

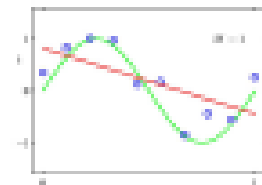
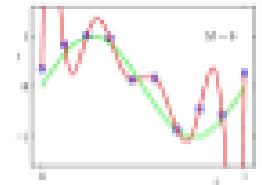


On Linear Regression

Alexander G. Ororbia II
Introduction to Machine Learning
CSCI-335
9/28/2026

Goal: Minimize Expected Loss

- We have decomposed expected loss into sum of (squared) bias, a variance and a constant noise term
- There is a trade-off between bias and variance
 - Very flexible models have low bias and high variance
 - Rigid models have high bias and low variance
 - Optimal model has the best balance



Machine Learning Algorithms

	Supervised Learning	Unsupervised Learning
Discrete	Classification	Clustering
Continuous	Regression	Dimensionality reduction

Machine Learning Algorithms

	Supervised Learning	Unsupervised Learning
Discrete	Classification	Clustering
Continuous	Regression	Dimensionality reduction

Recap: Nearest Neighbor Classifier

- **Training data**

$$(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(N)}, y^{(N)})$$

- **Learning**

Do nothing.

- **Testing**

$$h(x) = y^{(k)}, \text{ where } k = \operatorname{argmin}_i D(x, x^{(i)})$$



Recap: Instance/Memory-based Learning

1. A distance metric

- Continuous? Discrete? PDF? Gene data? Learn the metric?

2. How many nearby neighbors to look at?

- 1? 3? 5? 15?

3. A weighting function (optional)

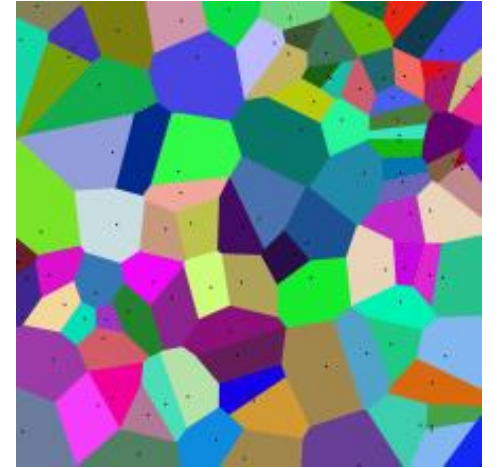
- Closer neighbors matter more

4. How to fit with the local points?

- Kernel regression (not discussed, but good to be aware of)

Things to Remember

- Supervised Learning
 - Training/testing data; classification/regression; hypothesis
- K-NN
 - Simplest learning algorithm
 - With sufficient data, very hard to beat “strawman” approach
- Problems with K-NN
 - Curse of dimensionality
 - Not robust to irrelevant features
 - Slow NN search: must remember (very large) dataset for prediction



Linear Regression

- **Model representation**
- Cost function
- Gradient descent
- Multivariate regression

Regression

real-valued output

Training set

Learning Algorithm

x

h

y

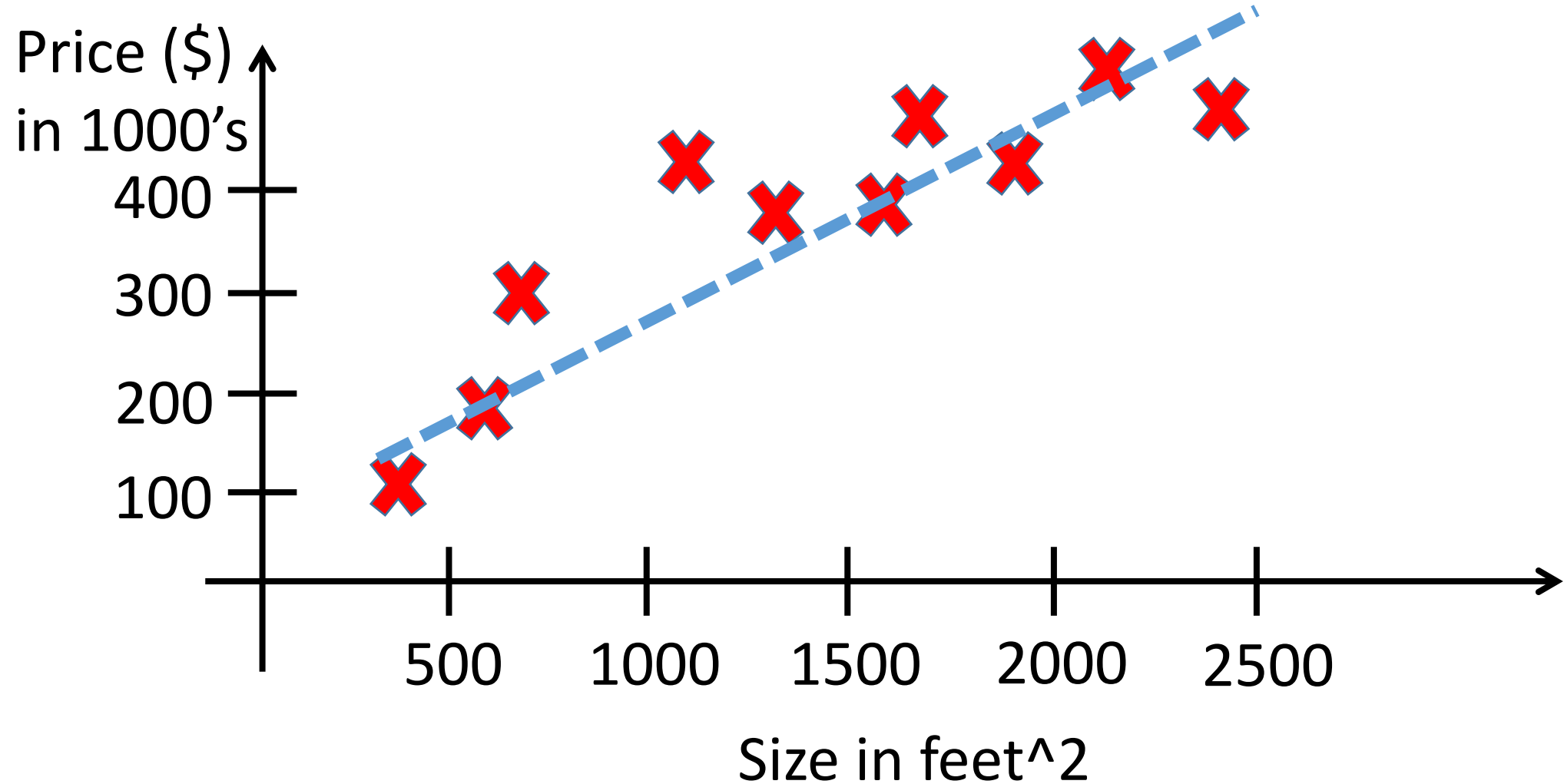
Size of house

Hypothesis

Estimated price

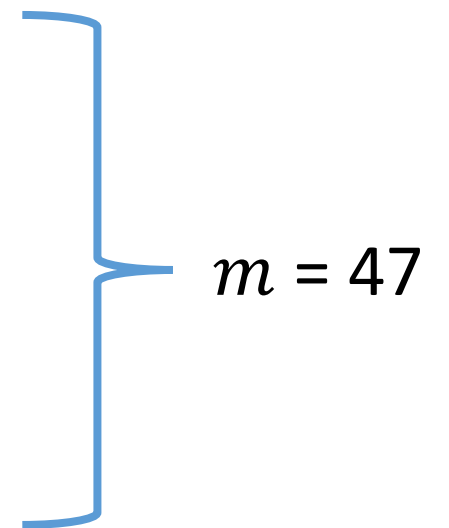


House Pricing Prediction



Training Set

Size in feet ² (x)	Price (\$) in 1000's (y)
2104	460
1416	232
1534	315
852	178
...	...



$m = 47$

- Notation:

- m = Number of training examples
- x = Input variable / features
- y = Output variable / target variable
- (x, y) = One training example
- $(x^{(i)}, y^{(i)}) = i^{th}$ training example

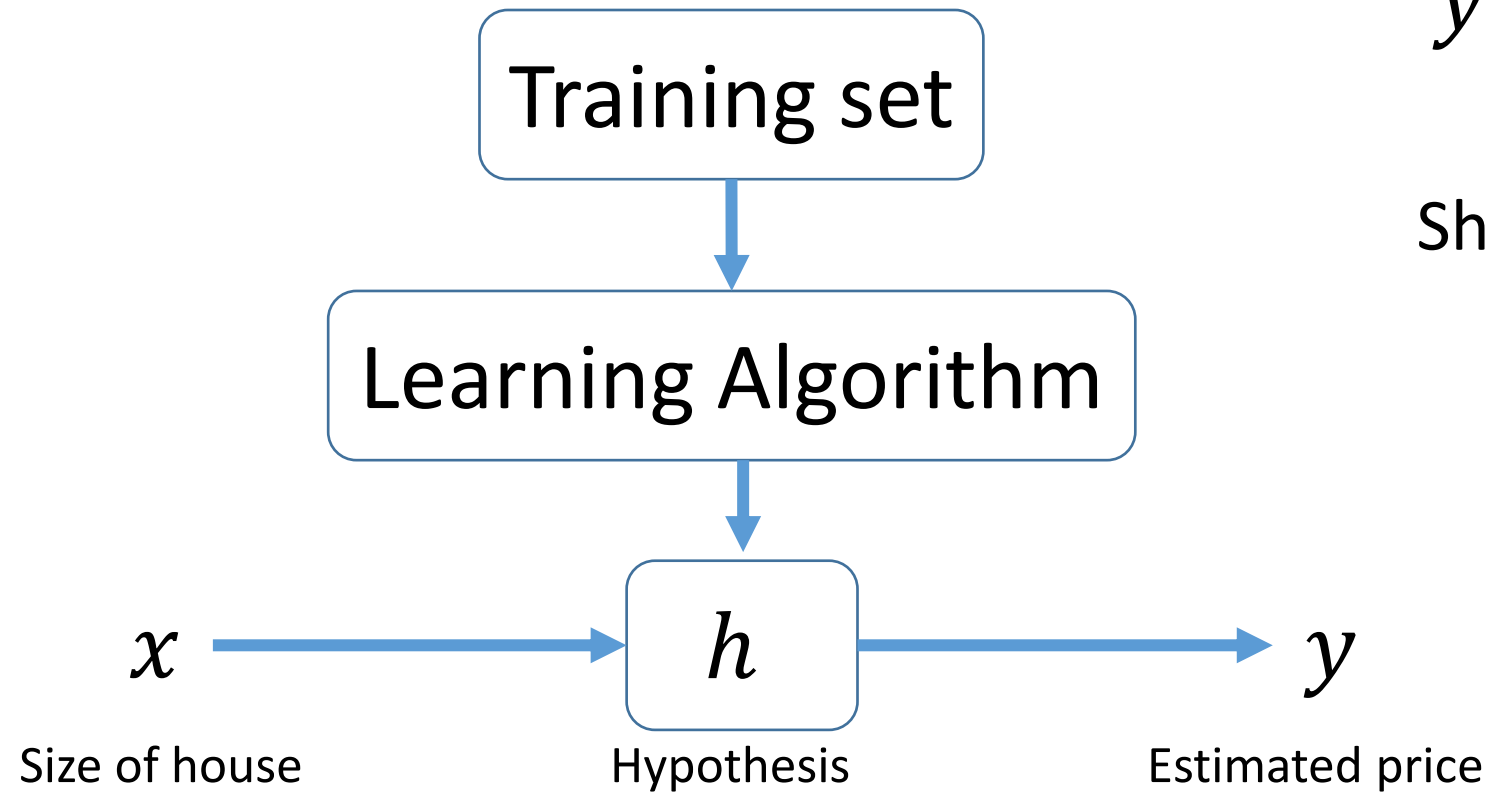
Examples:

$$x^{(1)} = 2104$$

$$x^{(2)} = 1416$$

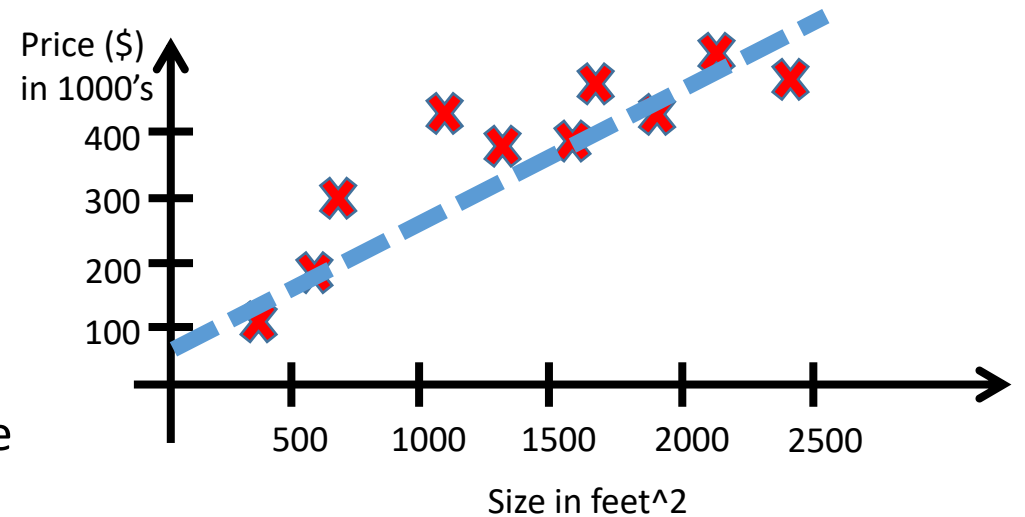
$$y^{(1)} = 460$$

Model Representation



$$y = h_{\theta}(x) = \theta_0 + \theta_1 x$$

Shorthand $h(x)$



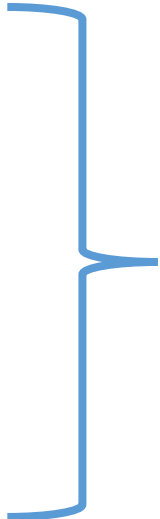
Univariate linear regression

Linear Regression

- Model representation
- **Cost function**
- Gradient descent
- Multivariate regression

Training Set

Size in feet ² (x)	Price (\$) in 1000's (y)
2104	460
1416	232
1534	315
852	178
...	...



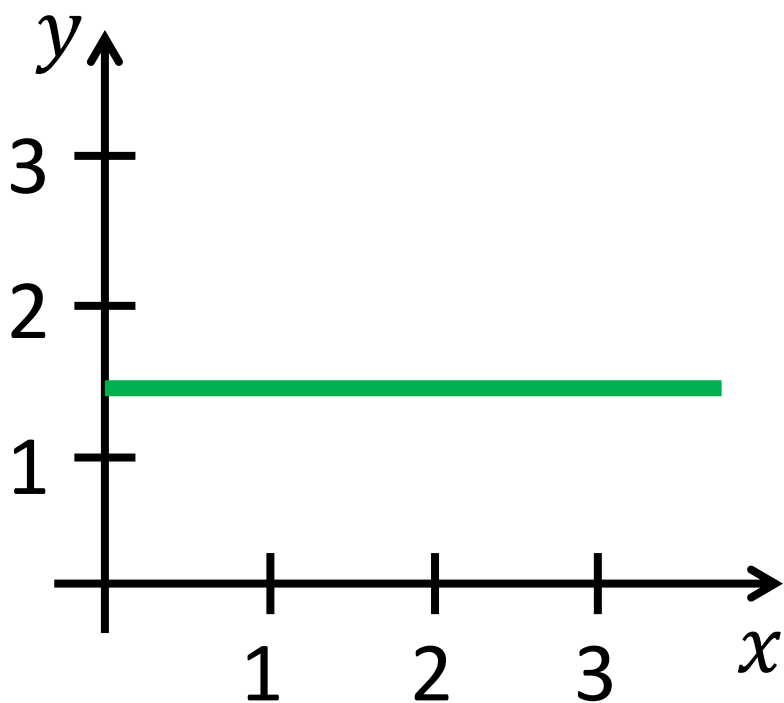
$m = 47$

• Hypothesis $h_{\theta}(x) = \theta_0 + \theta_1 x$

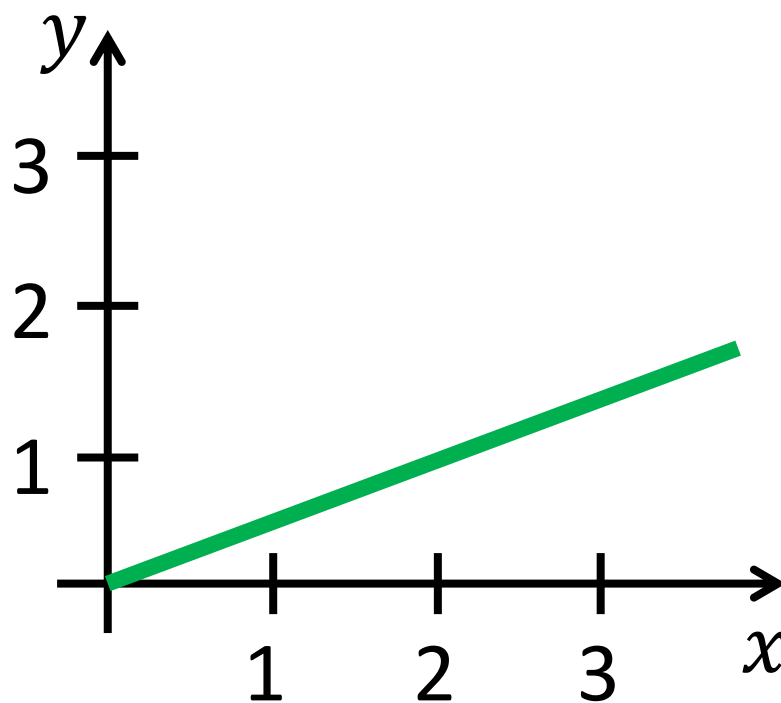
θ_0, θ_1 : parameters/weights

How to choose θ_i 's?

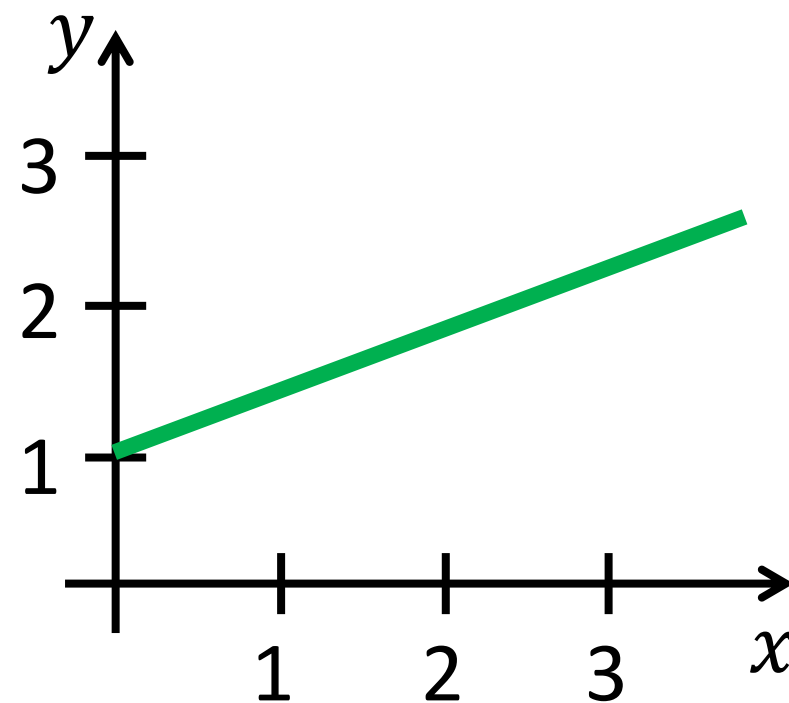
$$h_{\theta}(x) = \theta_0 + \theta_1 x$$



$$\begin{aligned}\theta_0 &= 1.5 \\ \theta_1 &= 0\end{aligned}$$



$$\begin{aligned}\theta_0 &= 0 \\ \theta_1 &= 0.5\end{aligned}$$

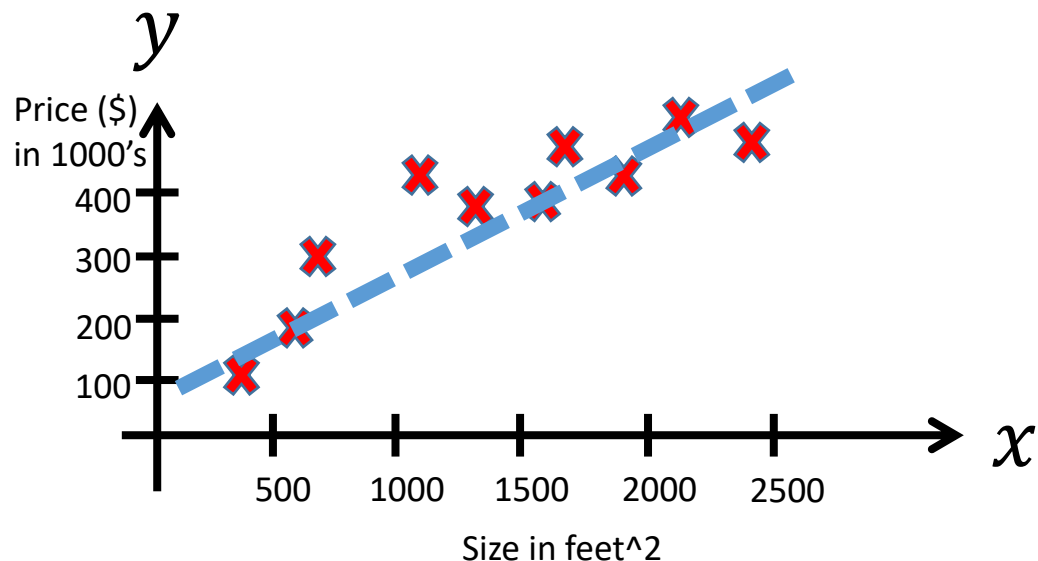


$$\begin{aligned}\theta_0 &= 1 \\ \theta_1 &= 0.5\end{aligned}$$

Cost Function

- Idea:

Choose θ_0, θ_1 so that $h_{\theta}(x)$ is close to y for our training example (x, y)



$$\underset{\theta_0, \theta_1}{\text{minimize}} \quad \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

$$h_{\theta}(x^{(i)}) = \theta_0 + \theta_1 x^{(i)}$$

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

$$\underset{\theta_0, \theta_1}{\text{minimize}} \quad \boxed{J(\theta_0, \theta_1)} \quad \text{Cost function}$$

Simplified

- **Hypothesis:**

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



- **Hypothesis:**

$$h_{\theta}(x) = \theta_1 x$$

$$\theta_0 = 0$$

- **Parameters:**

$$\theta_0, \theta_1$$



- **Parameters:**

$$\theta_1$$

- **Cost function:**

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



- **Cost function:**

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

- **Goal:**

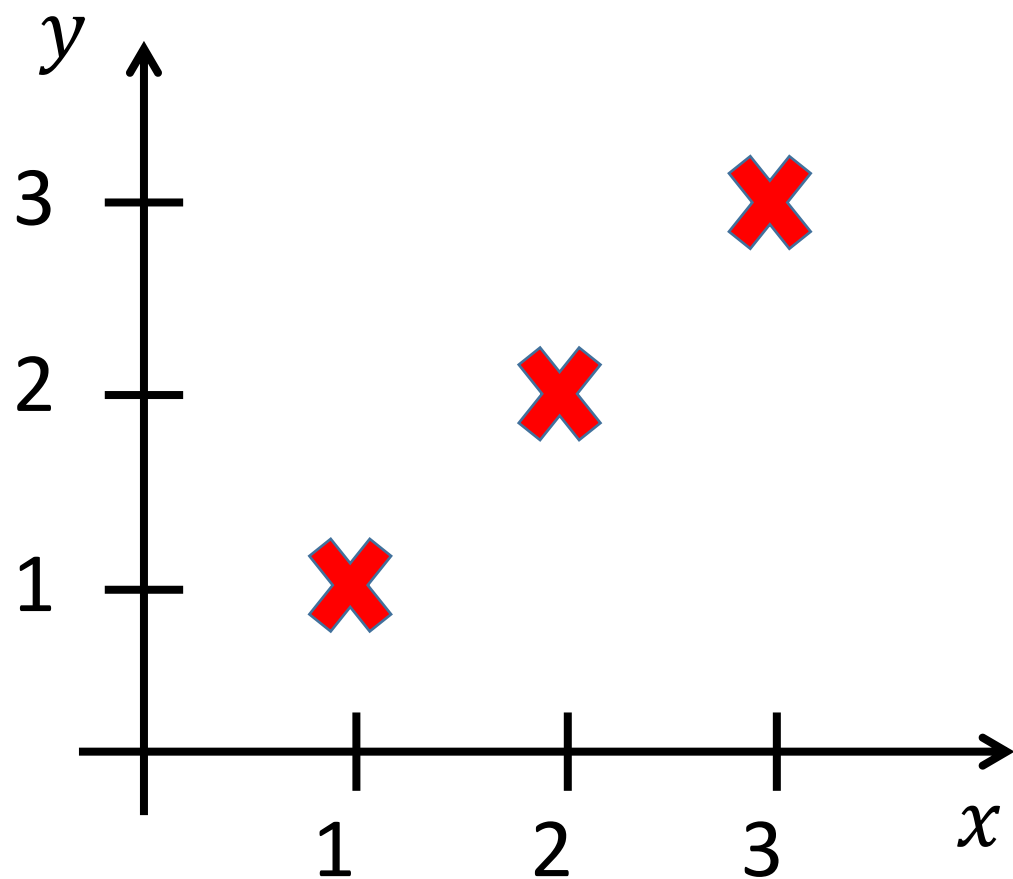
$$\underset{\theta_0, \theta_1}{\text{minimize}} J(\theta_0, \theta_1)$$



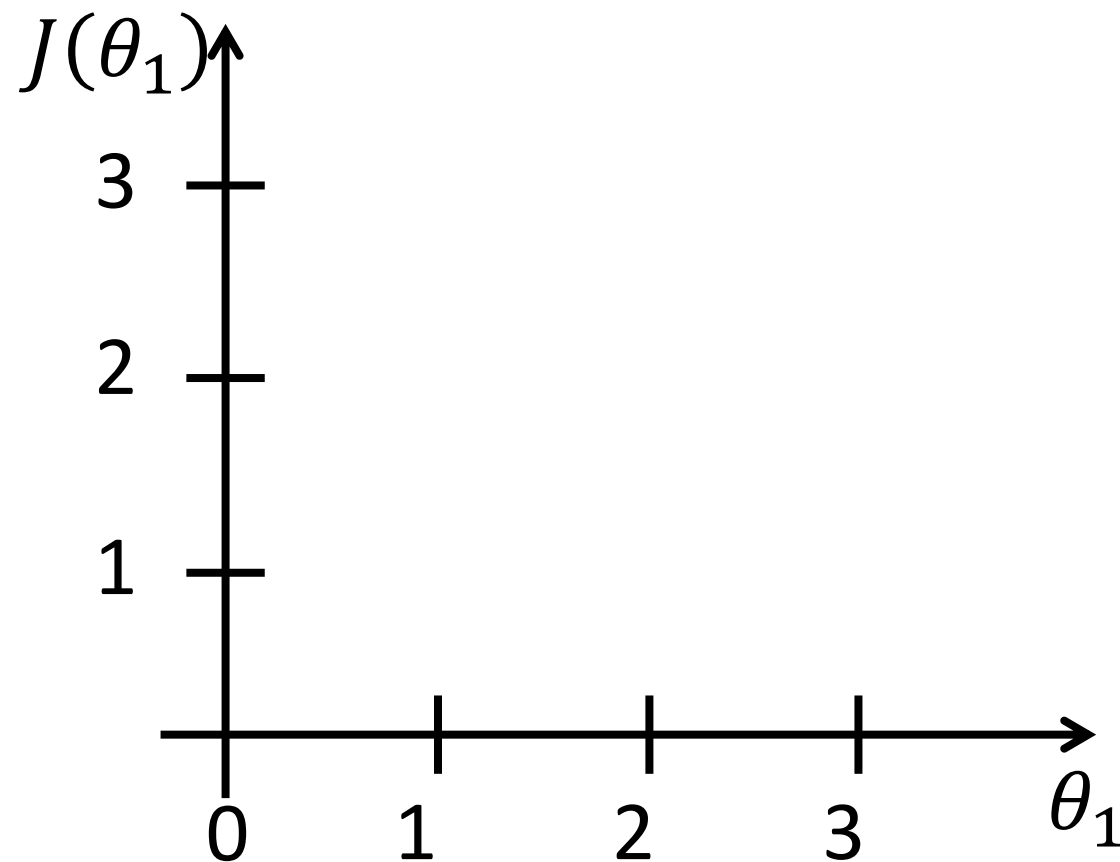
- **Goal:**

$$\underset{\theta_0, \theta_1}{\text{minimize}} J(\theta_1)$$

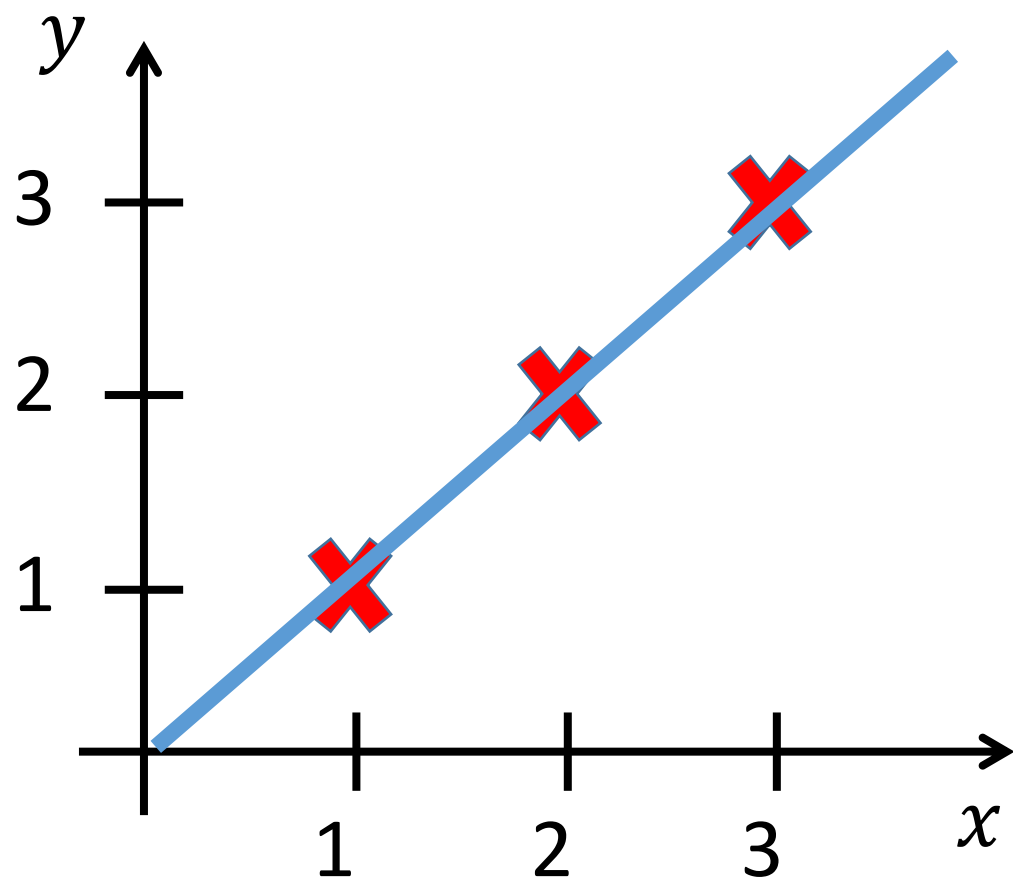
$h_{\theta}(x)$, function of x



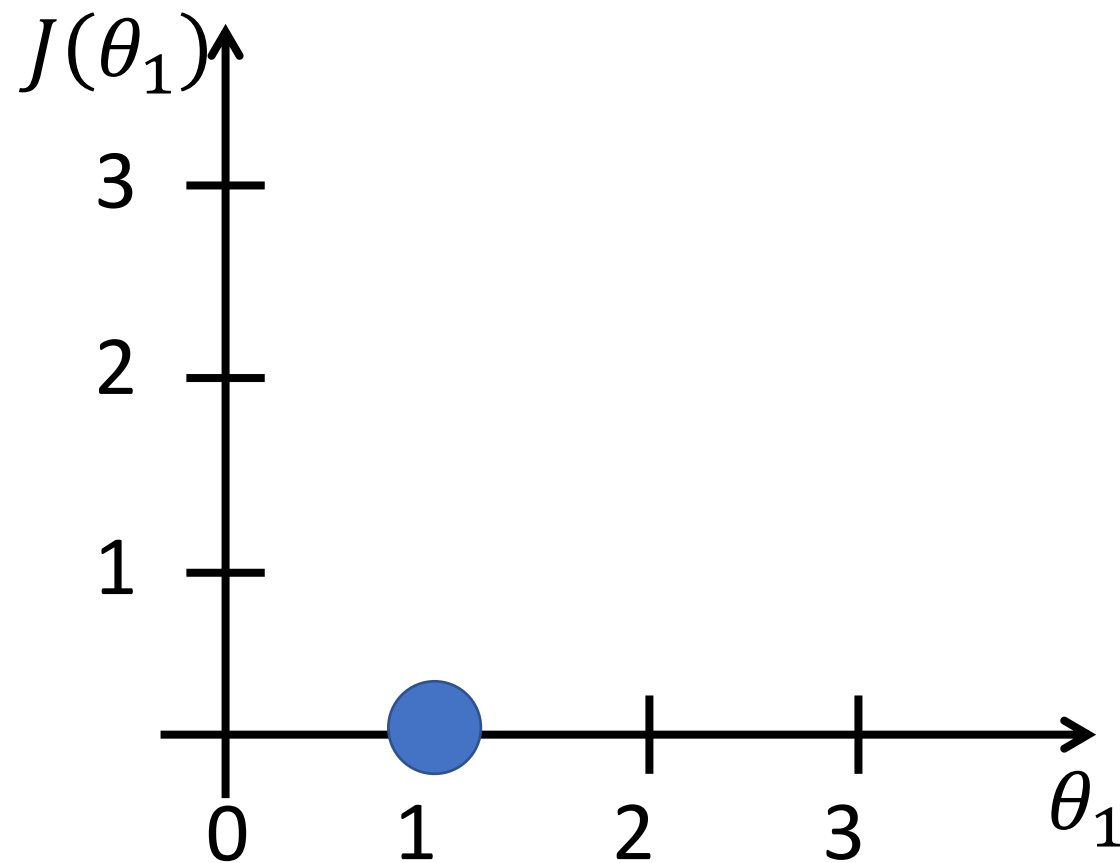
$J(\theta_1)$, function of θ_1



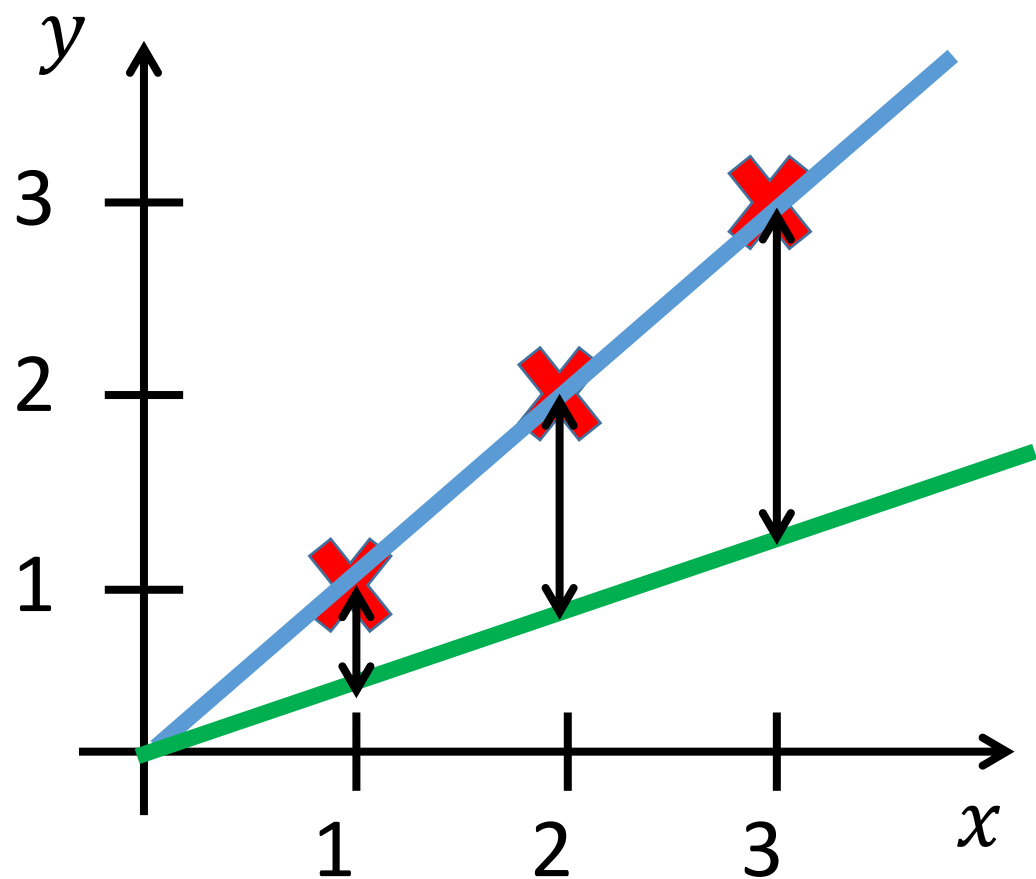
$h_{\theta}(x)$, function of x



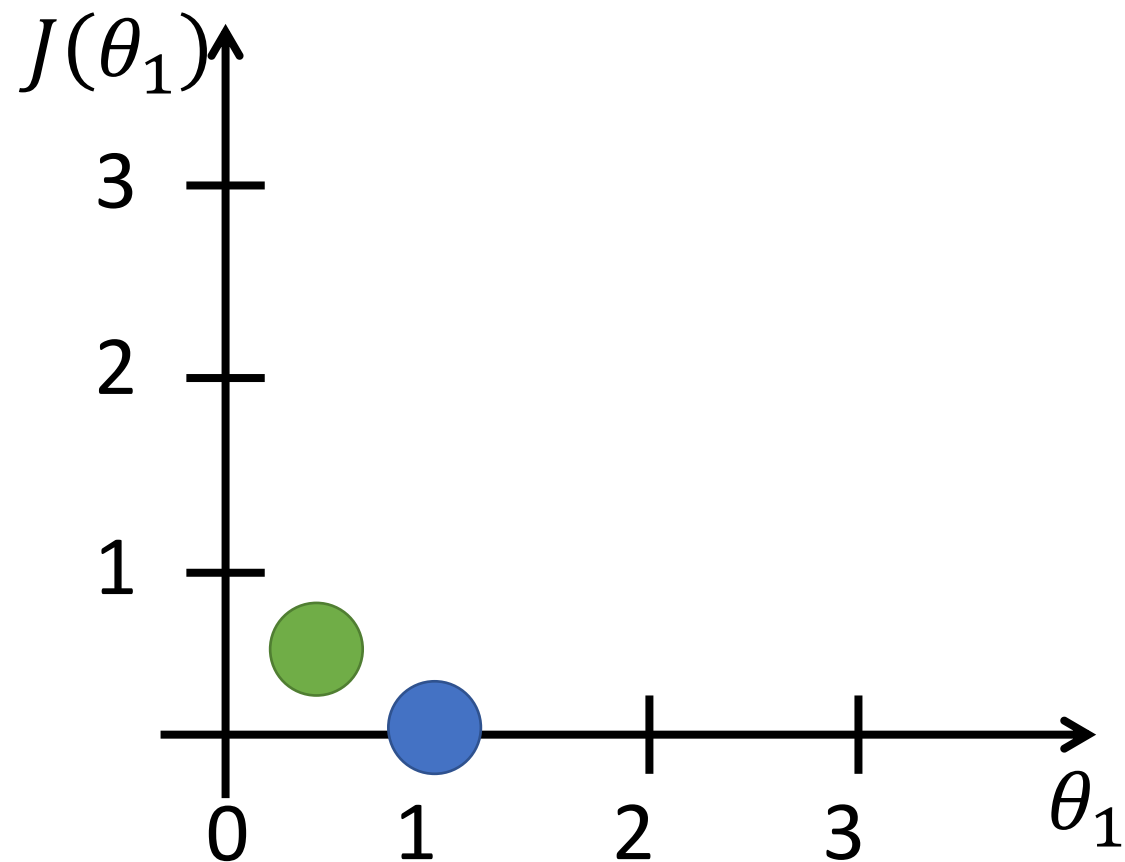
$J(\theta_1)$, function of θ_1



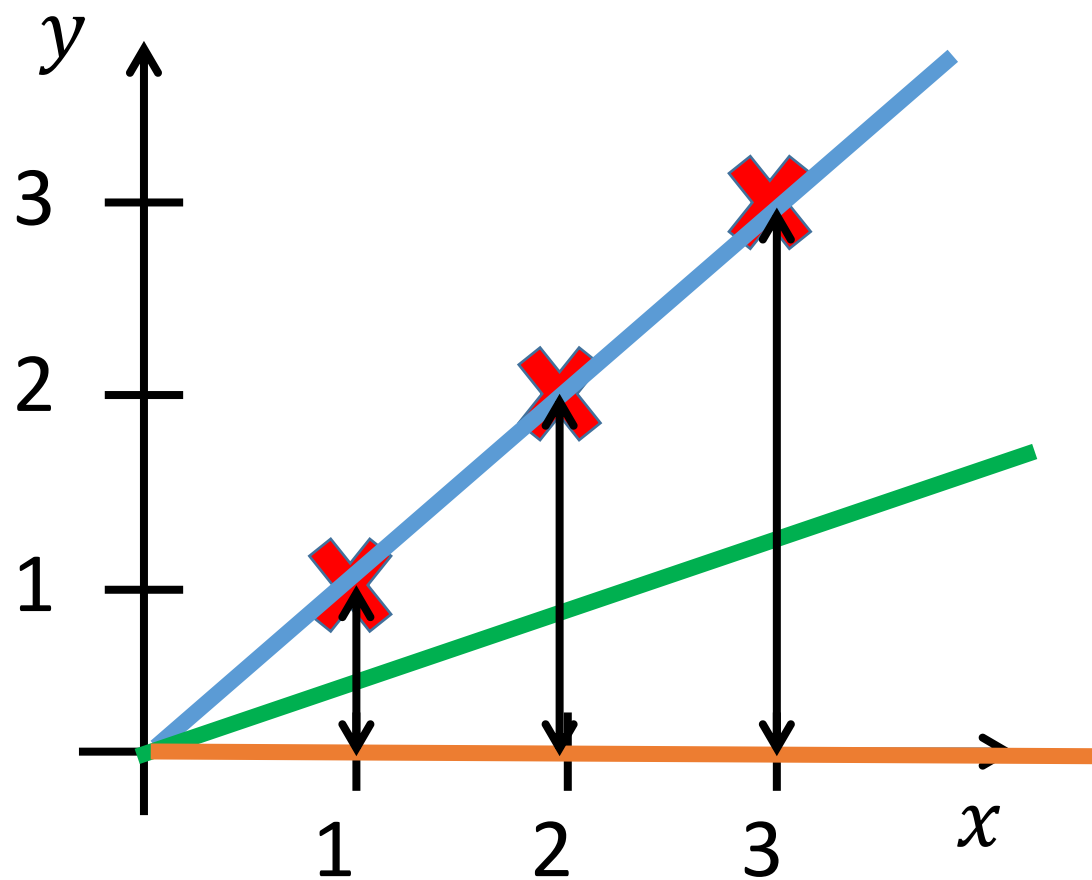
$h_{\theta}(x)$, function of x



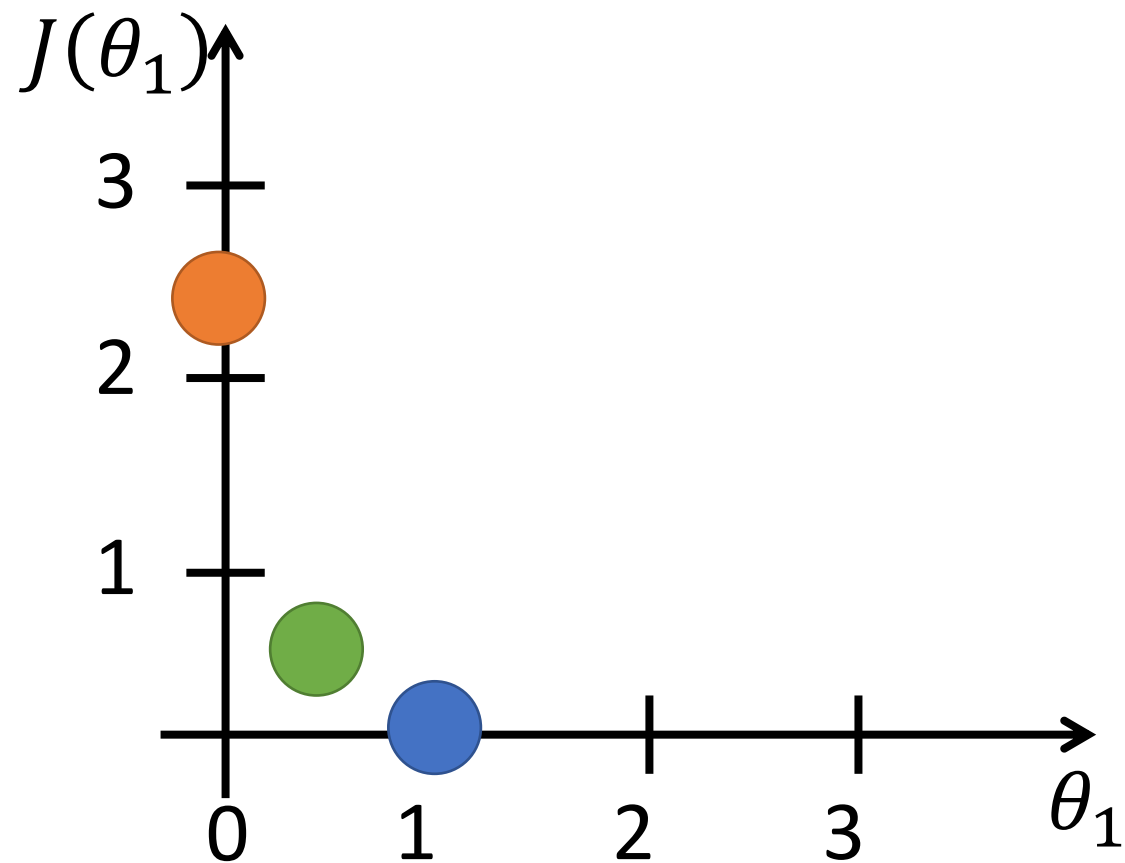
$J(\theta_1)$, function of θ_1



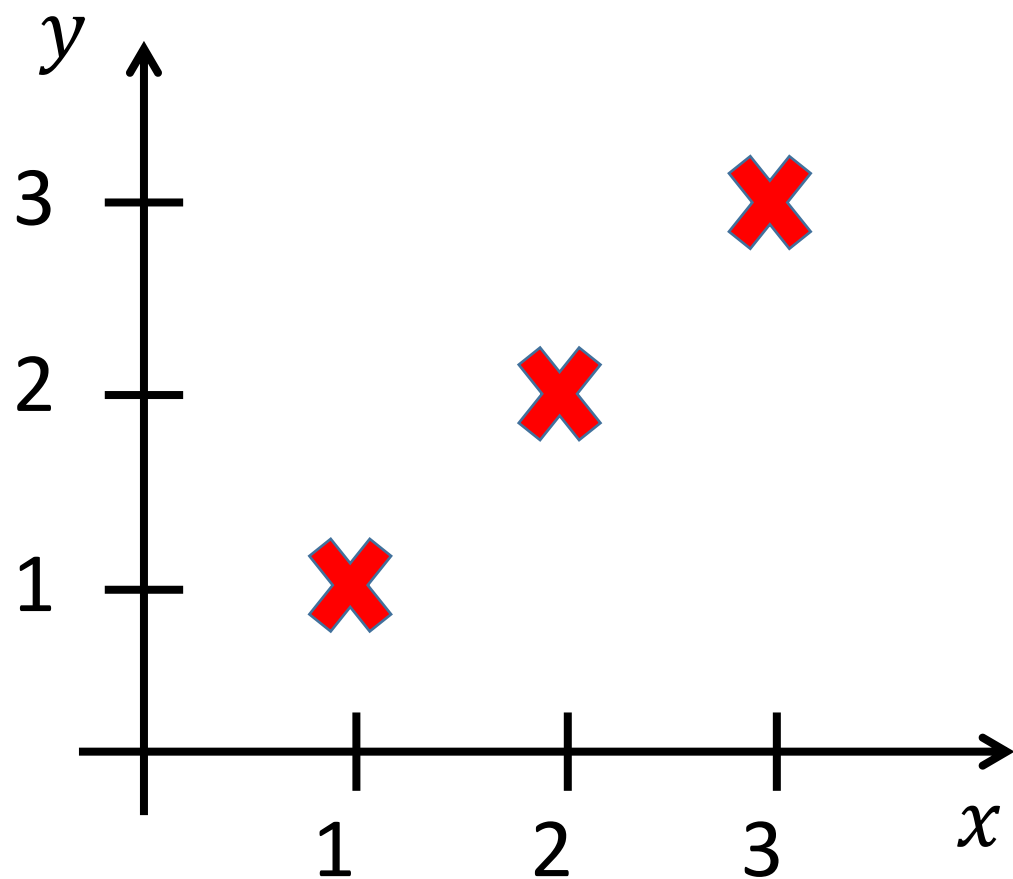
$h_{\theta}(x)$, function of x



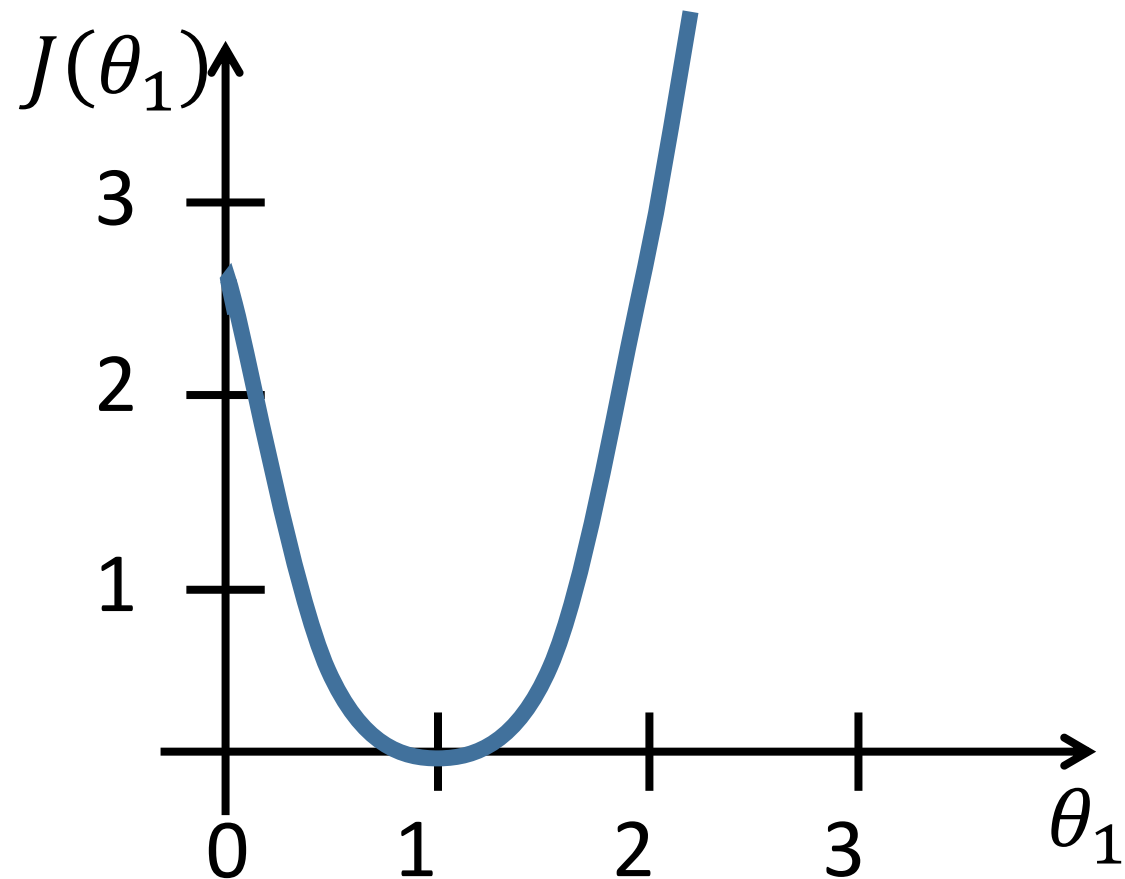
$J(\theta_1)$, function of θ_1



$h_{\theta}(x)$, function of x

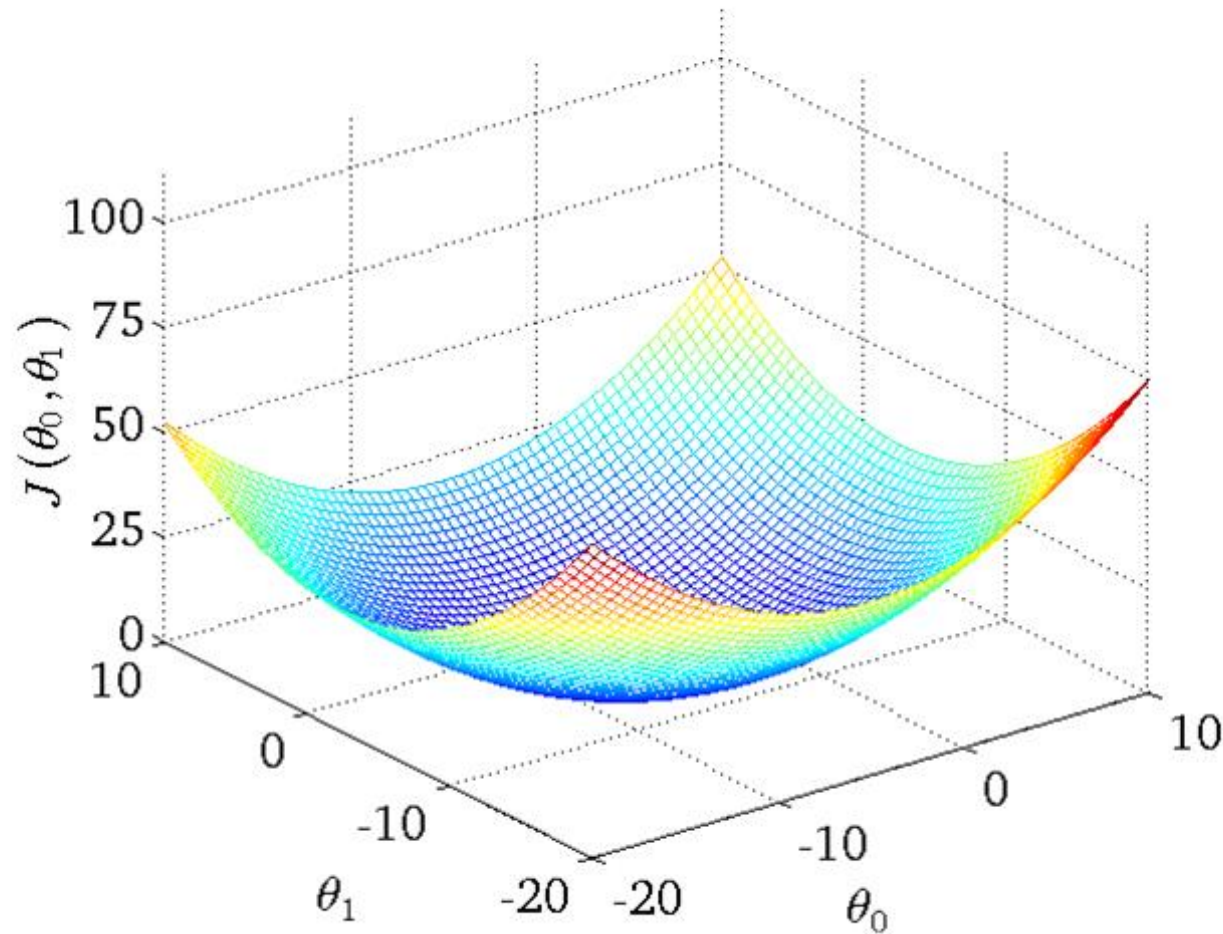


$J(\theta_1)$, function of θ_1



- **Hypothesis:** $h_{\theta}(x) = \theta_0 + \theta_1 x$
- **Parameters:** θ_0, θ_1
- **Cost function:** $J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$
- **Goal:** minimize $J(\theta_0, \theta_1)$
 θ_0, θ_1

Cost Function Surface



Questions?

Deep robots!

Deep questions?!

