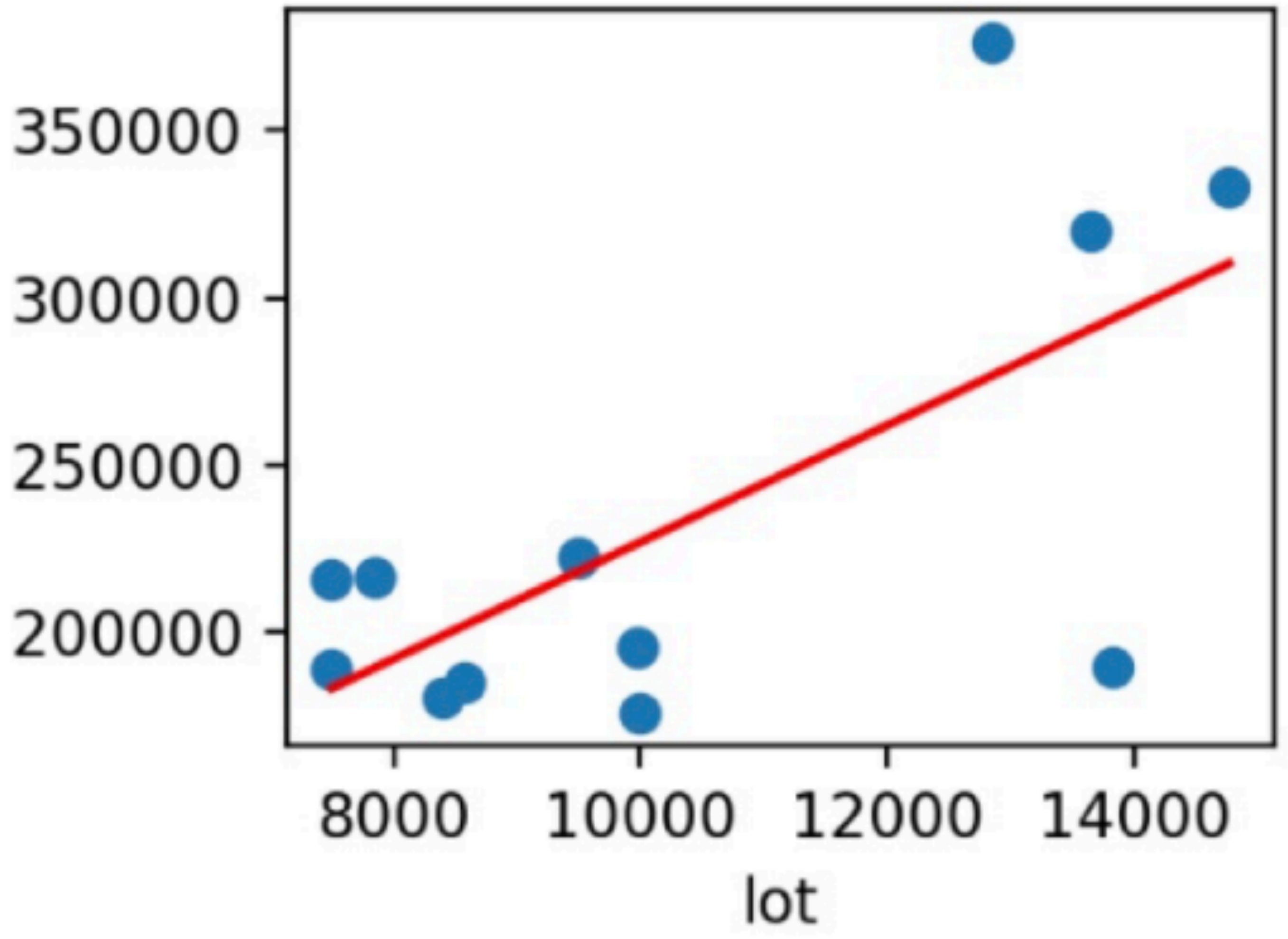
Least Squares



$$h_\theta(x) = \sum_{j=0}^{d} \theta_j x_j = x^T\theta$$

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

Least Squares



$$h_\theta(x) = \sum_{j=0}^{d} \theta_j x_j = x^T\theta$$

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

Choose

$$\theta = \operatorname*{argmin}_\theta J(\theta).$$

Questions?

Solving Least Square Problem

Direct Minimization

$$h_\theta(x) = \sum_{j=0}^{d} \theta_j x_j = x^T\theta \qquad J(\theta) = \frac{1}{2}\sum_{i=1}^{n}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

Choose

$$\theta = \underset{\theta}{\operatorname{argmin}}\, J(\theta).$$

Solving Least Square Problem

$$\begin{aligned}
\nabla_\theta J(\theta) &= \nabla_\theta \frac{1}{2}(X\theta - \vec{y})^T(X\theta - \vec{y}) \\
&= \frac{1}{2}\nabla_\theta \left((X\theta)^T X\theta - (X\theta)^T\vec{y} - \vec{y}^T(X\theta) + \vec{y}^T\vec{y}\right) \\
&= \frac{1}{2}\nabla_\theta \left(\theta^T(X^TX)\theta - \vec{y}^T(X\theta) - \vec{y}^T(X\theta)\right) \\
&= \frac{1}{2}\nabla_\theta \left(\theta^T(X^TX)\theta - 2(X^T\vec{y})^T\theta\right) \\
&= \frac{1}{2}\left(2X^TX\theta - 2X^T\vec{y}\right) \\
&= X^TX\theta - X^T\vec{y}
\end{aligned}$$

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

Solving Least Square Problem

$$
\begin{aligned}
\nabla_\theta J(\theta) &= \nabla_\theta \frac{1}{2}(X\theta - \vec{y})^T(X\theta - \vec{y}) \\
&= \frac{1}{2}\nabla_\theta \left((X\theta)^T X\theta - (X\theta)^T\vec{y} - \vec{y}^T(X\theta) + \vec{y}^T\vec{y}\right) \\
&= \frac{1}{2}\nabla_\theta \left(\theta^T(X^TX)\theta - \vec{y}^T(X\theta) - \vec{y}^T(X\theta)\right) \\
&= \frac{1}{2}\nabla_\theta \left(\theta^T(X^TX)\theta - 2(X^T\vec{y})^T\theta\right) \\
&= \frac{1}{2}\left(2X^TX\theta - 2X^T\vec{y}\right) \\
&= X^TX\theta - X^T\vec{y}
\end{aligned}
$$

Normal equations $X^TX\theta = X^T\vec{y}$ $\qquad$ $\theta = (X^TX)^{-1}X^T\vec{y}.$

Solving Least Square Problem

$$
\begin{aligned}
\nabla_\theta J(\theta) &= \nabla_\theta \frac{1}{2}(X\theta - \vec{y})^T(X\theta - \vec{y}) \\
&= \frac{1}{2}\nabla_\theta \left((X\theta)^T X\theta - (X\theta)^T\vec{y} - \vec{y}^T(X\theta) + \vec{y}^T\vec{y}\right) \\
&= \frac{1}{2}\nabla_\theta \left(\theta^T(X^TX)\theta - \vec{y}^T(X\theta) - \vec{y}^T(X\theta)\right) \\
&= \frac{1}{2}\nabla_\theta \left(\theta^T(X^TX)\theta - 2(X^T\vec{y})^T\theta\right) \\
&= \frac{1}{2}\left(2X^TX\theta - 2X^T\vec{y}\right) \\
&= X^TX\theta - X^T\vec{y}
\end{aligned}
$$

Normal equations $X^TX\theta = X^T\vec{y}$ $\theta = (X^TX)^{-1}X^T\vec{y}$.

When is X^TX invertible? What if it is not invertible?

Why Least-Square Loss Function?

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

Why Least-Square Loss Function?

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

Assume

$$y^{(i)} = \theta^T x^{(i)} + \epsilon^{(i)}$$

Why Least-Square Loss Function?

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

Assume

$$y^{(i)} = \theta^T x^{(i)} + \epsilon^{(i)}$$

x, y: random variable

Why Least-Square Loss Function?

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

Assume

$$y^{(i)} = \theta^T x^{(i)} + \epsilon^{(i)}$$

x, y: random variable

ϵ: deviation of prediction from the truth, Gaussian random variable

Why Least-Square Loss Function?

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

Assume

$$y^{(i)} = \theta^T x^{(i)} + \epsilon^{(i)}$$

x, y: random variable

ϵ: deviation of prediction from the truth, Gaussian random variable

$x^{(i)}, y^{(i)}$: observations, or the data

Why Least-Square Loss Function?

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

Assume

$$y^{(i)} = \theta^T x^{(i)} + \epsilon^{(i)}$$

x, y: random variable

ϵ: deviation of prediction from the truth, Gaussian random variable

$x^{(i)}, y^{(i)}$: observations, or the data

$\epsilon^{(i)}$: the actual prediction error of the i_{th} example, sampled from the Gaussian distribution, IID (independently and identically distributed)

Why Least-Square Loss Function?

Why Least-Square Loss Function?

$$p(\epsilon^{(i)}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\epsilon^{(i)})^2}{2\sigma^2}\right)$$

Why Least-Square Loss Function?

$$p(\epsilon^{(i)}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\epsilon^{(i)})^2}{2\sigma^2}\right)$$

$$p(y^{(i)}|x^{(i)};\theta) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right)$$

Why Least-Square Loss Function?

$$p(\epsilon^{(i)}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\epsilon^{(i)})^2}{2\sigma^2}\right)$$

$$p(y^{(i)}|x^{(i)};\theta) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right)$$

$$\begin{aligned} p(\vec{y}|X;\theta) &= \prod_{i=1}^{n} p(y^{(i)} \mid x^{(i)};\theta) \\ &= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right) \end{aligned}$$

Why Least-Square Loss Function?

$$p(\epsilon^{(i)}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\epsilon^{(i)})^2}{2\sigma^2}\right)$$

$$p(y^{(i)}|x^{(i)};\theta) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right)$$

$$p(\vec{y}|X;\theta) = \prod_{i=1}^{n} p(y^{(i)} \mid x^{(i)};\theta)$$

Function of θ

$$= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right)$$

Why Least-Square Loss Function?

Why Least-Square Loss Function?

$$
\begin{aligned}
L(\theta) &= \prod_{i=1}^{n} p(y^{(i)} \mid x^{(i)}; \theta) \\
&= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2} \right)
\end{aligned}
$$

Why Least-Square Loss Function?

$$
\begin{aligned}
L(\theta) &= \prod_{i=1}^{n} p(y^{(i)} \mid x^{(i)}; \theta) \\
&= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right)
\end{aligned}
$$

Likelihood Function

Why Least-Square Loss Function?

$$\begin{aligned} L(\theta) &= \prod_{i=1}^{n} p(y^{(i)} \mid x^{(i)}; \theta) \\ &= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right) \end{aligned}$$

Likelihood Function

What is a reasonable guess of θ?

Why Least-Square Loss Function?

$$
\begin{aligned}
L(\theta) &= \prod_{i=1}^{n} p(y^{(i)} \mid x^{(i)}; \theta) \\
&= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right)
\end{aligned}
$$

Likelihood Function

What is a reasonable guess of θ?

Maximize the probability of Y's happening!

Maximum Likelihood Estimation (MLE)

$$\begin{aligned}
\ell(\theta) &= \log L(\theta) \\
&= \log \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right) \\
&= \sum_{i=1}^{n} \log \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right) \\
&= n \log \frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{\sigma^2} \cdot \frac{1}{2} \sum_{i=1}^{n} (y^{(i)} - \theta^T x^{(i)})^2.
\end{aligned}$$

Why MLE?

$$
\begin{aligned}
L(\theta) &= \prod_{i=1}^{n} p(y^{(i)} \mid x^{(i)}; \theta) \\
&= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right)
\end{aligned}
$$

Likelihood Function

What is a reasonable guess of θ?

Maximize the probability of Y's happening?

Why MLE?

$$
\begin{aligned}
L(\theta) &= \prod_{i=1}^{n} p(y^{(i)} \mid x^{(i)}; \theta) \\
&= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right)
\end{aligned}
$$

Likelihood Function

What is a reasonable guess of θ?

Maximize the probability of Y's happening?

Maximizing likelihood estimation -> $\hat{\theta}$

Why MLE?

$$\begin{aligned} L(\theta) &= \prod_{i=1}^{n} p(y^{(i)} \mid x^{(i)}; \theta) \\ &= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right) \end{aligned}$$

Likelihood Function

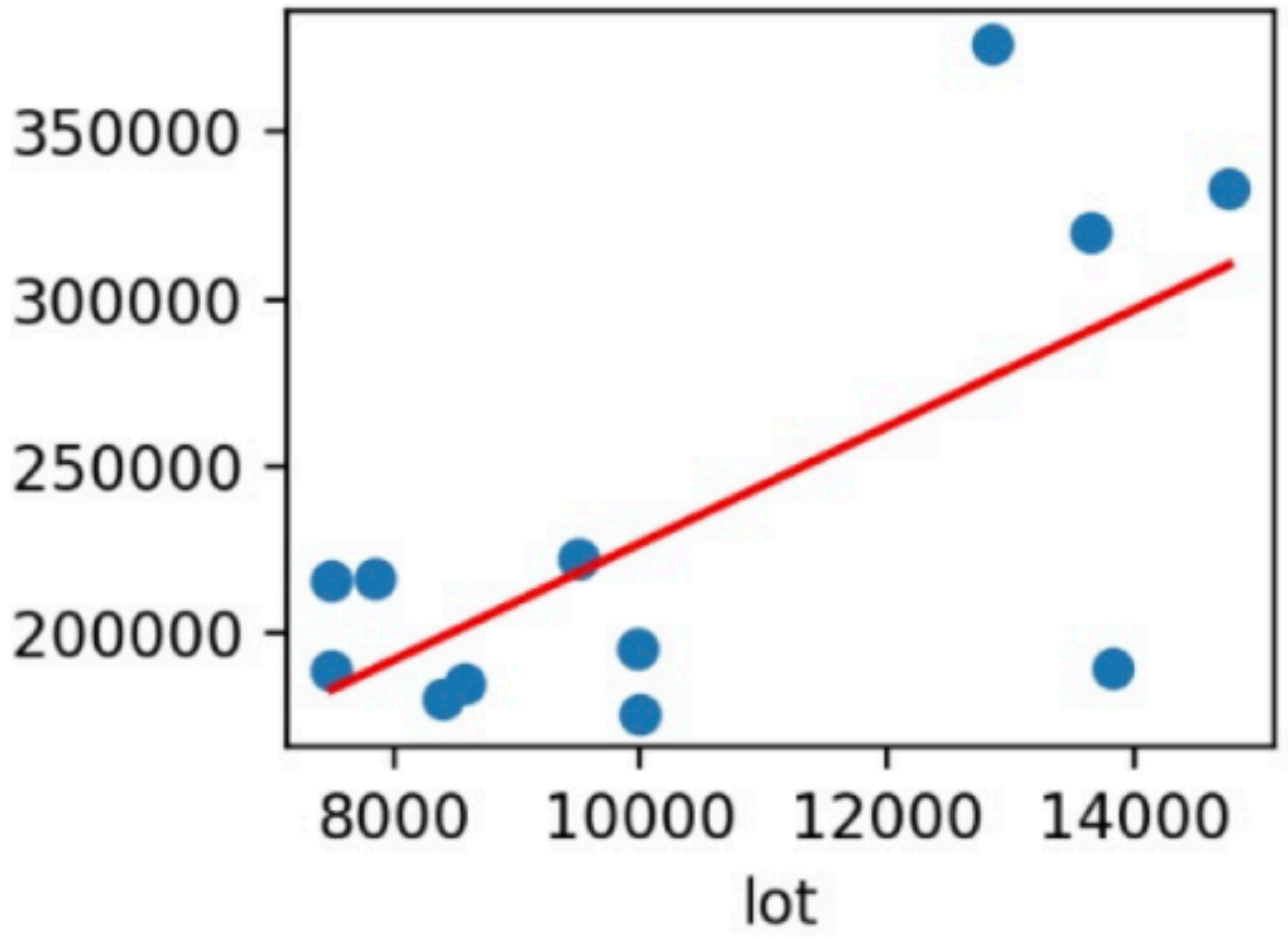
What is a reasonable guess of θ?

Maximize the probability of Y's happening?

Maximizing likelihood estimation -> $\hat{\theta}$

Ground-truth $\theta*$

Another Solution — Gradient Descent



$$h_\theta(x) = \sum_{j=0}^{d} \theta_j x_j = x^T\theta \qquad J(\theta) = \frac{1}{2}\sum_{i=1}^{n}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

Choose

$$\theta = \underset{\theta}{\operatorname{argmin}}\, J(\theta).$$

Gradient Descent

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

Gradient Descent

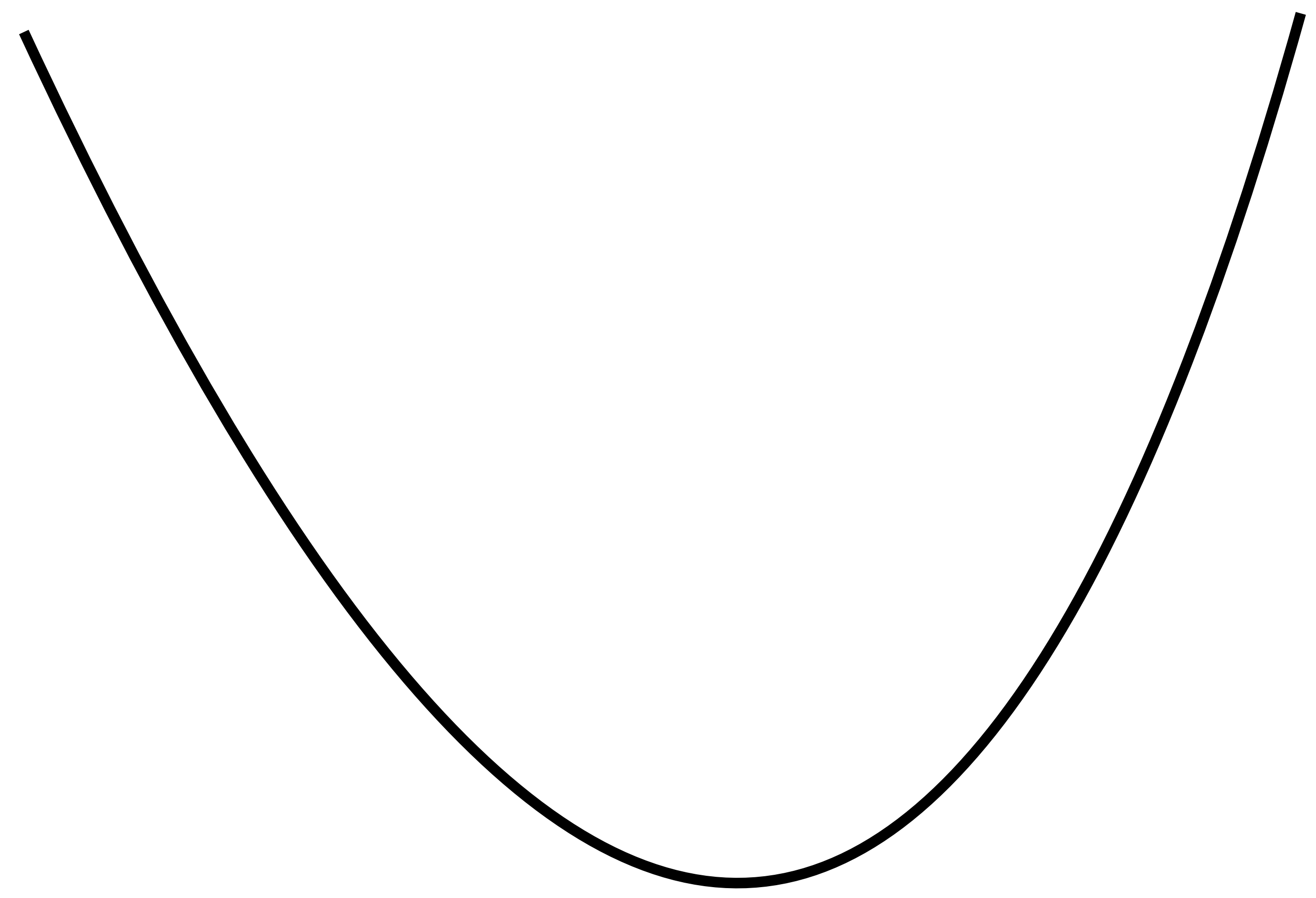
$$J(\theta) = \frac{1}{2}\sum_{i=1}^{n}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

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

Gradient Descent

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

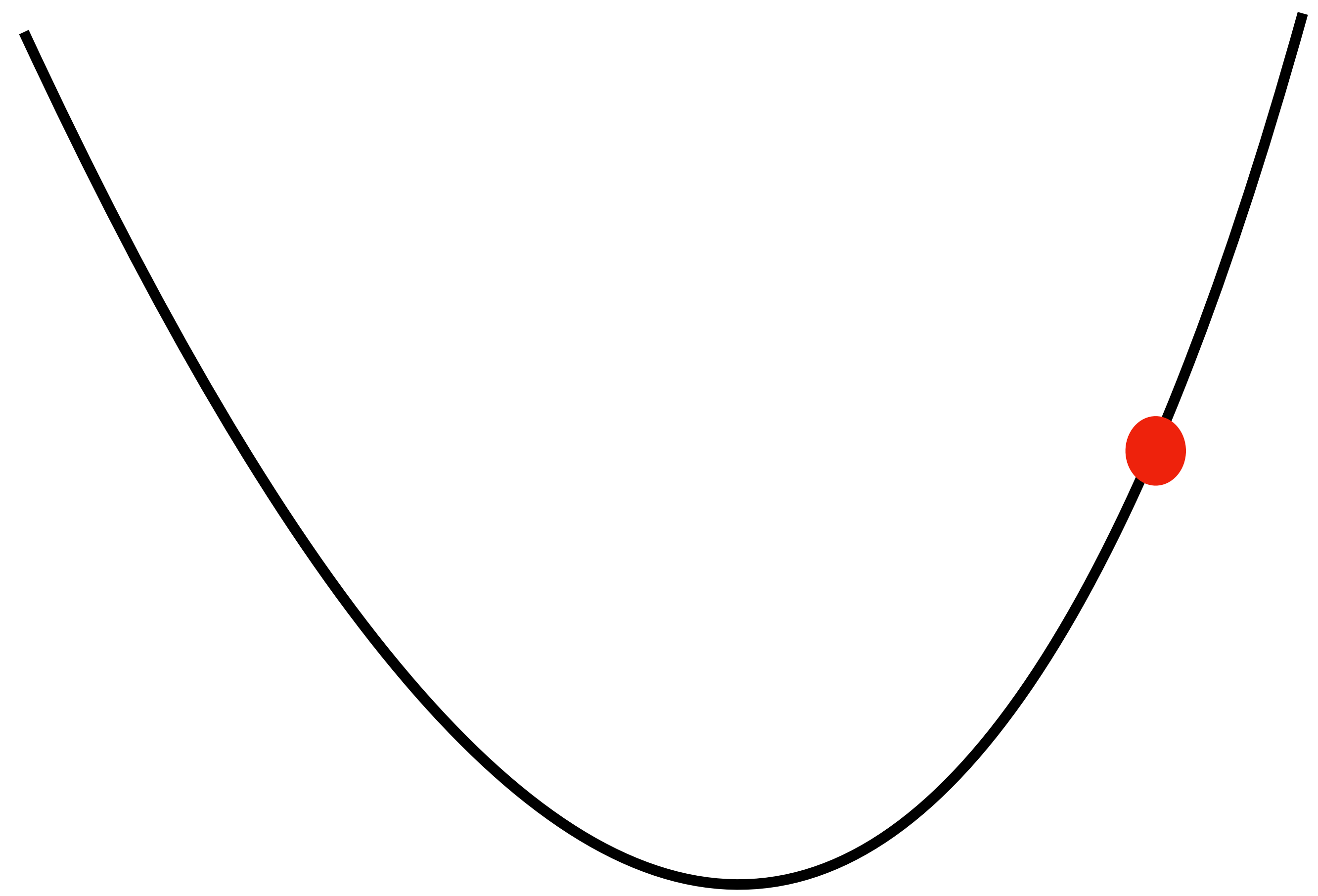
$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$



Gradient Descent

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

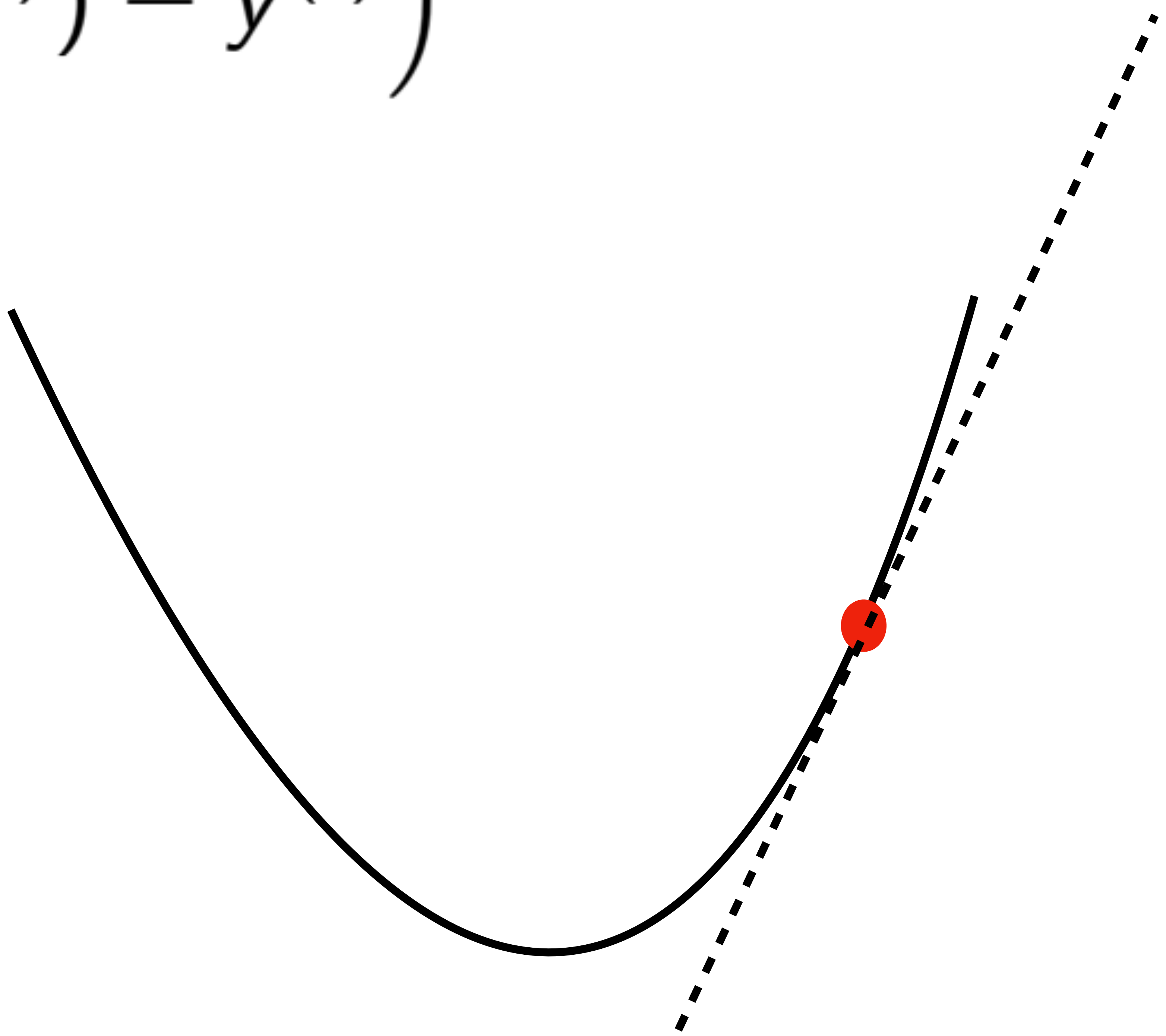
$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$



Gradient Descent

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

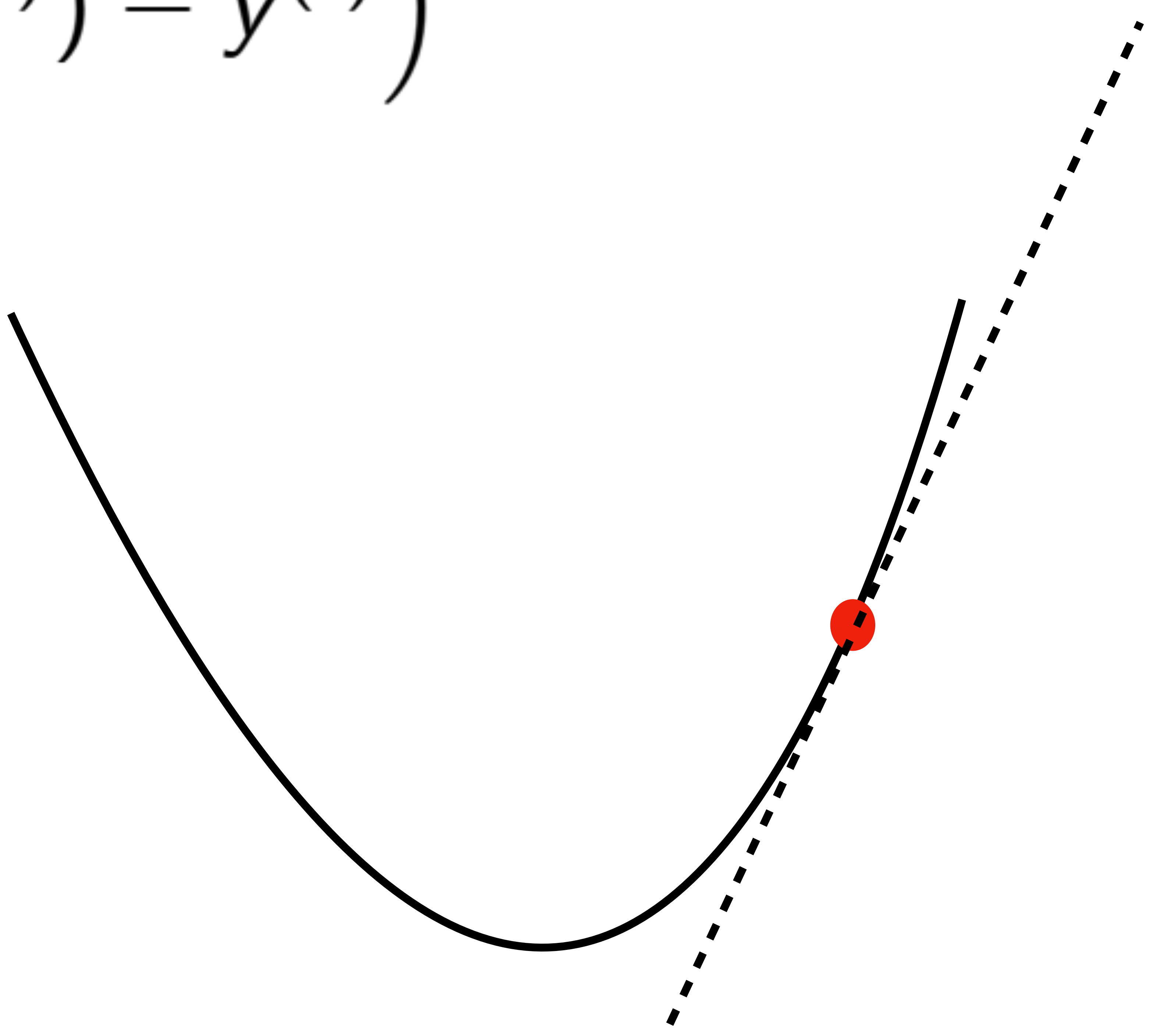
$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$



Gradient Descent

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

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



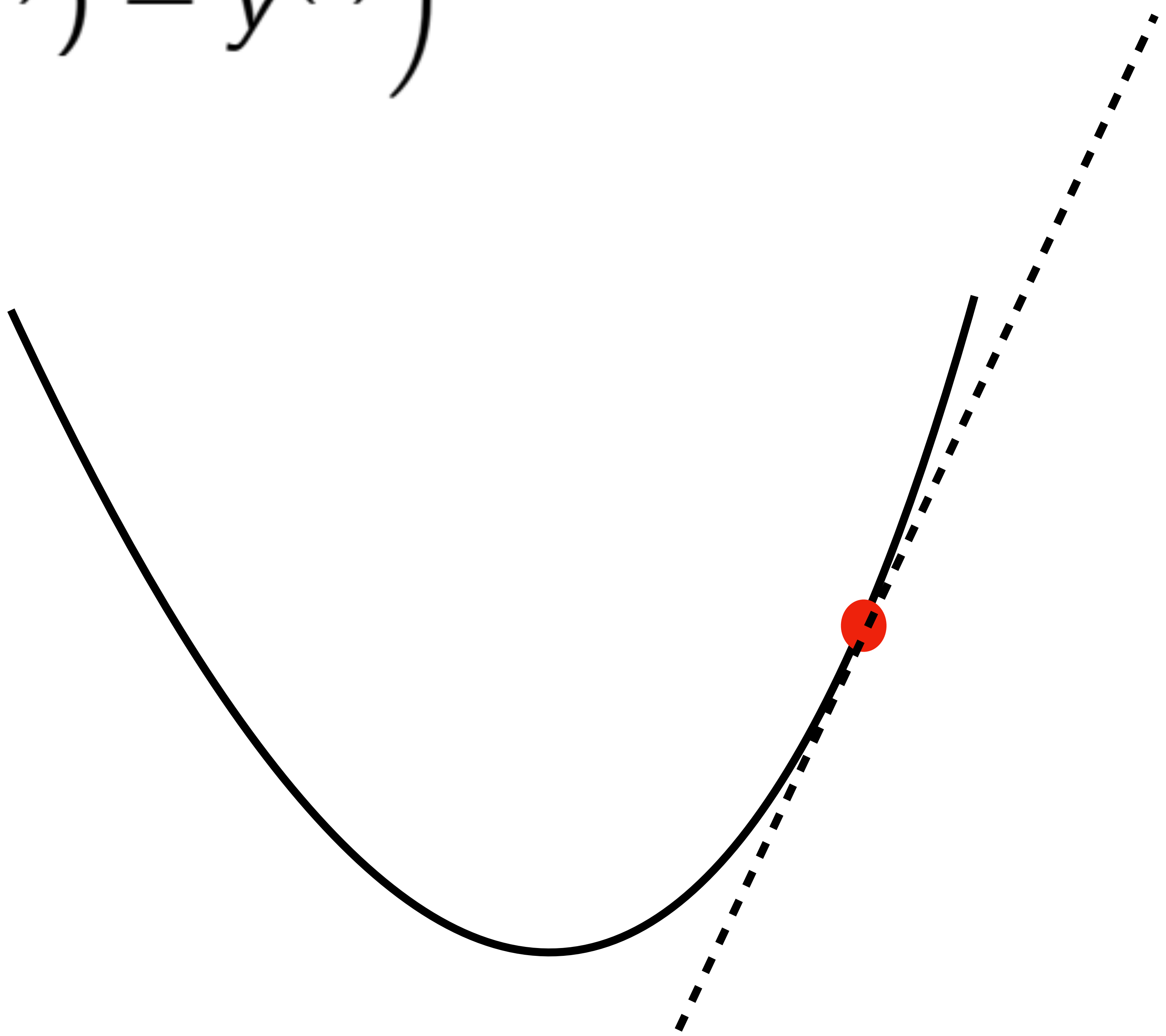
The direction of the steepest descrease of J

Gradient Descent

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

Learning Rate

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



The direction of the steepest descrease of J

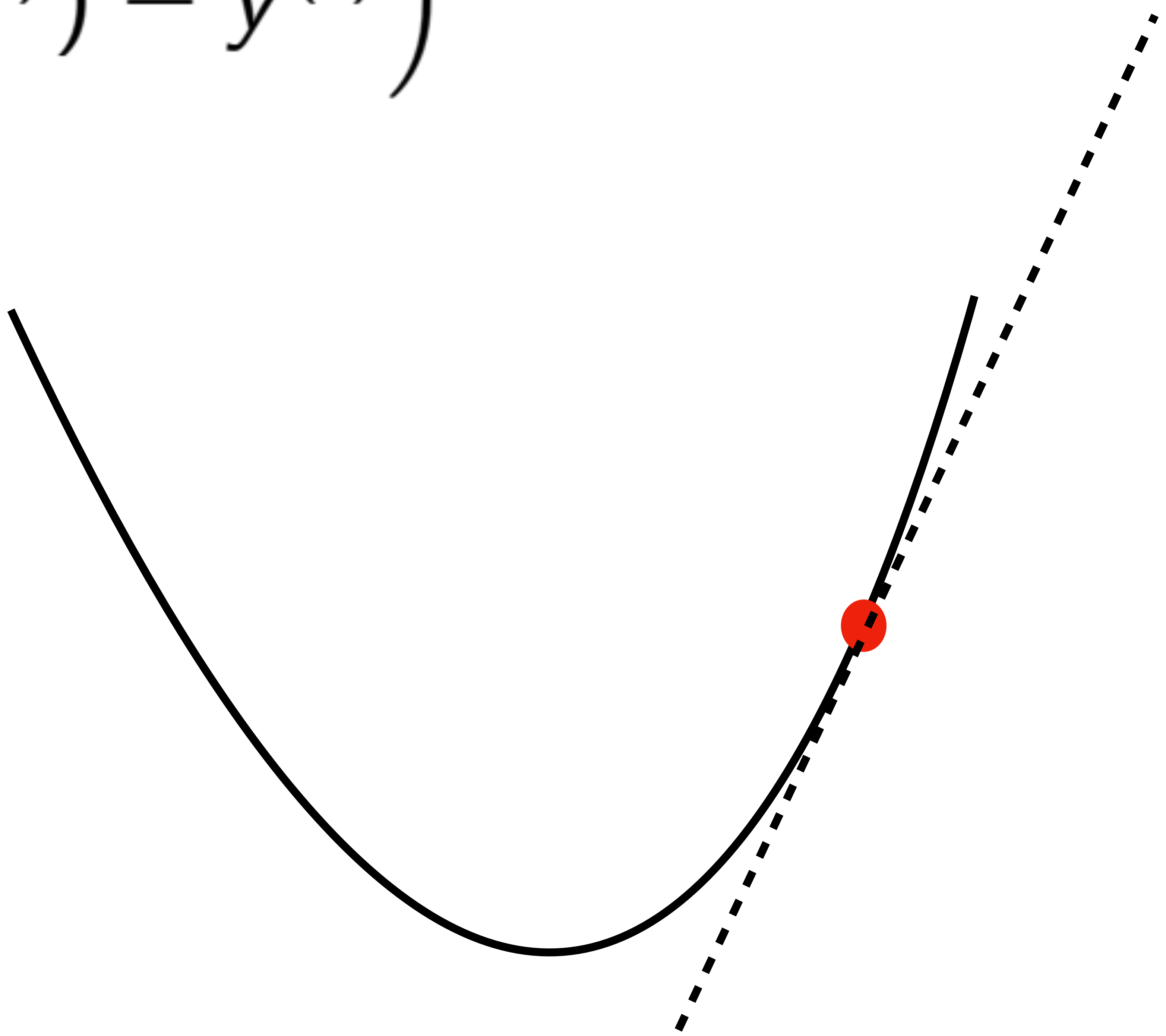
Gradient Descent

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

Learning Rate

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

This update is simultaneously performed for all values of j = 0,...,d.



The direction of the steepest descrease of J

Gradient Descent

For a single training example:

$$
\begin{aligned}
\frac{\partial}{\partial \theta_j} J(\theta) &= \frac{\partial}{\partial \theta_j} \frac{1}{2} \left(h_\theta(x) - y\right)^2 \\
&= 2 \cdot \frac{1}{2} \left(h_\theta(x) - y\right) \cdot \frac{\partial}{\partial \theta_j} \left(h_\theta(x) - y\right) \\
&= \left(h_\theta(x) - y\right) \cdot \frac{\partial}{\partial \theta_j} \left(\sum_{i=0}^{d} \theta_i x_i - y \right) \\
&= \left(h_\theta(x) - y\right) x_j
\end{aligned}
$$

Gradient Descent

For a single training example:

$$
\begin{aligned}
\frac{\partial}{\partial \theta_j} J(\theta) &= \frac{\partial}{\partial \theta_j} \frac{1}{2} \left(h_\theta(x) - y\right)^2 \\
&= 2 \cdot \frac{1}{2} \left(h_\theta(x) - y\right) \cdot \frac{\partial}{\partial \theta_j} \left(h_\theta(x) - y\right) \\
&= \left(h_\theta(x) - y\right) \cdot \frac{\partial}{\partial \theta_j} \left(\sum_{i=0}^{d} \theta_i x_i - y\right) \\
&= \left(h_\theta(x) - y\right) x_j
\end{aligned}
$$

LMS (Least Mean Square) Update Rule

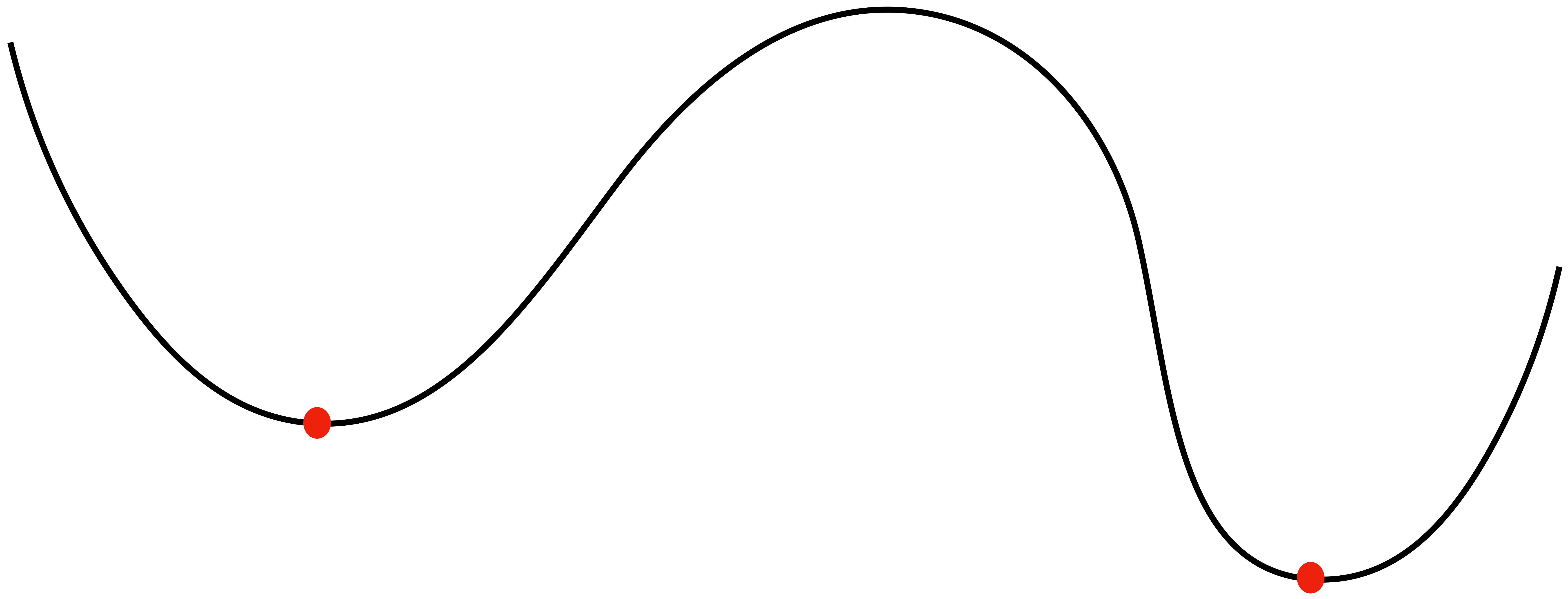
Batch Gradient Descent

For a multiple training examples:

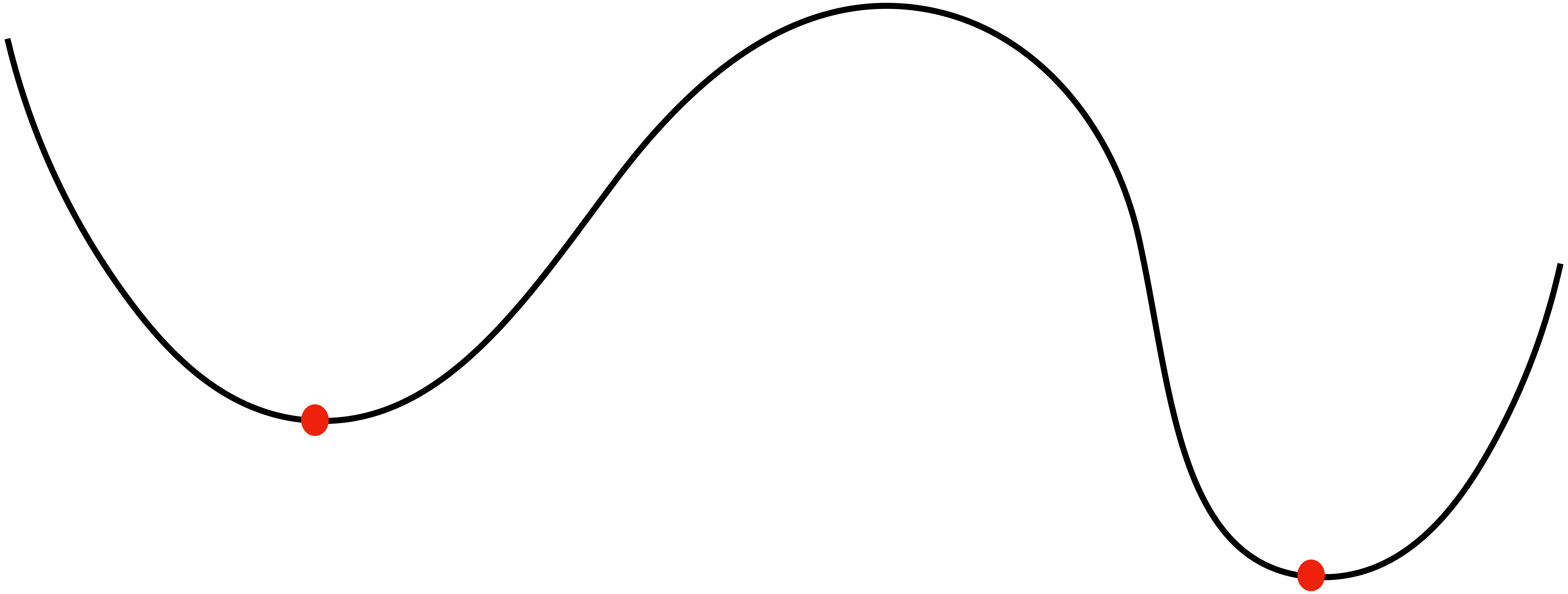
$$\theta_j := \theta_j + \alpha \sum_{i=1}^{n} \left(y^{(i)} - h_\theta(x^{(i)})\right) x_j^{(i)}$$

Repeat until convergence

Local Minimum



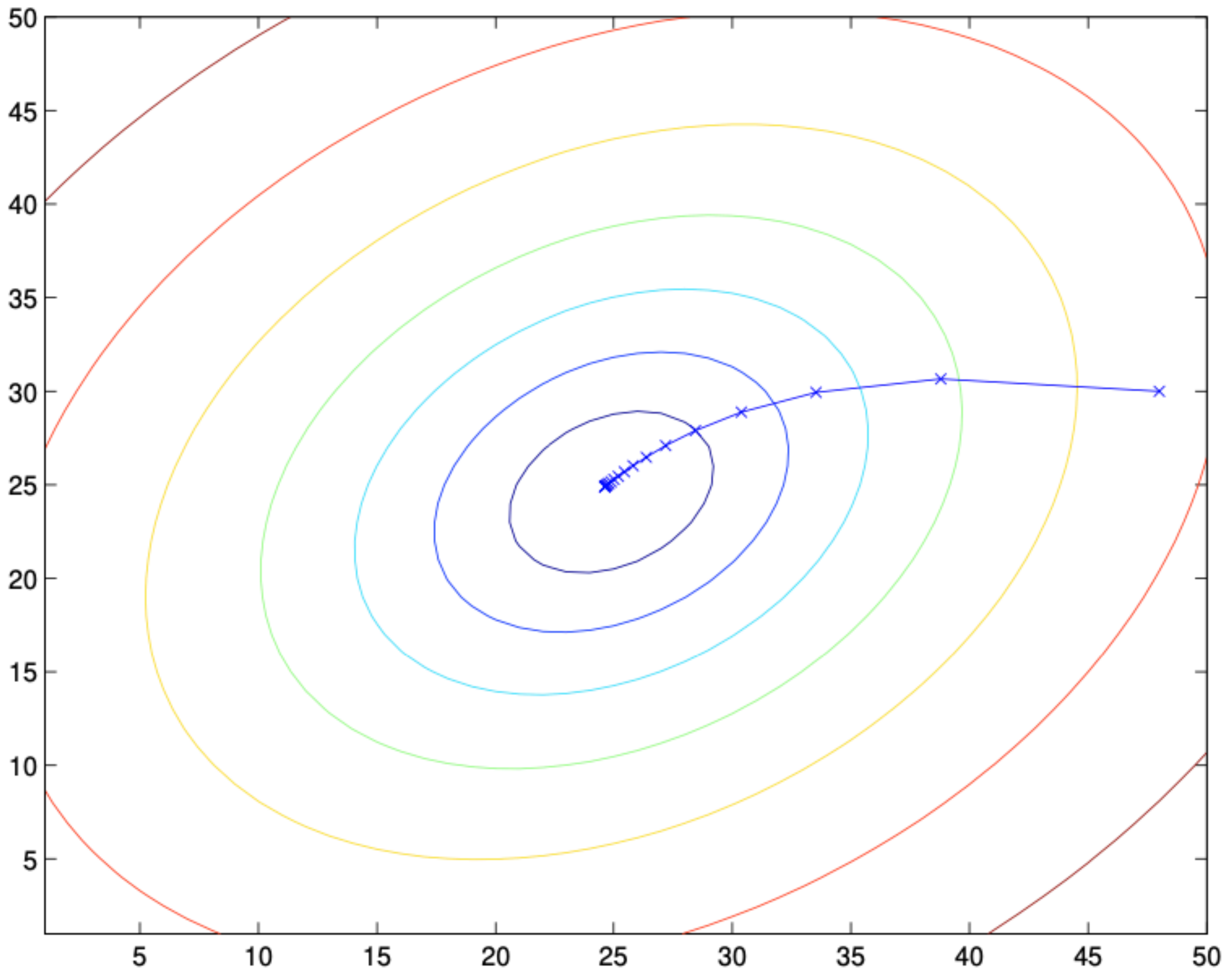
Local Minimum



For least square optimization, are we likely to get local minima rather than the global minima through gradient descent?

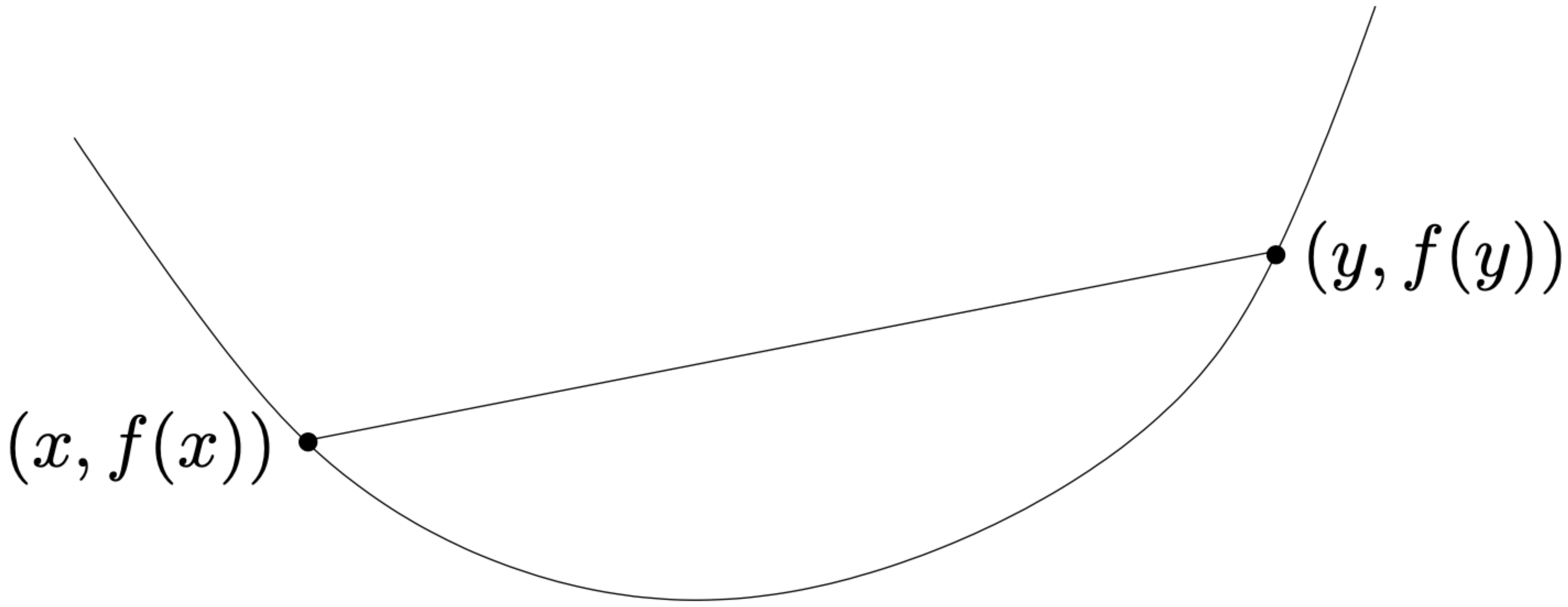
J is a convex quadratic function

There is only one local minima for J



Convex Function

$$f(tx + (1-t)y) \leq tf(x) + (1-t)f(y) \quad \text{for } 0 \leq t \leq 1$$



Thank You!
Q & A