Machine Learning Crash Course

Week 1

“Field of study that gives computers the ability to learn without being explicitly programmed.” ~Arthur Samuel

"A computer program is said to learn from experience E with respect to some class of tasks T and performance P, if its performance in tasks T, as measured by P, improves with experience E." ~Tom Michel

The attempt to minimize the error and maximize accuracy at the cost of explainability of a given model.

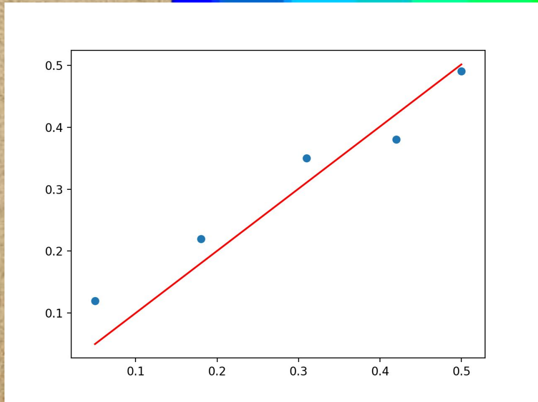
Supervised learning
Unsupervised learning
Reinforcement learning
Recommender systems

Supervised learning

- Regression Problem
- Classification Problem

unSupervised learning

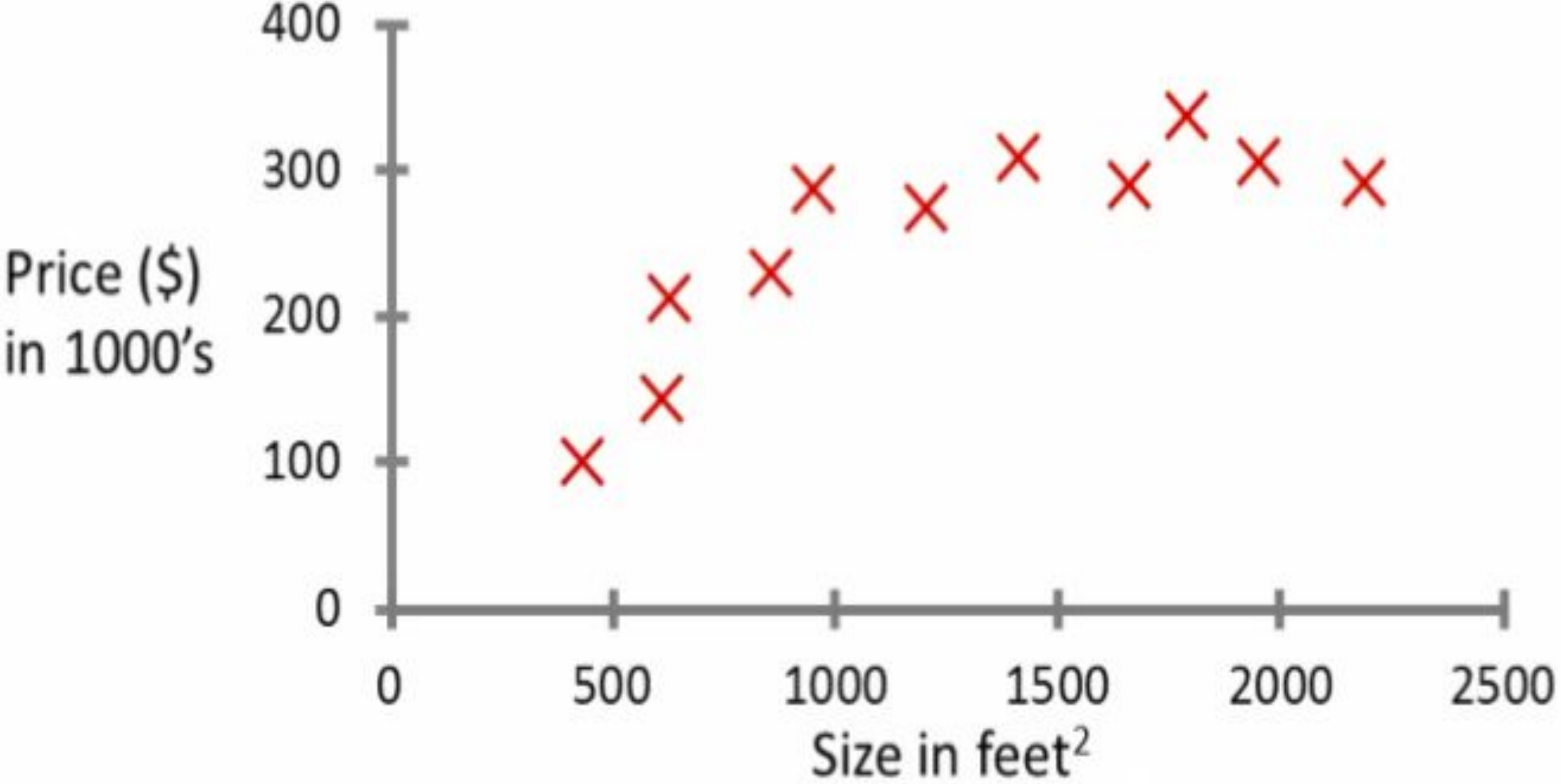
- Clustering Problem
- Cocktail Party Problem



Topics (W1)

- Linear Regression (Univariate)
  - ※ Model Representation
  - ※ Cost function
  - ※ Gradient descent
- Linear Algebra
  - ※ Matrices and Vectors
  - ※ Matrix & Vector Addition and Multiplication
  - ※ Matrix Inverse and Transpose

Housing price prediction.



Size in feet² (x)	Price ($) in 1000's (y)
2104	460
1416	232
1534	315
852	178
...	...

$m = 47$

- Number of training examples (m)
- Inpur features (x)
- Output features (y)
- Ith index to the training data

Goal

Model Representation (univariate)

- Define a hypothesis (h)
- Whereas :
  - Theta (sub-0) is the zero condition
  - Theta (sub-1) is the gradient
- x is some arbitrary variable
- The function uses parameters learned by the system to give a prediction

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

Initial Condition

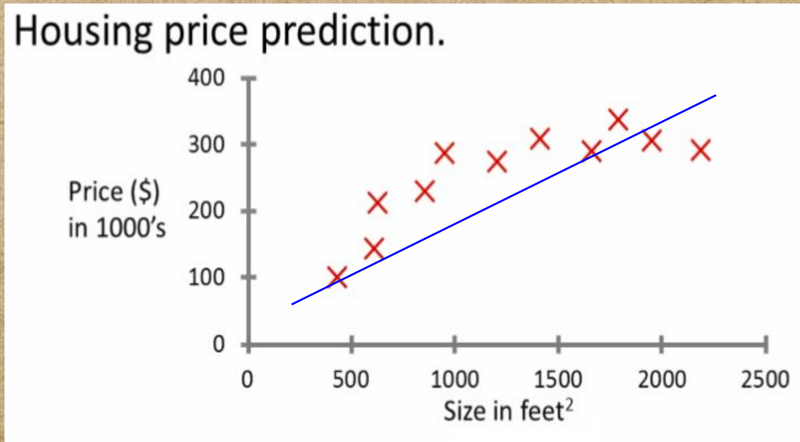
Gradient

Refined Objective

- Using different values for Theta, define a cost function to best fit the line to the data
- Generate parameters such that:
    - The hypothesis (h) is very close to the actual (y) value
- Minimize the squared difference between h(x) and y for every sample

More Formally

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



Cost function

- Minimize the cost function for all the training data

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

EXAMPLE 1

- Minimize the cost function
  - Assume theta (sub-0) = 0

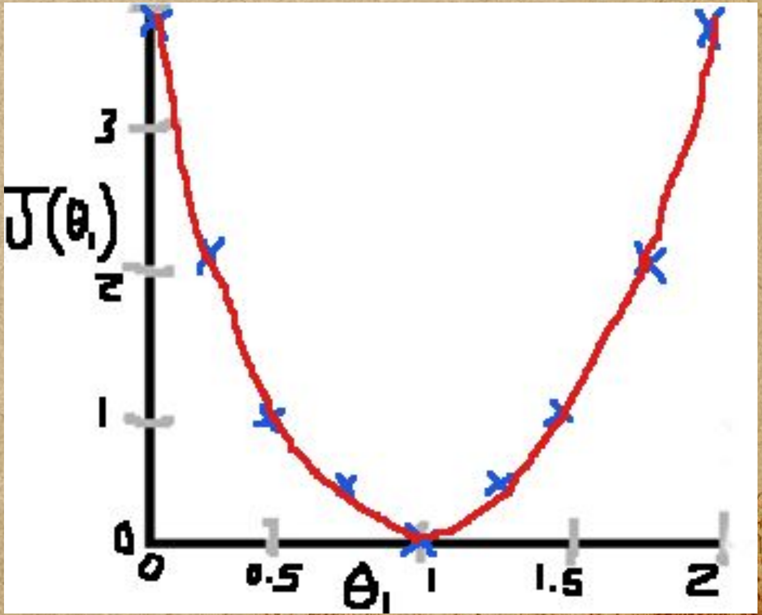
$$h_\theta(x) = \theta_0 + \theta_1 x$$

$$\theta_0 = 0$$

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

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

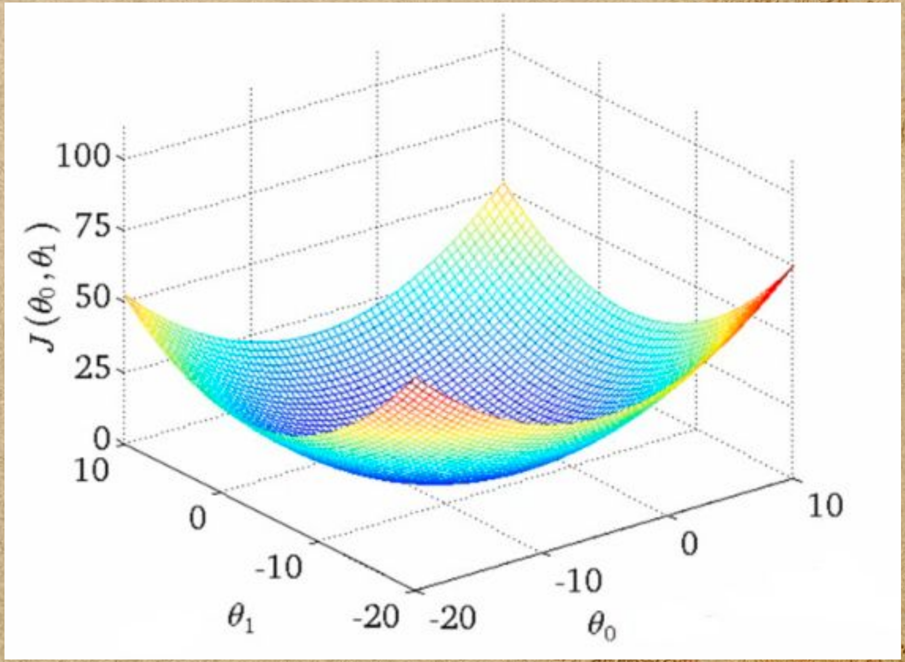
θ_1 is the gradient



EXAMPLE 2

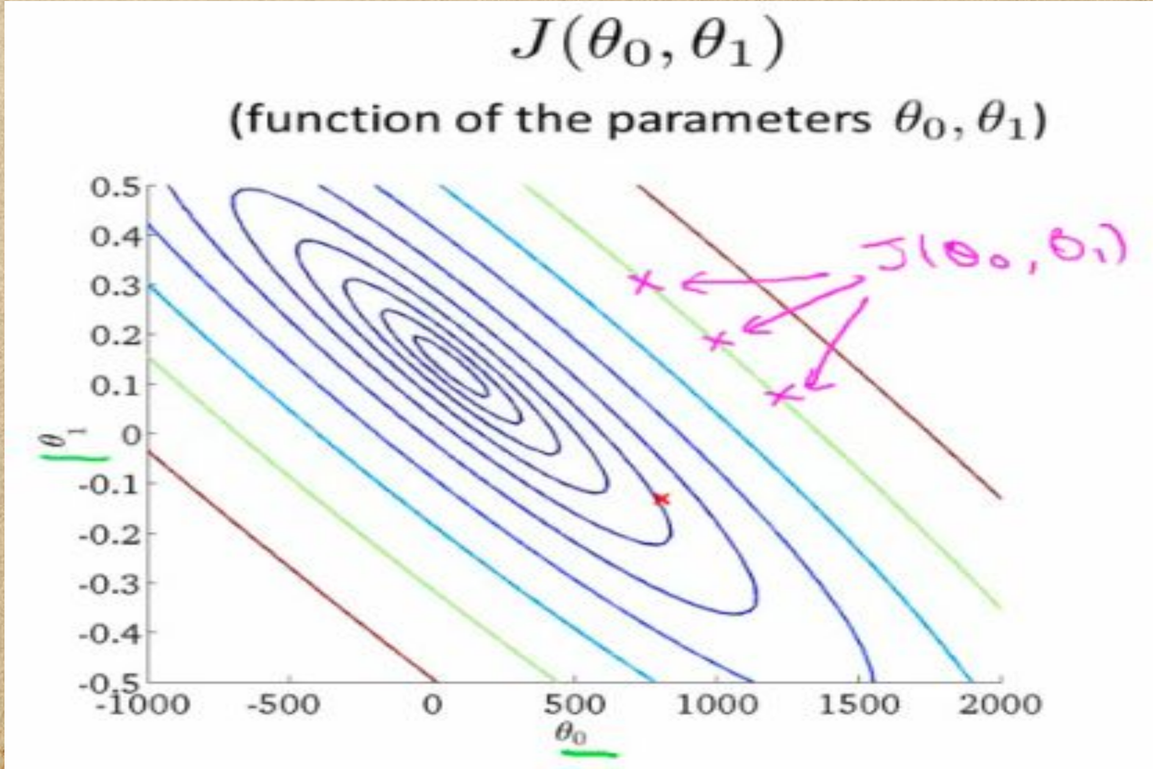
- Minimize the cost function
    - Assume theta (sub-0) = 0

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



EXAMPLE 2 (continued)

- Use a contour plot to visualize the cost function



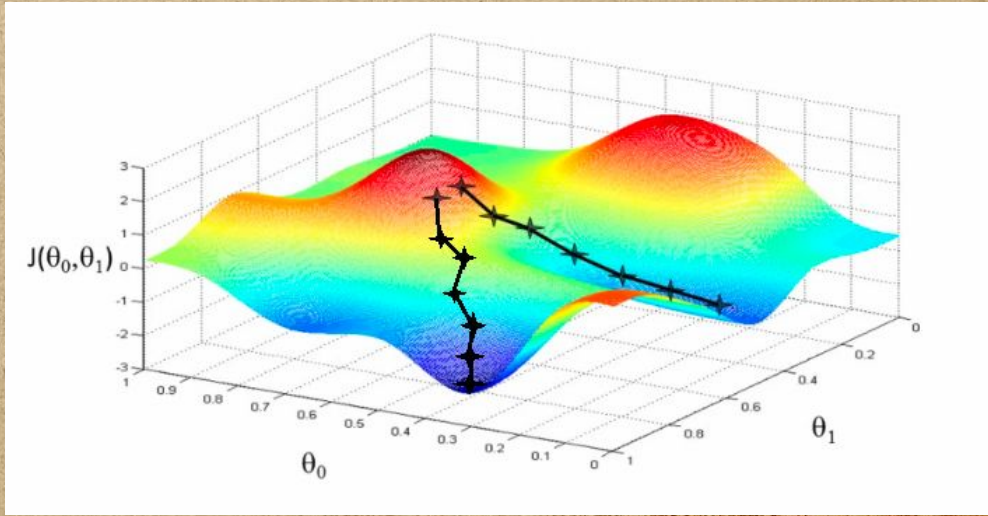
Gradient descent

- Used to minimize the cost function with N parameters given by:

$$J(\theta_0, \theta_1, \ldots, \theta_n)$$

Algorithm

- Make an initial guess
- Change the parameters in order to reduce the cost function at each step
- Repeat until convergence at a local minimum



Gradient descent (continued)

More Formally: (repeat the following step)

$$\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

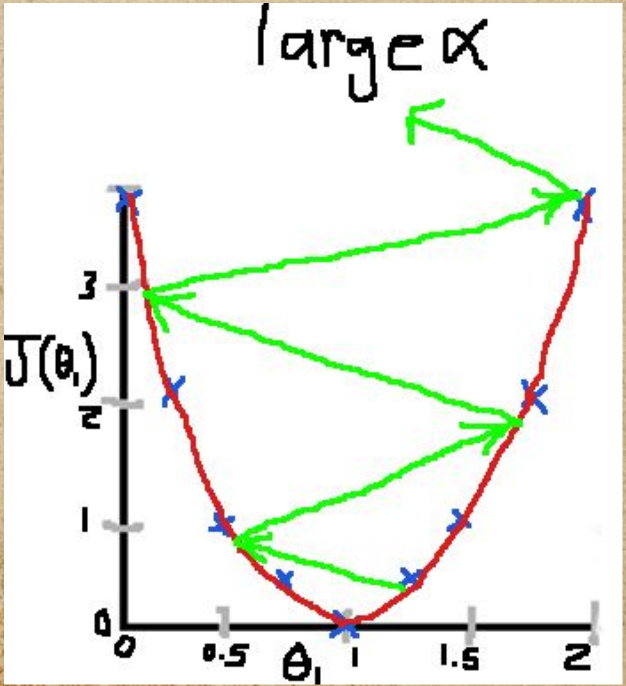
(for j = 0 and j = 1)

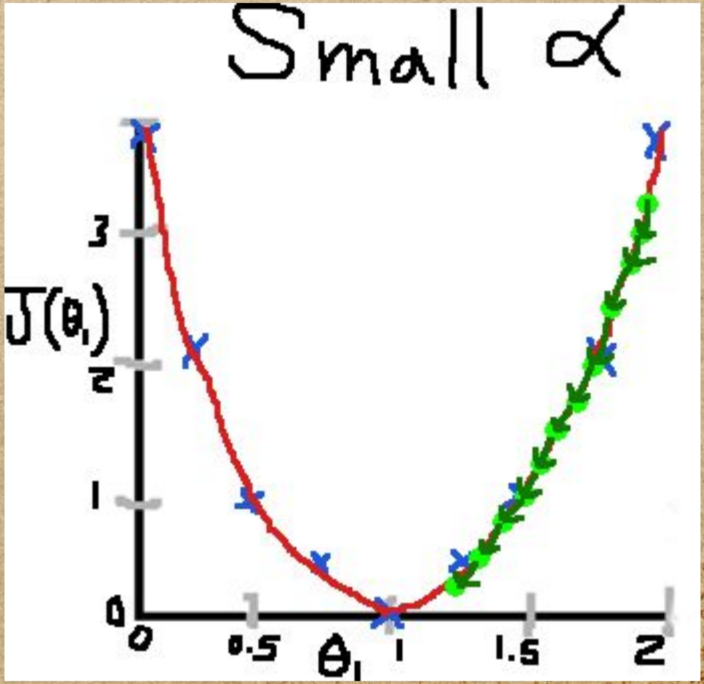
- α (alpha) is the learning rate
  - Controls the step size

- Simultaneously update the parameters

Choosing an Optimal Learning Rate

Consider $J(\theta_1)$





Derivation of the Cost Function

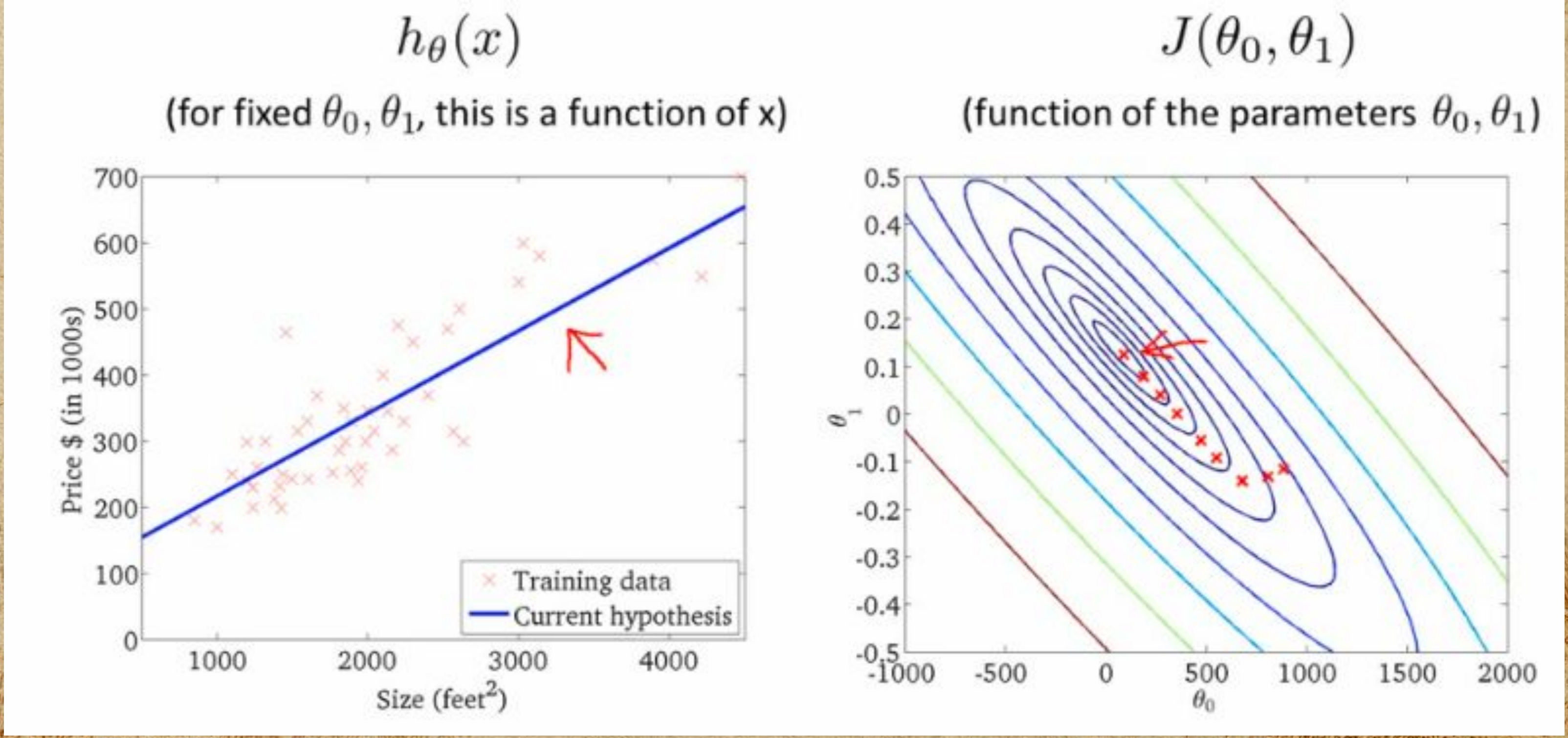
$$\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) = \frac{\partial}{\partial \theta_j} \cdot \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^i) - y^i\right)^2$$

$$= \frac{\partial}{\partial \theta_j} \frac{1}{2m} \sum_{i=1}^{m} \left(\theta_0 + \theta_1 x^i - y^i\right)^2$$

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

Derivation of the Cost Function (continued)

The linear regression function is always a convex function with one minimum



Fun Fact

Batch Gradient Descent: Iterating over all the training data at every step.

Numerical solutions exist (**Normal Equations**), but don’t scale as well.

System of linear equations

Recall: $y = mx + b$

$$y = 3x_1 + 2x_3$$

$$y = 2x_1 - 2x_3$$

$$y = x_2 + x_3$$

Matrices

System of Equations

$$Y = 3x_1 + 2x_3$$
$$Y = 2x_1 - 2x_3$$
$$Y = x_2 + x_3$$

Coefficient Matrix

$$\begin{bmatrix} 3 & 0 & 2 \\ 2 & 0 & -2 \\ 0 & 1 & 1 \end{bmatrix}$$

Matrices

Identity Matrix (3x3)

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Properties:

- 1s along the diagonal
- Same number of rows and columns

Facts:

- It is the optimal end form of row reduced augment matrices
- It is used as one way of calculating a matrix's inverse

Matrices

System of Equations

$$\begin{aligned} 3x_1 + 2x_3 &= 0 \\ 2x_1 - 2x_3 &= 0 \\ x_2 + x_3 &= 0 \end{aligned}$$

Augmented Matrix

$$\left[\begin{array}{ccc|c} 3 & 0 & 2 & 0 \\ 2 & 0 & -2 & 0 \\ 0 & 1 & 1 & 0 \end{array}\right]$$

Matrix Operations

- Swap 2 rows
- Multiple a row by a scalar
- Add 2 rows together to re-write that row

Matrices

$$\left[\begin{array}{ccc|c} 3 & 0 & 2 & 0 \\ 2 & 0 & -2 & 0 \\ 0 & 1 & 1 & 0 \end{array}\right]$$

$$\downarrow$$

$$\left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{array}\right]$$

Matrix Operations to RREF:

- r1 + r2 → r1
- (1 / 5) * r1 → r1
- (- 2) * r1 + r2 → r2
- (- 1 / 2) * r2 → r2
- r1 ←→ r3
- (- 1) * r3 + r2 → r2

Matrices

Matrix addition

$$\begin{bmatrix} 3 & 0 & 2 \\ 2 & 0 & -2 \\ 0 & 1 & 1 \end{bmatrix} + \begin{bmatrix} 3 & 0 & 2 \\ 2 & 0 & -2 \\ 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 6 & 0 & 4 \\ 4 & 0 & -4 \\ 0 & 2 & 2 \end{bmatrix}$$

Matrices

Matrix multiplication

$$\begin{bmatrix} 3 & 0 & 2 \\ 2 & 0 & -2 \\ 0 & 1 & 1 \end{bmatrix} * \begin{bmatrix} 3 \\ 2 \\ 0 \end{bmatrix} = \begin{bmatrix} 9 \\ 6 \\ 2 \end{bmatrix}$$

3x3 **3x1** **3x1**

Vectors

EXAMPLES

Properties:

- N x 1 matrix

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad \textbf{or} \quad \begin{bmatrix} 3 \\ 2 \\ 0 \end{bmatrix}$$

3x1 **3x1**

Facts:

- Can be used to express a system of linear equations

$$A\underline{x} = \underline{b}$$

Matrices

Transpose

$$A = \begin{bmatrix} 6 & 4 & 0 \\ 0 & 0 & 2 \end{bmatrix} \longrightarrow A^T = \begin{bmatrix} 6 & 0 \\ 4 & 0 \\ 0 & 2 \end{bmatrix}$$

2x3 **3x2**

Matrices

Inverse

$$[A \mid I] = \left[\begin{array}{ccc|ccc} 3 & 0 & 2 & 1 & 0 & 0 \\ 2 & 0 & -2 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{array}\right]$$

$$[I \mid A^{-1}] = \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 0.2 & 0.2 & 0 \\ 0 & 1 & 0 & -0.2 & 0.3 & 1 \\ 0 & 0 & 1 & 0.2 & -0.3 & 0 \end{array}\right]$$

Matrix Operations to RREF:

- r1 + r2 → r1
- (1 / 5) * r1 → r1
- (- 2) * r1 + r2 → r2
- (- 1 / 2) * r2 → r2
- r1 ←→ r3
- (- 1) * r3 + r2 → r2

APPLICATION

House sizes:

$$\begin{cases} 2104 \\ 1416 \\ 1534 \\ 852 \end{cases}$$

Have 3 competing hypotheses:

1. $h_\theta(x) = -40 + 0.25x$
2. $h_\theta(x) = 200 + 0.1x$
3. $h_\theta(x) = -150 + 0.4x$

Matrix

Matrix

$$\begin{bmatrix} 1 & 2104 \\ 1 & 1416 \\ 1 & 1534 \\ 1 & 852 \end{bmatrix} \times \begin{bmatrix} -40 & 200 & -150 \\ 0.25 & 0.1 & 0.4 \end{bmatrix} = \begin{bmatrix} 486 & 410 & 692 \\ 314 & 342 & 416 \\ 344 & 353 & 464 \\ 173 & 285 & 191 \end{bmatrix}$$

Prediction of first h_θ

Predictions of 2nd h_θ

SOURCES

- http://www.holehouse.org/mlclass/01_02_Introduction_regression_analysis_and_gr.html
- https://www.mathsisfun.com/algebra/matrix-inverse-row-operations-gauss-jordan.html
- cs.oswego.edu/~kzeller