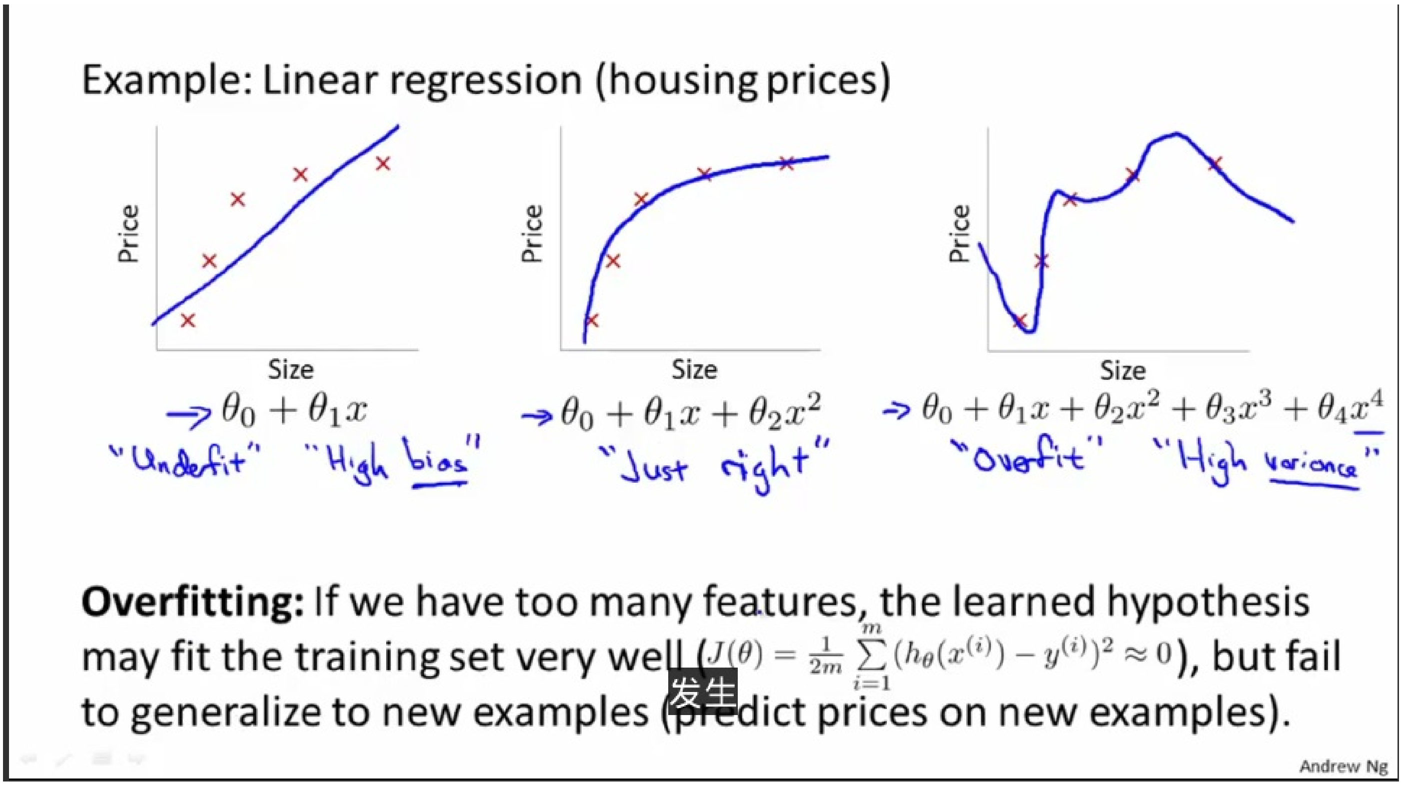
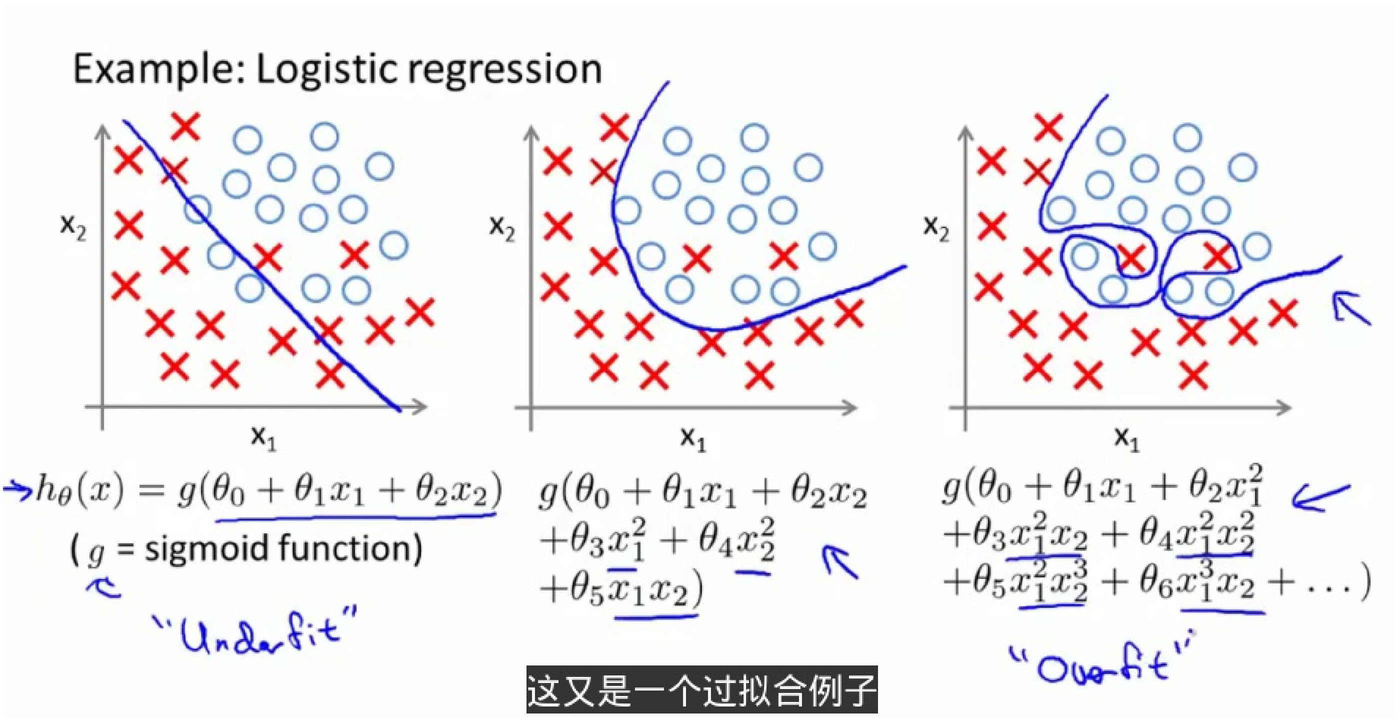
1. 过拟合（Overfit）。太多的变量虽然代价函数非常接近0, 但是使得预测偏差变大。线性回归中：



逻辑回归中：



2. 解决过拟合：

（1）减少特征数量。

-手动选择保持特征

-模型选择算法

问题是：舍弃掉一些特征便舍弃掉一些信息。

（2）正规化（Regularization）

-保持特征数量，但是减少theta的影响（减少theta的值）

-特征数量多的时候很有效，每个特征都有影响。

Addressing overfitting:

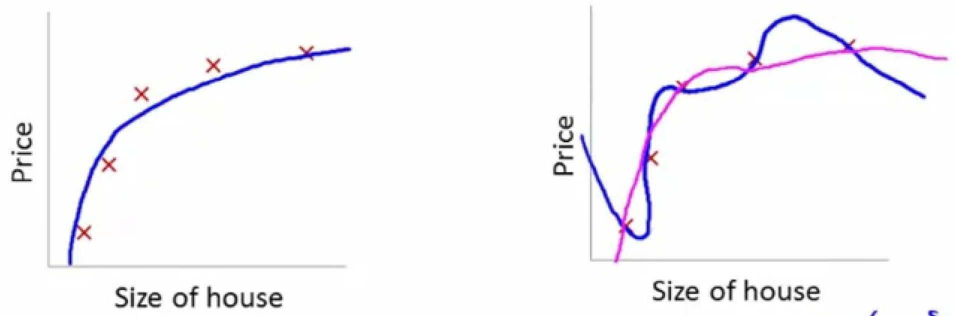
Options:

1. Reduce number of features.
   - Manually select which features to keep.
   - Model selection algorithm (later in course).
2. Regularization.
   - Keep all the features, but reduce magnitude/values of parameters θ_j.
   - Works well when we have a lot of features, each of which contributes a bit to predicting y.

正如我们在房价的例子中看到的那样

通过减小theta的值来减小某些特征的贡献：

Intuition



$\theta_0 + \theta_1 x + \theta_2 x^2$

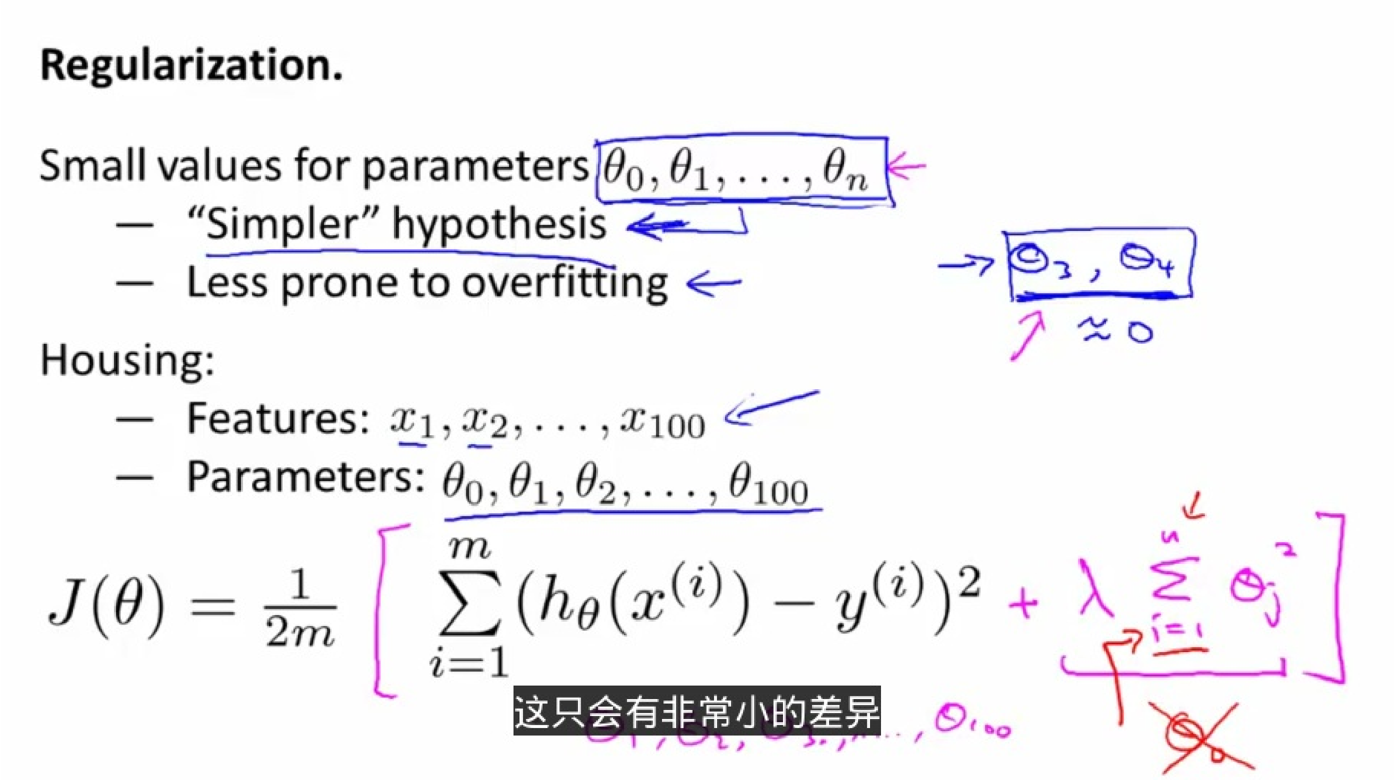
$\theta_0 + \theta_1 x + \theta_2 x^2 + \cancel{\theta_3 x^3} + \cancel{\theta_4 x^4}$

Suppose we penalize and make θ_3, θ_4 really small.

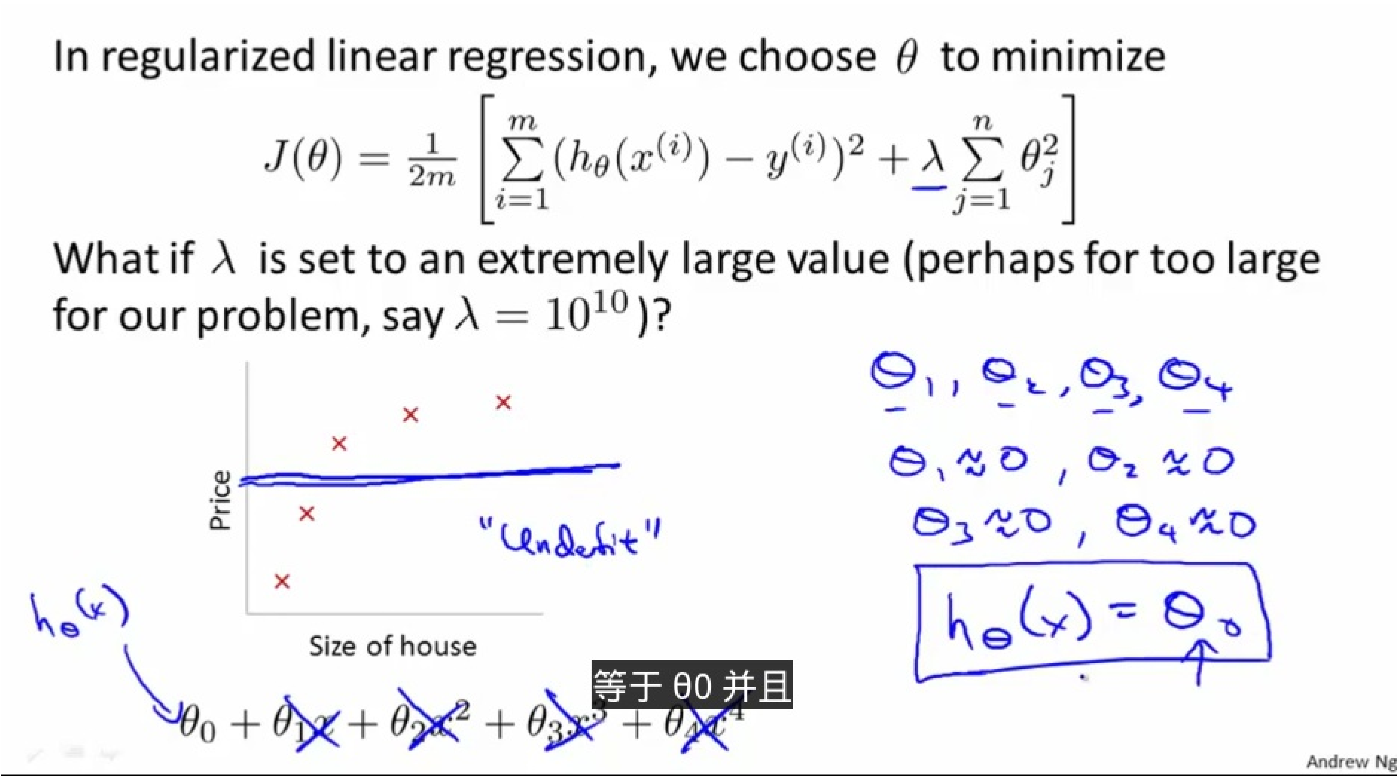
$$\min_\theta \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 + 1000\,\theta_3^2 + 1000\,\theta_4^2$$

$\theta_3 \approx 0$ 这些很小的项 贡献很小 $\theta_4 \approx 0$

注意，不对theta0进行正规化：



如果lambda太大，thetaj(j~=0)都被训练为约为h（x）约为theta0，模型欠拟合。



正则化线性回归：梯度下降法：

Gradient descent

θ_0 $\theta_1, \theta_2, \dots, \theta_n$

$\frac{\partial}{\partial \theta_0} J(\theta)$

Repeat {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \boxed{\sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_0^{(i)}}$$

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

$$(j = \cancel{0}, 1, 2, 3, \dots, n)$$

}

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

$1 - \alpha \frac{\lambda}{m} < 1$ 更小了 0.99 $\theta_j \times 0.99$ θ_j^2

正则化正规方程法：

Non-invertibility (optional/advanced).

Suppose $m \le n$,

(#examples) (#features)

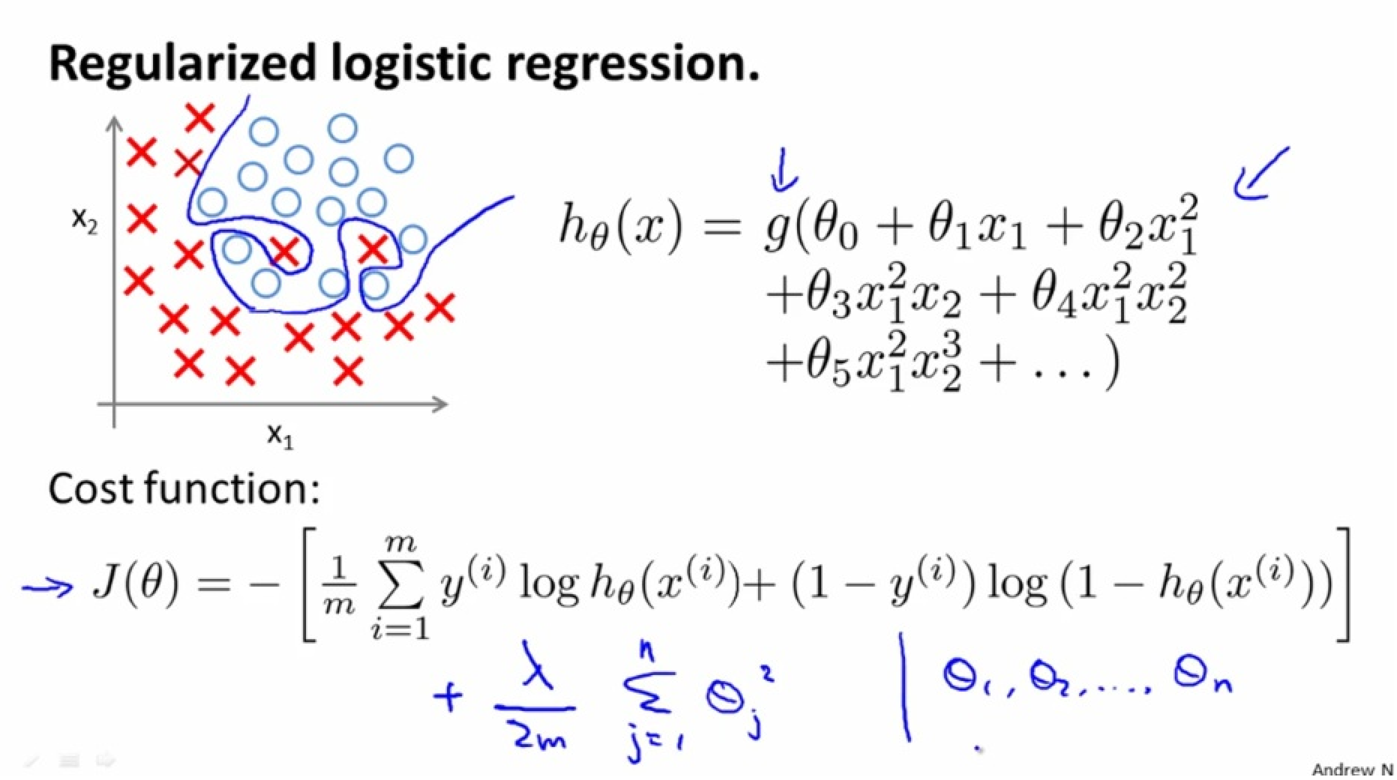
$$\theta = \underbrace{(X^T X)^{-1}}_{\text{non-invertible / singular}} X^T y$$

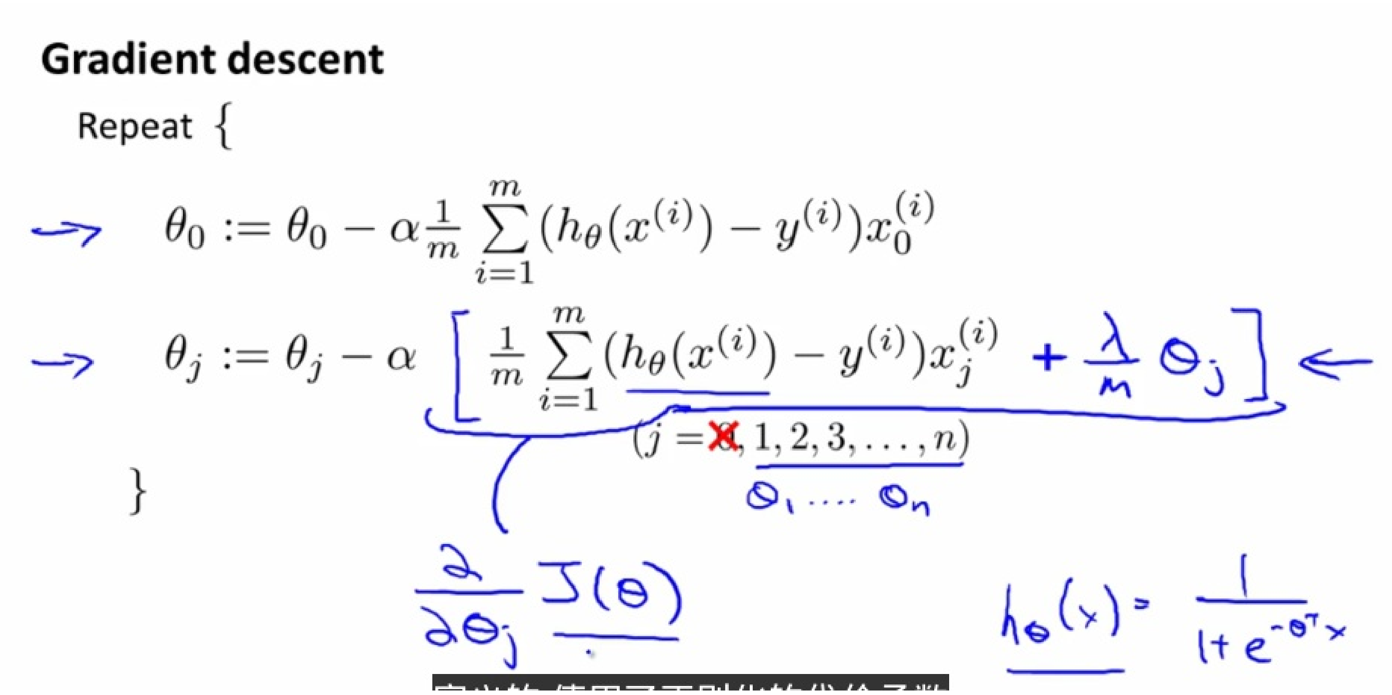
pinv

If $\lambda > 0$,

$$\theta = \left(X^T X + \lambda \begin{bmatrix} 0 & & & \\ & 1 & & \\ & & 1 & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix} \right)^{-1} X^T y$$

正则化逻辑回归：


Regularized logistic regression.
x_2
x_1
$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \dots)$
Cost function:
$J(\theta) = -\left[\frac{1}{m}\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1-y^{(i)}) \log (1-h_\theta(x^{(i)}))\right]$
$+ \frac{\lambda}{2m}\sum_{j=1}^{n} \theta_j^2$
$\theta_1, \theta_2, \dots, \theta_n$
Andrew N


Gradient descent
Repeat {
$\theta_0 := \theta_0 - \alpha \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_0^{(i)}$
$\theta_j := \theta_j - \alpha \left[\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)} + \frac{\lambda}{m}\theta_j\right]$
$(j = \cancel{0}, 1, 2, 3, \dots, n)$
$\theta_1, \dots, \theta_n$
}
$\frac{\partial}{\partial \theta_j} J(\theta)$
$h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$

高级优化算法的正则化：

Advanced optimization

fminunc(@costFunction...) $\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix}$ theta(1) theta(2) theta(n+1)

```
function [jVal, gradient] = costFunction(theta)
    jVal = [code to compute J(θ)];
```

$$J(\theta) = \left[-\frac{1}{m}\sum_{i=1}^{m} y^{(i)}\log\left(h_\theta(x^{(i)})\right) + (1-y^{(i)})\log 1 - h_\theta(x^{(i)})\right] + \boxed{\frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2}$$

```
    gradient(1) = [code to compute ∂/∂θ0 J(θ)];
```

$$\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_0^{(i)}$$

```
    gradient(2) = [code to compute ∂/∂θ1 J(θ)];
```

$$\left(\boxed{\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_1^{(i)}} + \frac{\lambda}{m}\theta_1\right)$$

$J(\theta)$

```
    gradient(3) = [code to compute ∂/∂θ2 J(θ)];
```

$$\left(\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_2^{(i)} + \frac{\lambda}{m}\theta_2\right)$$

⋮

```
    gradient(n+1) = [code to compute ∂/∂θn J(θ)];
```

Andrew Ng