Machine Learning

Florida International University

Thomaz L. Santana

August 2014

Machine Learning

- Start with fundamental theory
- Gain an intuition into the topic
- Get hands dirty in applying it
- Venture into state-of-the-art machine learning

Machine Learning

- The study of systems that can learn from data
- Types:
  - Supervised learning
  - Unsupervised learning
  - Reinforcement learning
- Focus on simple supervised learning
  - Generate a function that maps inputs to desired outputs

Machine Learning

Two collage students in physics take different paths…
One in theoretical physics and the other in experimental physics.

30 years later they meet again in the lab…
The theoretical physicist tells the experimental physicist: “I told you 30 years ago that you just have to place the atoms close together.”

Fusion Energy… not so easy to apply

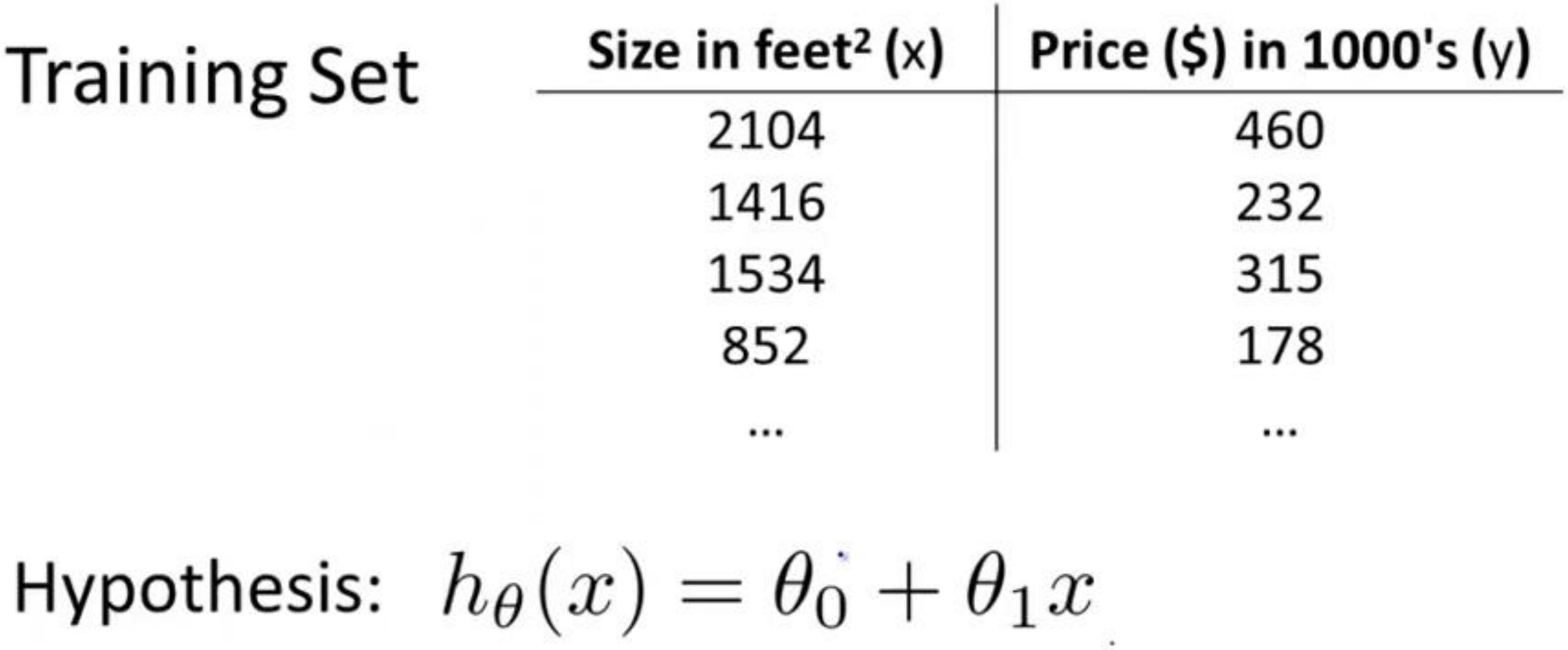
Machine Learning

Goal: Learn the theory and get hands dirty in applying it.

Machine Learning

- Courses on Coursera.org
  - Machine Learning by: Andrew Ng
  - Neural Networks by: Geoffrey Hinton

Simplest Example



Training Set

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

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

Picture from Andrew Ng, Coursera.org machine learning course

Machine Learning

- Simplest example looks just like slope-intercept form (y = mx + b)
- For one feature and two training samples it might as well be

- For one feature and more than two training samples it might as well be considered a regression analysis

- However, for more than one feature and more than two training samples… it starts getting complicated

- Matrix “A” contains the features
- Vector “X” contains the coefficient's that the algorithm will solve for
- Vector “h” is the hypothesis

$$A_{ij} \times X_i = h_i$$

$$\underset{m \times n}{\begin{bmatrix} \quad & \quad \\ & \end{bmatrix}} \times \underset{n \times 1}{\begin{bmatrix} \quad \\ \\ \end{bmatrix}} = \underset{m \times 1}{\begin{bmatrix} \quad \\ \\ \end{bmatrix}}$$

Cost Function

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

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

- J is the cost function also known as:
  - Error Function from Geoffrey’s course
  - Performance Function from MIT course
- m is the number of training examples
- y is the desired output
- h is the hypothesis

Gradient Descent

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

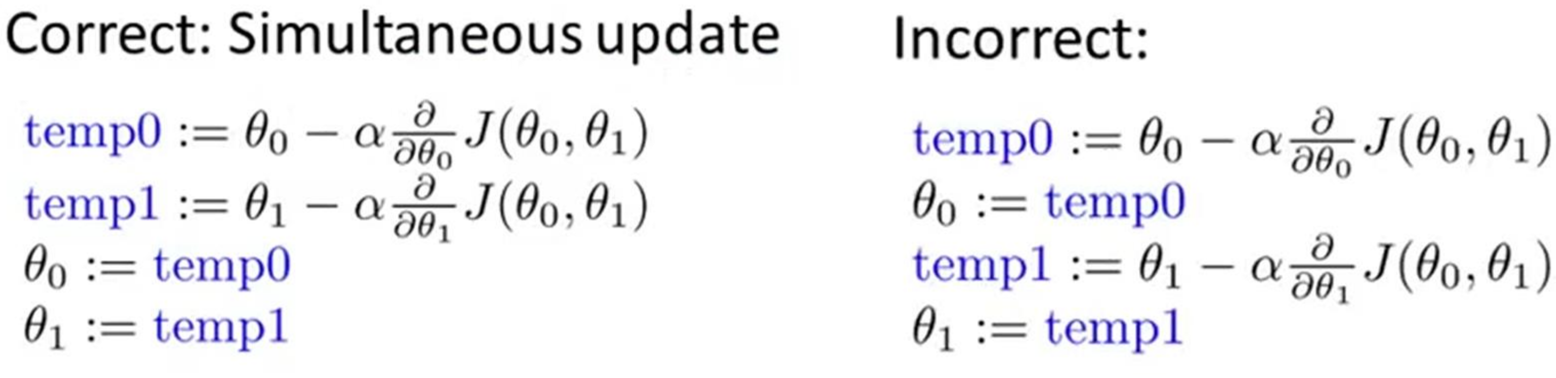
- θ_j Coefficients to be solved for to minimize J
- α Learning rate

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

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

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

Updating Thetas



Correct: Simultaneous update

$\text{temp0} := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$
$\text{temp1} := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$
$\theta_0 := \text{temp0}$
$\theta_1 := \text{temp1}$

Incorrect:

$\text{temp0} := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$
$\theta_0 := \text{temp0}$
$\text{temp1} := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$
$\theta_1 := \text{temp1}$

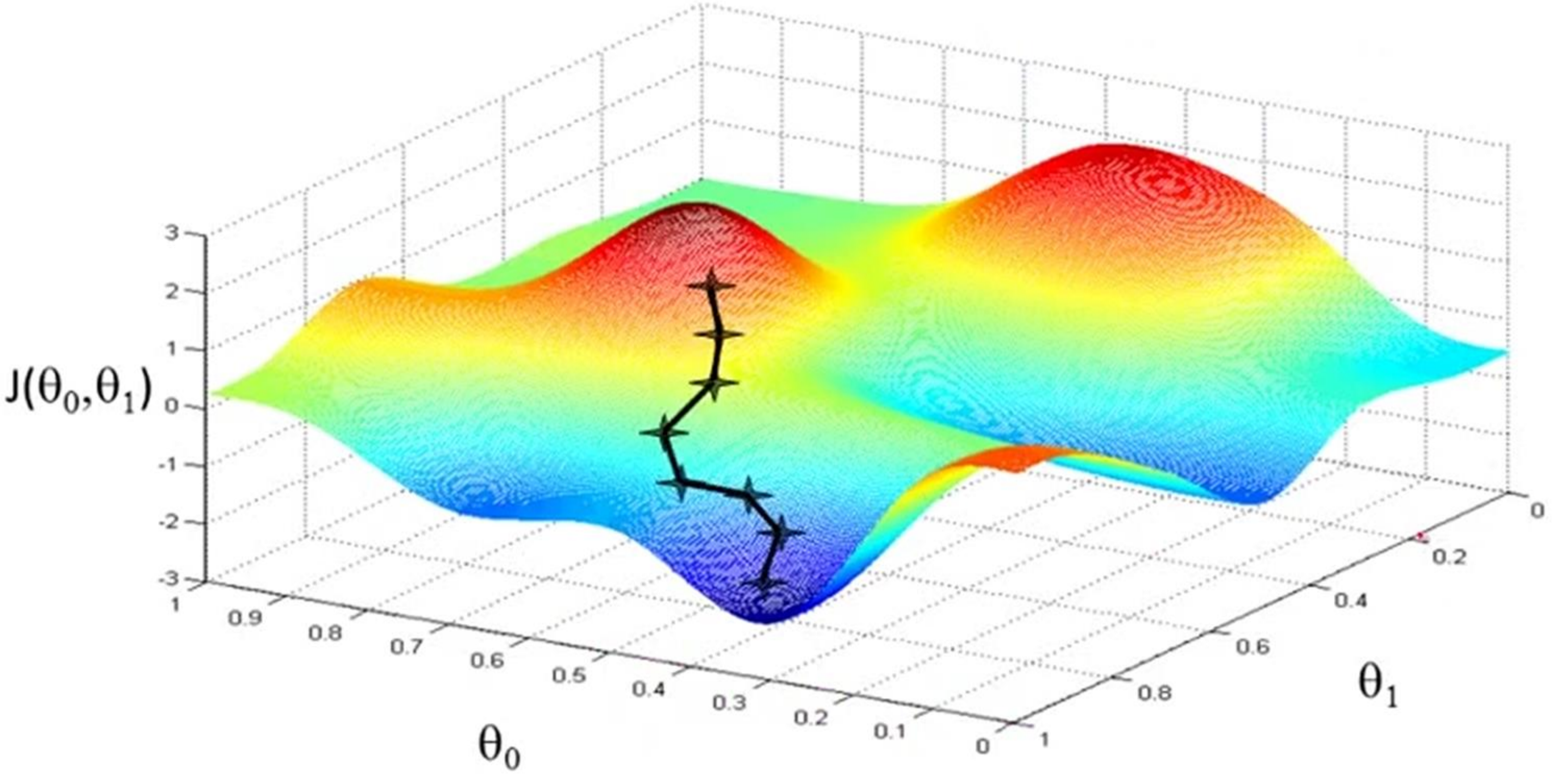
Picture from Andrew Ng, Coursera.org machine learning course

Updating Thetas

- It is proper to compute the partial derivative with respect to each feature all at once then update the thetas all at once
- The initial theta value can be any random number
- The gradient descent method is not guaranteed to produce an absolute minimum but rather a local minimum

Local Minimum

- Values depend on how the thetas were initialized



Picture from Andrew Ng, Coursera.org machine learning course

Learning Rate & Features

- Features are data points for a specific training example. e.g. house size: 950 sq-ft, 2 car garage, 3 bedroom, etc…
- Learning rate is tricky… Too large makes the function oscillate. Too small takes allot of iterations to converge.

Learning Rate & Features

- Examples learned in “getting hands dirty” in machine learning

- Features: 2 car garage, 1,500 sq-ft
- A learning rate too small would cause the “sq-ft” feature to take ~1,000 times more iterations
- A learning rate too large would cause the “garage” feature to oscillate and prevent the algorithm from reaching a minimum

Learning Rate & Features

- “Massage” the features to be within a order of magnitude of each other e.g. multiply the garage feature by a thousand to be within a order of magnitude of the sq-ft feature
- Learning rate must be small enough to accommodate the smallest feature
- A feature one order of magnitude larger than the others would cause the algorithm one order of magnitude more iterations

Example

$$\begin{bmatrix}1 & 950 & 1\\ 1 & 1500 & 1\\ 1 & 2000 & 2\end{bmatrix} X \begin{bmatrix}\theta_0\\ \theta_1\\ \theta_2\end{bmatrix} = \begin{bmatrix}h_0\\ h_1\\ h_2\end{bmatrix}\Bigg\} \text{3 training examples}$$

"b" in $y = mx + b$

house size

garage

Initialized zeros

$$Y_1 = 180{,}000 \;, Y_2 = 240{,}000 \;, Y_3 = 310{,}000$$

house Prices for the 3 examples

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

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

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

$$\theta_0 = 0 - \alpha \left[(0 - 180{,}000)\tfrac{1}{3} + (-240{,}000)\tfrac{1}{3} + (310{,}000)\tfrac{1}{3} \right]$$

$$\theta_1 = 0 - \alpha \left[(-180k)\frac{950}{3} + (-240k)\frac{1500}{3} + (-310k)\frac{2000}{3} \right]$$

$$\theta_2 = 0 - \alpha \left[(-180k)\frac{1}{3} + (-240k)\frac{1}{3} + (-310k)\frac{2}{3} \right]$$

$$\theta_0 = 0 - \alpha(-243{,}333)$$

$$\theta_1 = 0 - \alpha(-383{,}666{,}666)$$

$$\theta_2 = 0 - \alpha(-346{,}666)$$

Getting Hands Dirty

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

Snippet of code

```
def GradientDescent(self):
        hypo = np.dot(self.A,self.theta)              # the hypothesis vector
        m = self.A.shape[0]                            # the number of training examples
        n = self.A.shape[1]                            # the number of features
        diff = np.add(hypo, self.y*(-1))               # a vector of the difference between hypo and target value
        ab = np.multiply(diff, self.A)
        k = 0
        while k < n:
            self.theta[k,0] -= self.alpha/float(m) * ab.sum(axis=0)[k,]
            k+=1
```

np is Numpy a Python matrix library

A is feature matrix

y is target output

Data Source

Apple Support Communities

Welcome, Guest | Sign in

Apple Support Communities > iPhone > Using iPhone > Discussions

Is there a way to turn off 3g on iOS 6.1.3?

9 Views **1 Reply** Latest reply: Apr 14, 2013 3:43 PM by Allan Sampson

emirgrbich

Level 1 (0 points)

Apr 14, 2013 3:04 PM

If not, how do I restore to previois version without LTE support so that my battery can again
last longer?

Categories: Wi-Fi, 3G and Bluetooth Tags: 3g

I have this question too (0)

Allan Sampson Central Texas

Level 10 (114,680 points)

Re: Is there a way to turn off 3g on iOS 6.1.3?

Apr 14, 2013 3:43 PM (in response to emirgrbich)

This depends on your carrier.

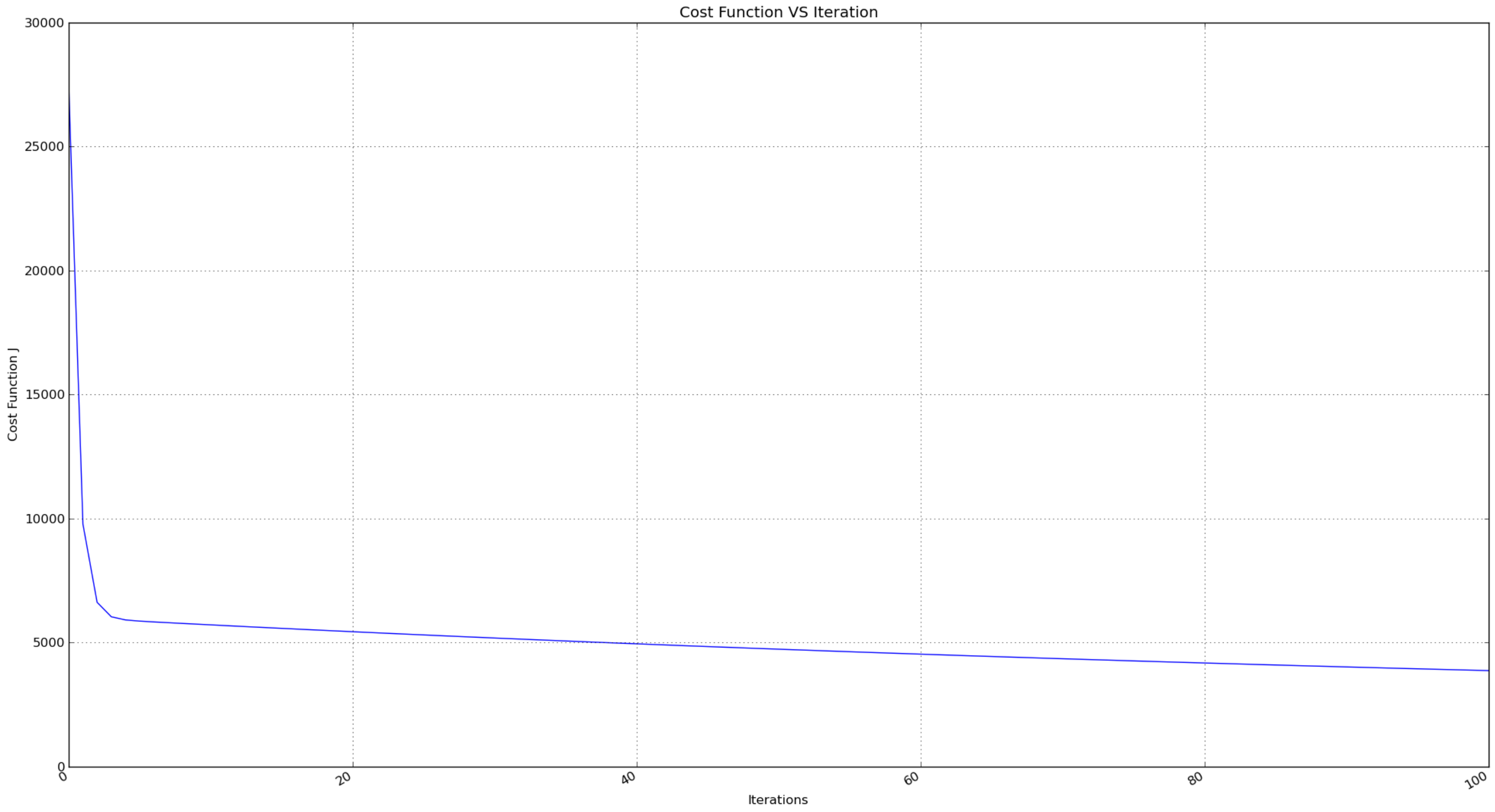
If your carrier is AT&T, the option is no longer available.

Like (0)

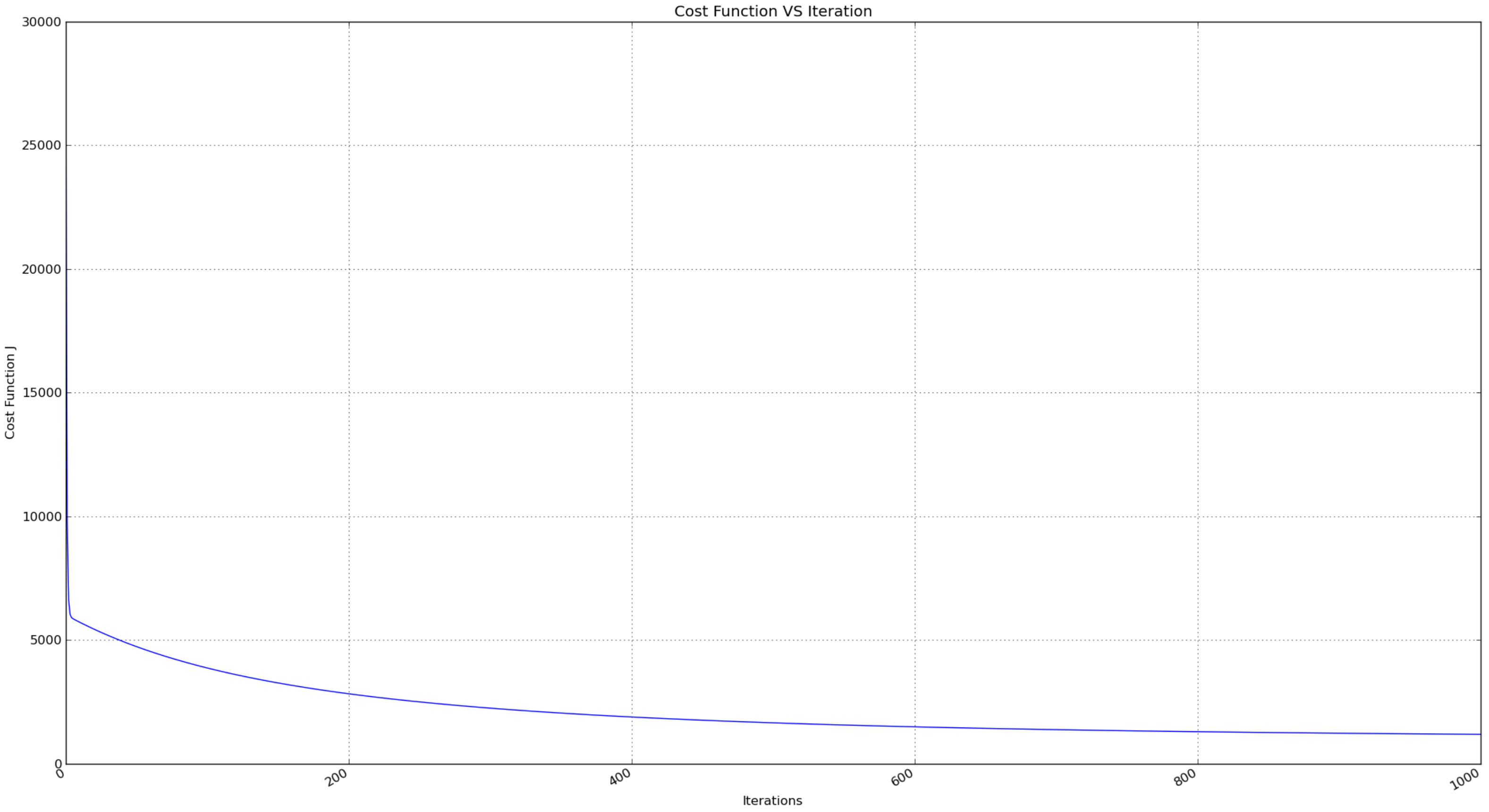
Details

- Training data from Jan, 2007 to Feb 1, 2012
- Testing data from Feb 2, 2012 to Feb 27, 2013
- Features used:
    - Inflation, Forum post volume (daily activity), volume of Google searches for Apples products, volume of Google searches for Microsoft, and 56 randomly selected words with sufficient volume from the word lists

Cost Function VS Iteration



Cost Function VS Iteration



Features

Features	Theta
b	0.277
Apple Trends	4.708
Microsoft Trends	-0.156
Post Volume	-0.664
Inflation	8.475
afraid	-0.272
annoying	-0.301
avoid	-0.043
bad	-0.222
break	-0.034
broken	0.122
bug	0.645
corrupted	-0.043
dead	-0.194
error	0.253
expensive	-0.152
failed	-0.302

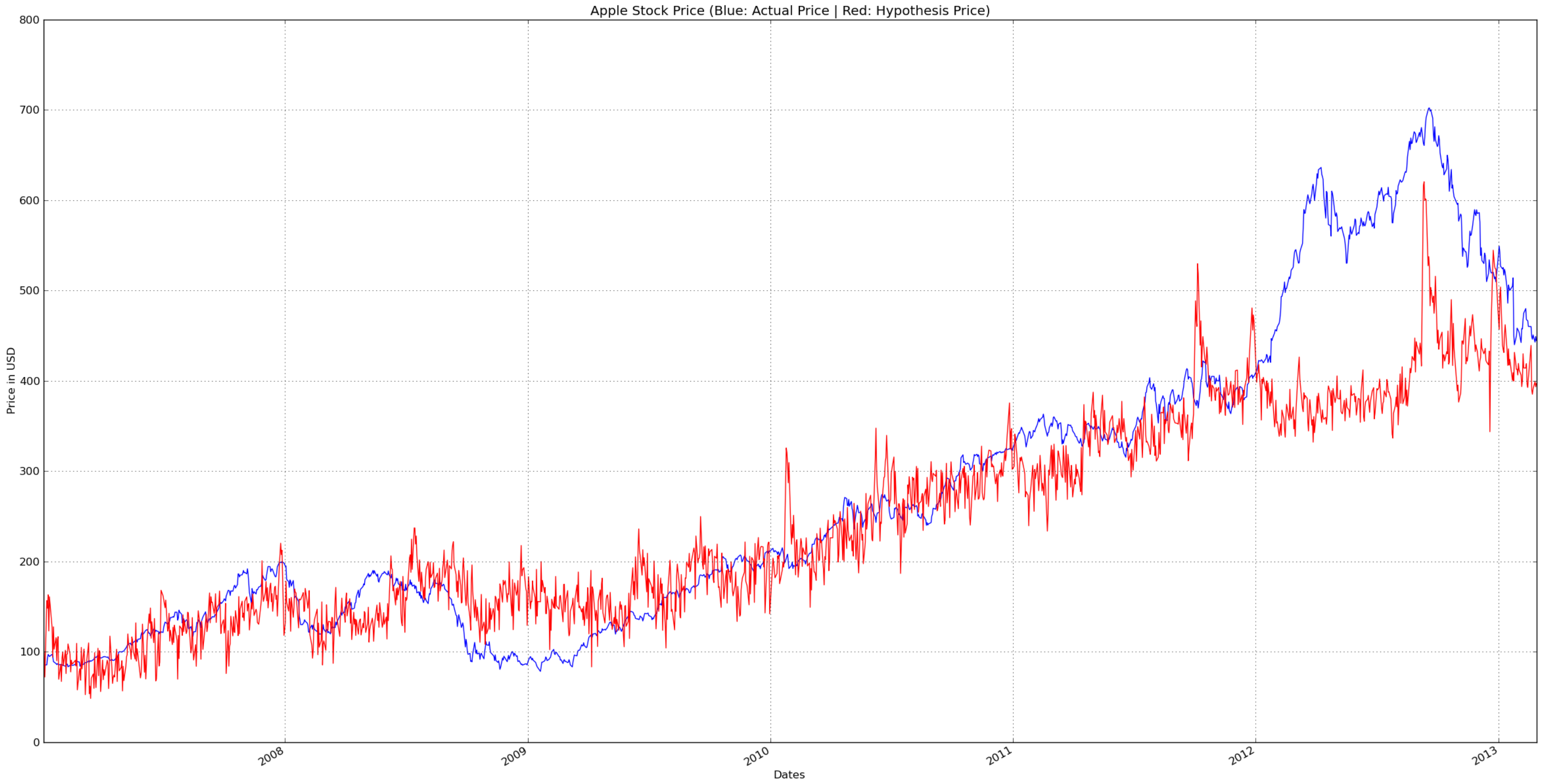
Positive Theta: Positive correlation, the larger the feature is the higher the predicted stock price will be.

Negative Theta: Negative correlation, the larger the feature is the lower the predicted stock price will be.

Magnitude of Theta: A large absolute value means there is a strong correlation.

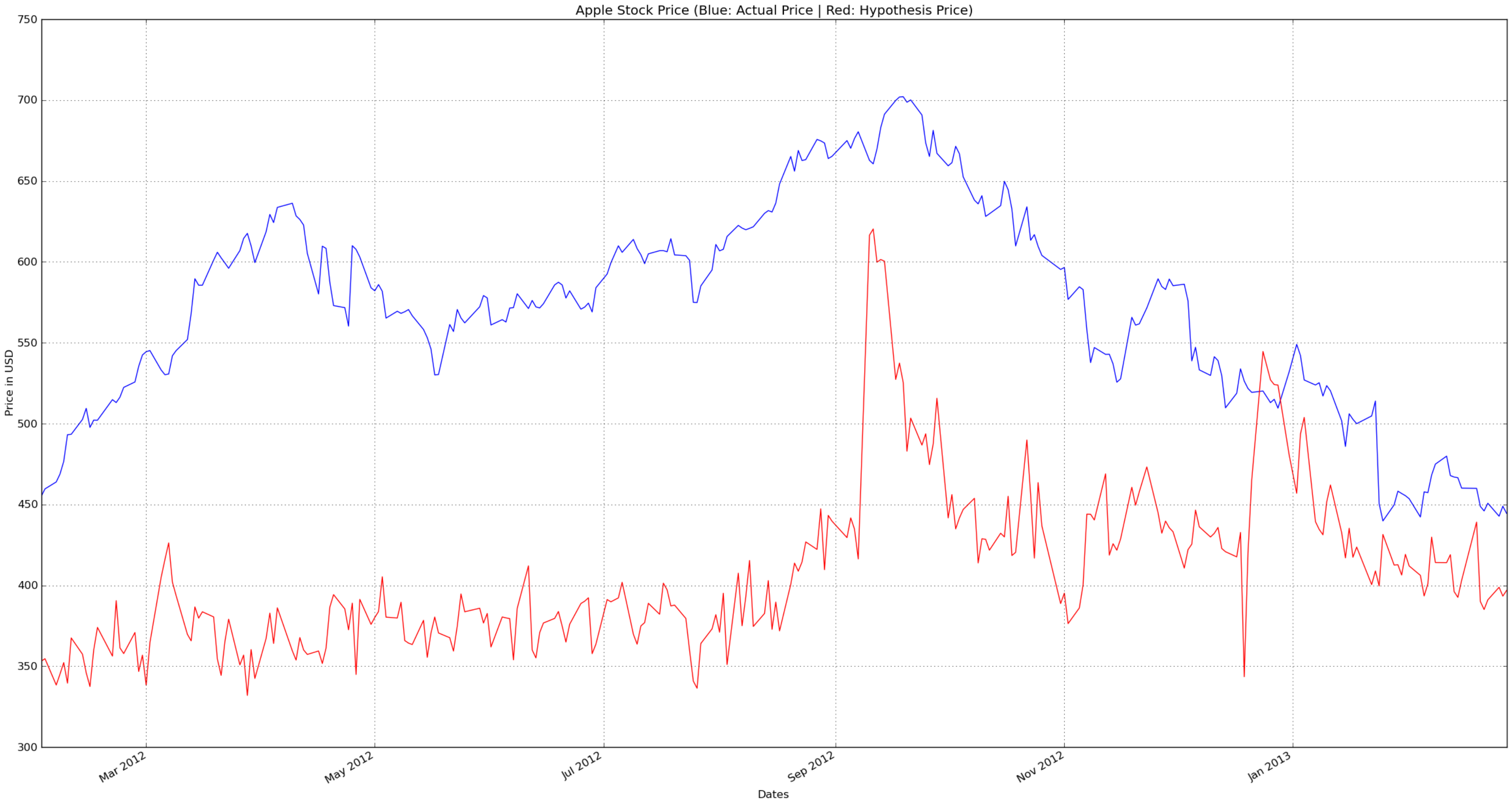
Features	Theta
fix	1.118
missing	0.024
order	-0.235
problem	-1.431
problems	-0.464
trouble	0.168
unfourtunately	0.242
wrong	-0.612
annoying	-0.301
fustrating	0.064
quit	-0.343
better	-1.211
easy	-0.365
exellent	0.356
good	0.298
great	-0.391
happy	-0.443
perfect	0.123
perfectly	0.236
best	0.091
crash	-0.095
dont	-1.734

Features	Theta
hard	-1.782
lost	0.425
love	-0.276
luck	0.208
missing	0.024
no	1.869
not	1.909
smart	0.141
sorry	0.442
strange	-0.032
weird	-0.050
yes	-0.451
doesnt	0.240
update	-0.911
free	-0.593
upgrade	-0.231
buy	0.141
only	-2.007
security	0.431
isnt	0.194
downgrade	0.036
hate	-0.117



Blue line: Actual Stock Price Red line: Hypothesis Stock Price
X-axis: Dates (Jan 1, 2007 – Feb 27, 2013) Y-Axis: Stock Price in USD

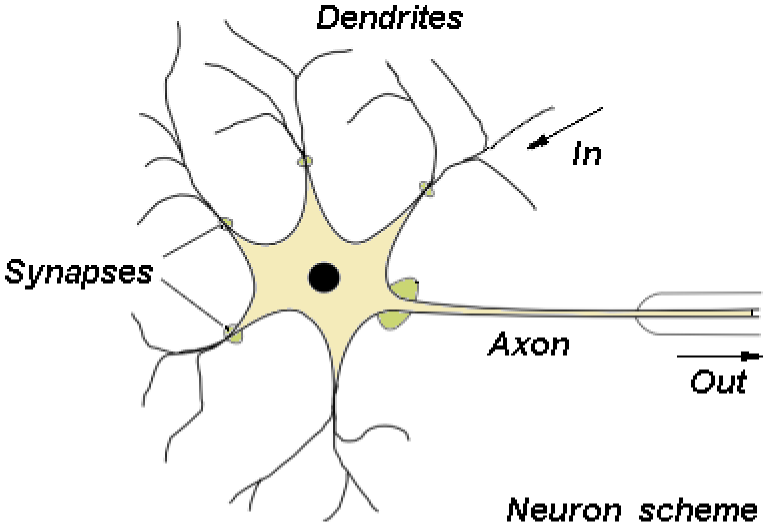
Learning Data: Jan 1, 2007 – Feb 1, 2012 Testing Data: Feb 2, 2012 – Feb 27, 2013

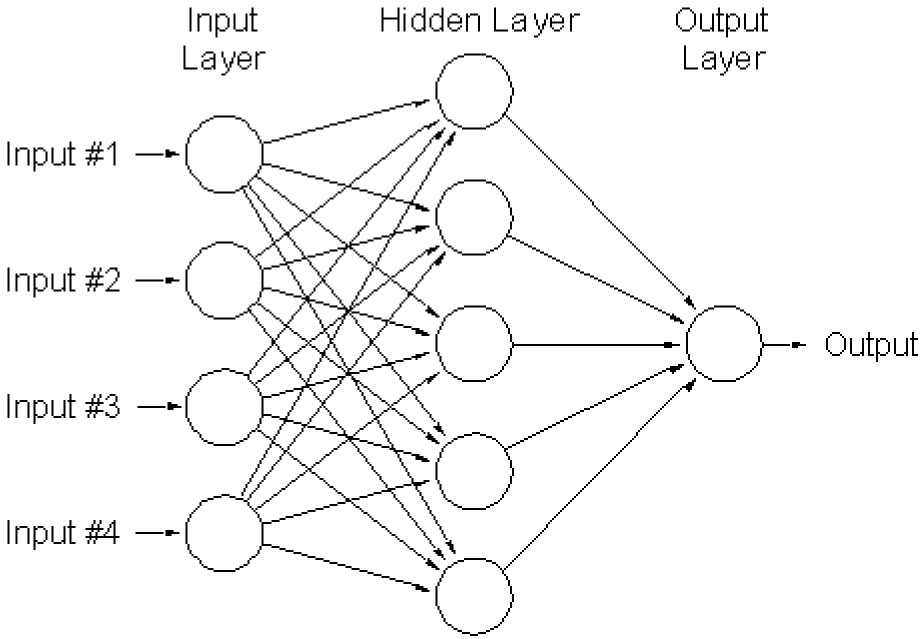


Blue line: Actual Stock Price   Red line: Hypothesis Stock Price

X-axis: Dates (Feb 2, 2012 – Feb 27, 2013)   Y-Axis: Stock Price in USD

Overview of Neural Networks





Overview of Neural Networks