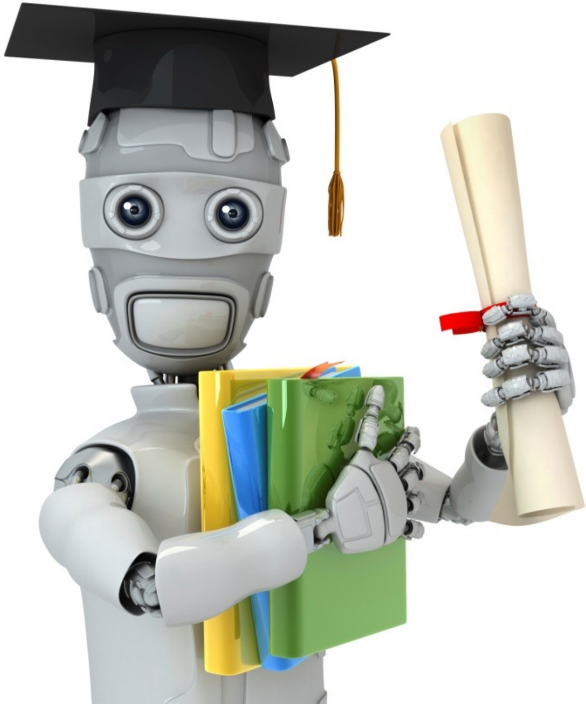
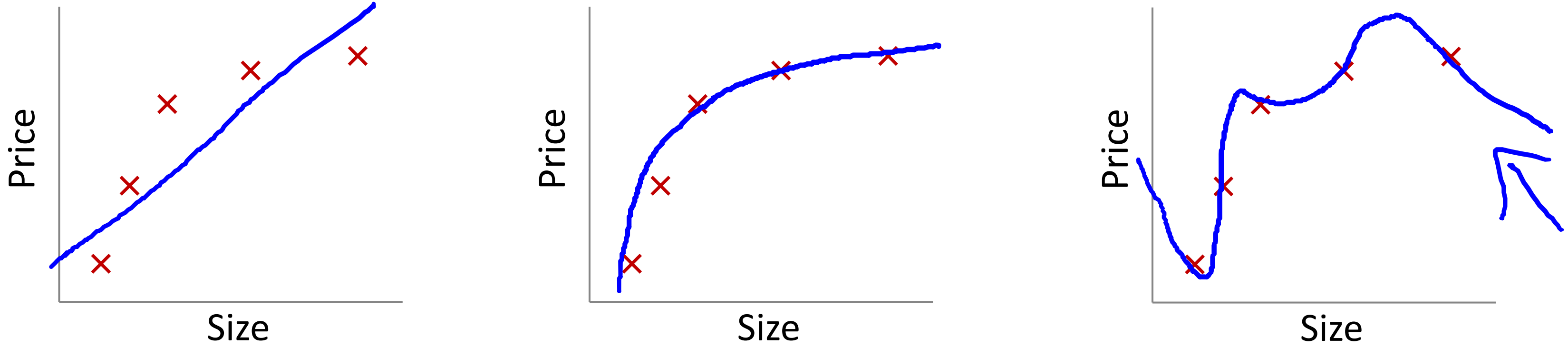


Machine Learning

Regularization

The problem of overfitting

Example: Linear regression (housing prices)



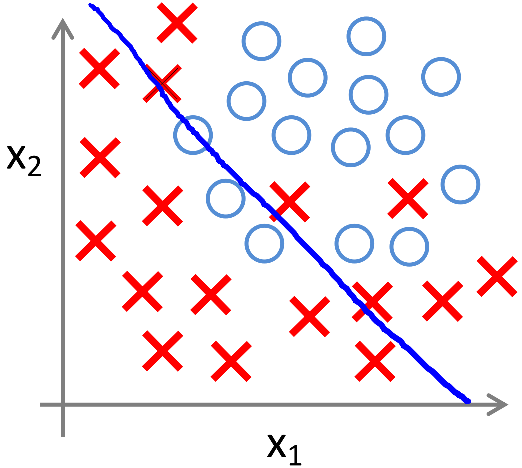
$\theta_0 + \theta_1 x$ — "Underfit" "High bias"

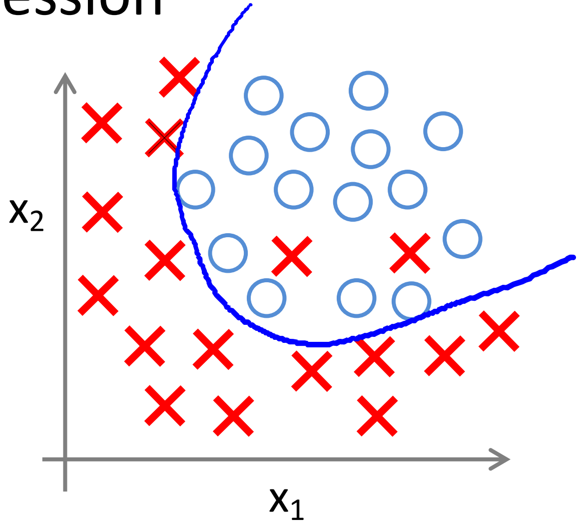
$\theta_0 + \theta_1 x + \theta_2 x^2$ — "Just right"

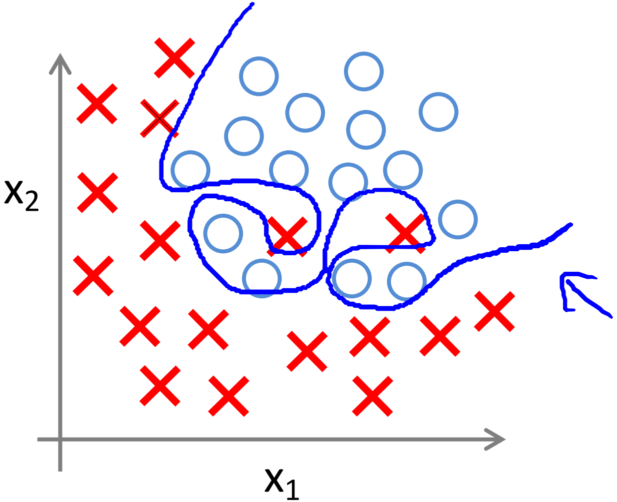
$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$ — "Overfit" "High variance"

Overfitting: If we have too many features, the learned hypothesis may fit the training set very well ($J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 \approx 0$), but fail to generalize to new examples (predict prices on new examples).

Example: Logistic regression







$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

(g = sigmoid function)

"Underfit"

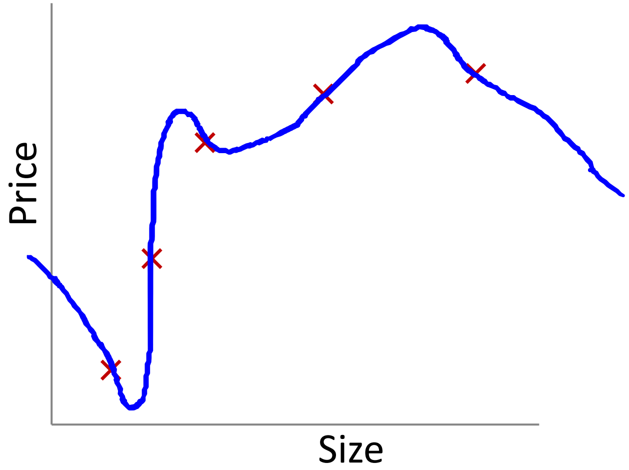
$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2)$$

$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \theta_6 x_1^3 x_2 + \dots)$$

"Overfit"

Addressing overfitting:

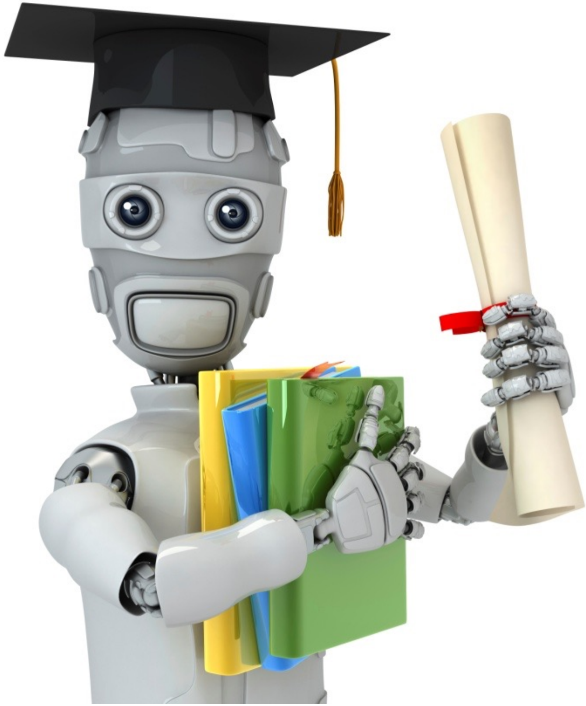
$x_1 =$ size of house
$x_2 =$ no. of bedrooms
$x_3 =$ no. of floors
$x_4 =$ age of house
$x_5 =$ average income in neighborhood
$x_6 =$ kitchen size
$\vdots$
x_{100}



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.

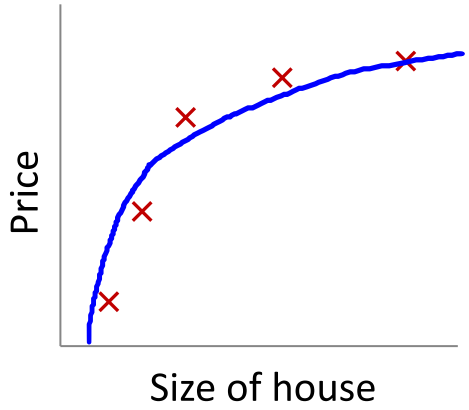


Machine Learning

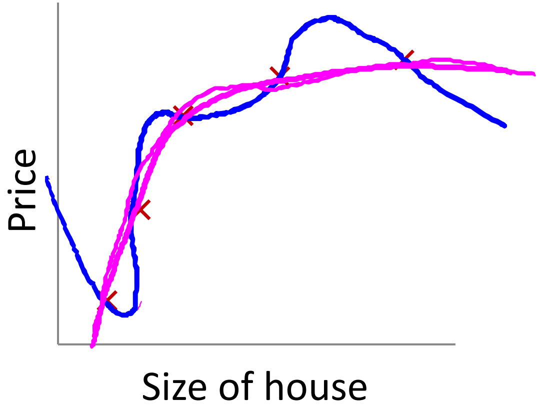
Regularization

Cost function

Intuition



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



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

Suppose we penalize and make θ_3, θ_4 really small.

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

$\theta_3 \approx 0 \qquad \theta_4 \approx 0$

Regularization.

Small values for parameters $\boxed{\theta_0, \theta_1, \ldots, \theta_n}$

- "Simpler" hypothesis
- Less prone to overfitting

$\boxed{\theta_3, \theta_4} \approx 0$

Housing:

- Features: $x_1, x_2, \ldots, x_{100}$
- Parameters: $\theta_0, \theta_1, \theta_2, \ldots, \theta_{100}$

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

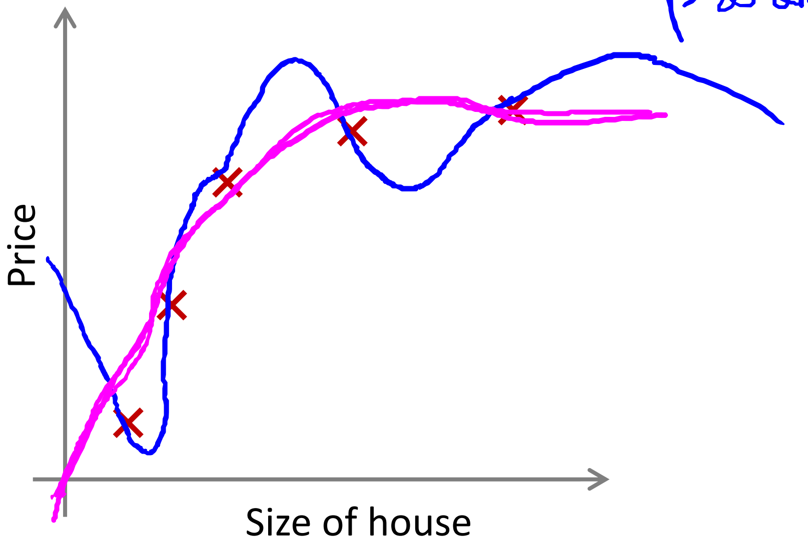
$\cancel{\theta_0}$ $\theta_1, \theta_2, \theta_3, \ldots, \theta_{100}$ $\cancel{\theta_0}$

Regularization.

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

regularization parameter

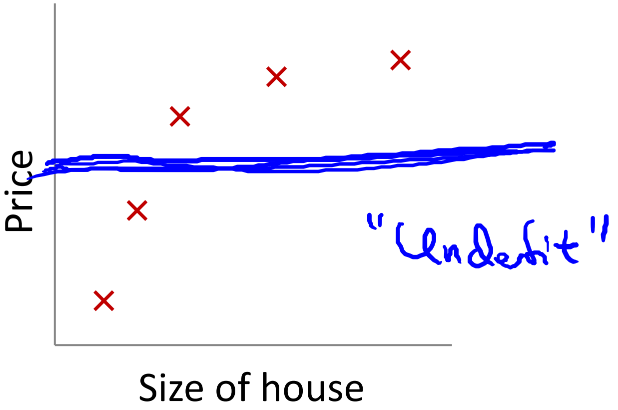
$$\min_\theta J(\theta)$$



In regularized linear regression, we choose θ to minimize

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

What if λ is set to an extremely large value (perhaps for too large for our problem, say $\lambda = 10^{10}$)?



$h_\theta(x)$

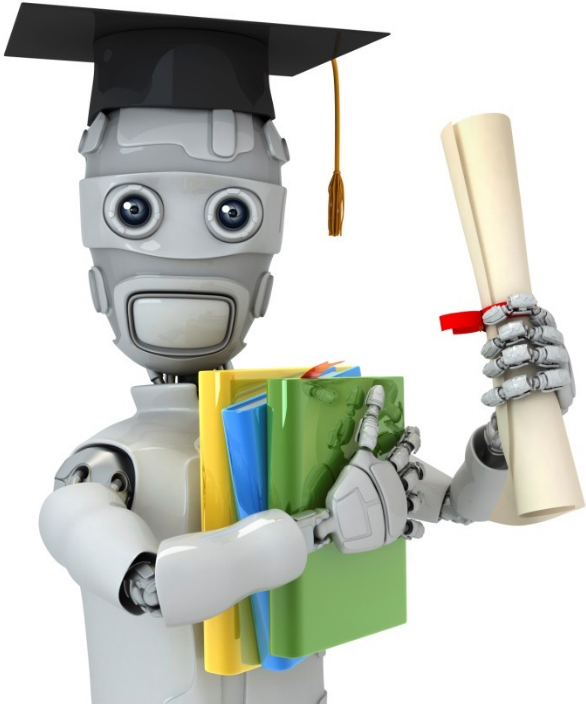
$$\theta_0 + \cancel{\theta_1 x} + \cancel{\theta_2 x^2} + \cancel{\theta_3 x^3} + \cancel{\theta_4 x^4}$$

$\underline{\theta_1}, \underline{\theta_2}, \underline{\theta_3}, \underline{\theta_4}$

$\theta_1 \approx 0, \theta_2 \approx 0$

$\theta_3 \approx 0, \theta_4 \approx 0$

$$\boxed{h_\theta(x) = \theta_0}$$



Machine Learning

Regularization

Regularized linear regression

Regularized linear regression

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

$$\min_{\theta} J(\theta)$$

Gradient descent

$\theta_0 \quad \theta_1, \theta_2, \ldots, \theta_n$

Repeat $\{$

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

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

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

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

$\}$

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

$\rightarrow J(\theta)$

$1 - \alpha \frac{\lambda}{m} < 1 \qquad 0.99 \qquad \theta_j \times 0.99$

$\boxed{\theta_j^2}$

Normal equation

$$\underline{X} = \begin{bmatrix} (x^{(1)})^T \\ \vdots \\ (x^{(m)})^T \end{bmatrix} \qquad y = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(m)} \end{bmatrix} \quad \mathbb{R}^m$$

$m \times (n+1)$

$$\frac{\partial}{\partial \theta_j} J(\theta) \stackrel{\text{set}}{=} 0$$

$$\rightarrow \min_\theta \underline{J(\theta)}$$

$$\rightarrow \theta = \Big(X^T X + \lambda \underbrace{\begin{bmatrix} 0 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix}}_{(n+1)\times(n+1)} \Big)^{-1} X^T y$$

E.g. $n=2$ $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Non-invertibility (optional/advanced).

Suppose $m \leq n$,

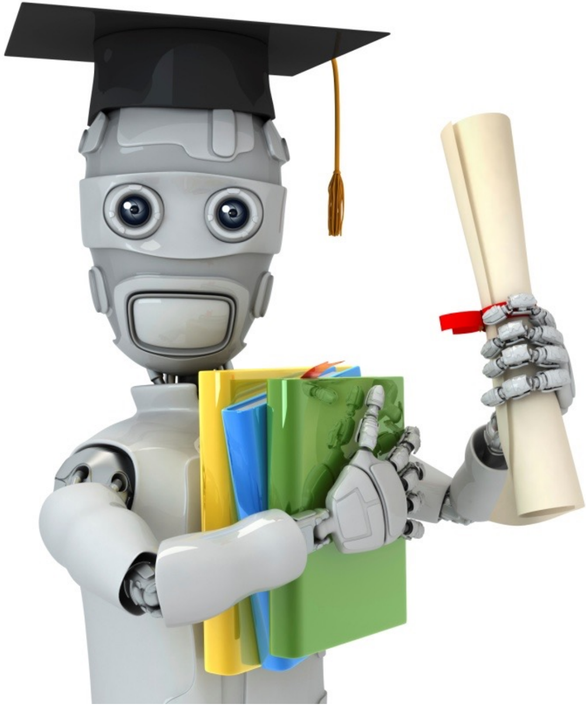
(#examples) (#features)

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

pinv inv

If $\lambda > 0$,

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

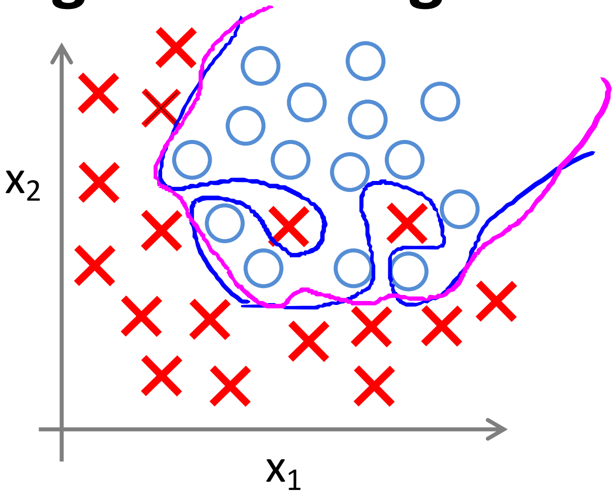


Machine Learning

Regularization

Regularized logistic regression

Regularized logistic regression.



$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \dots)$$

Cost function:

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

$\boxed{\theta_1, \theta_2, \dots, \theta_n}$

Gradient descent

Repeat $\{$

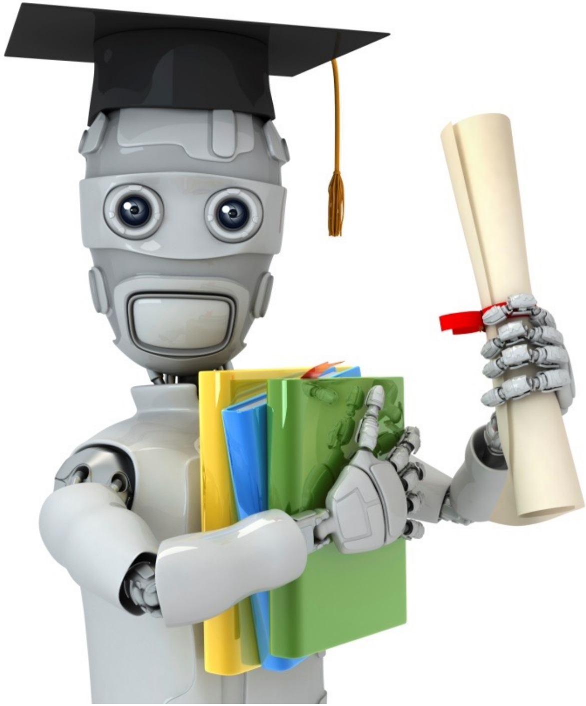
$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_0^{(i)}$$

$$\theta_j := \theta_j - \alpha \underbrace{\left[\frac{1}{m} \sum_{i=1}^{m} (\underline{h_\theta(x^{(i)})} - y^{(i)}) x_j^{(i)} - \frac{\lambda}{m} \theta_j \right]}_{\frac{\partial}{\partial \theta_j} \underline{J(\theta)}}$$

$$(j = \cancel{0}, \underbrace{1, 2, 3, \ldots, n}_{\theta_1 \ldots \theta_n})$$

$\}$

$$\underline{h_\theta(x)} = \frac{1}{1 + e^{-\theta^T x}}$$



Machine Learning

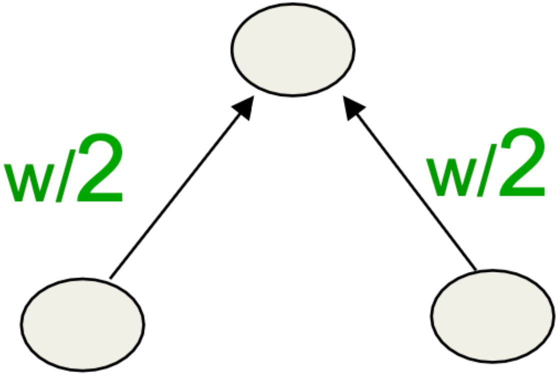
Regularization

Regularizer for sparsity

L2-Norm vs L1-Norm

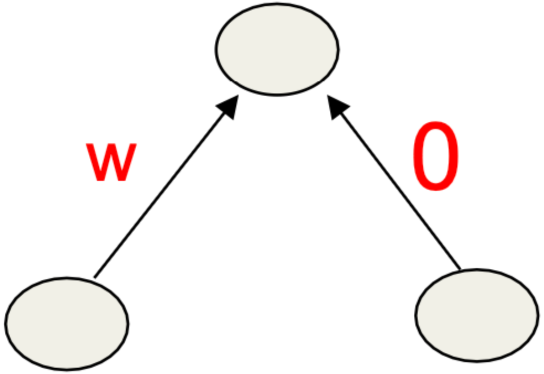
$$\frac{1}{2m}[\sum^{m}(\mathrm{h}_{\theta}(\mathrm{x})-y)^{2}+\lambda\sum_{\mathrm{j}=1}^{\mathrm{n}}\theta_{\mathrm{j}}^{2}]$$

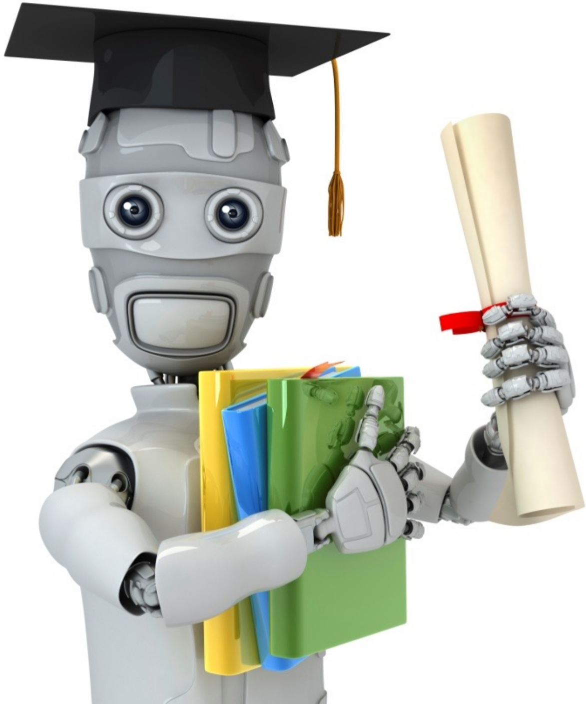
$$\theta_1^2+\theta_2^2\leq \mathrm{S}$$



$$\frac{1}{2m}[\sum^{m}(\mathrm{h}_{\theta}(\mathrm{x})-y)^{2}+\lambda\sum_{\mathrm{j}=1}^{\mathrm{n}}|\theta_{\mathrm{j}}|]$$

$$|\theta_1|+|\theta_2|\leq \mathrm{S}$$





Machine Learning

Validation

Validation

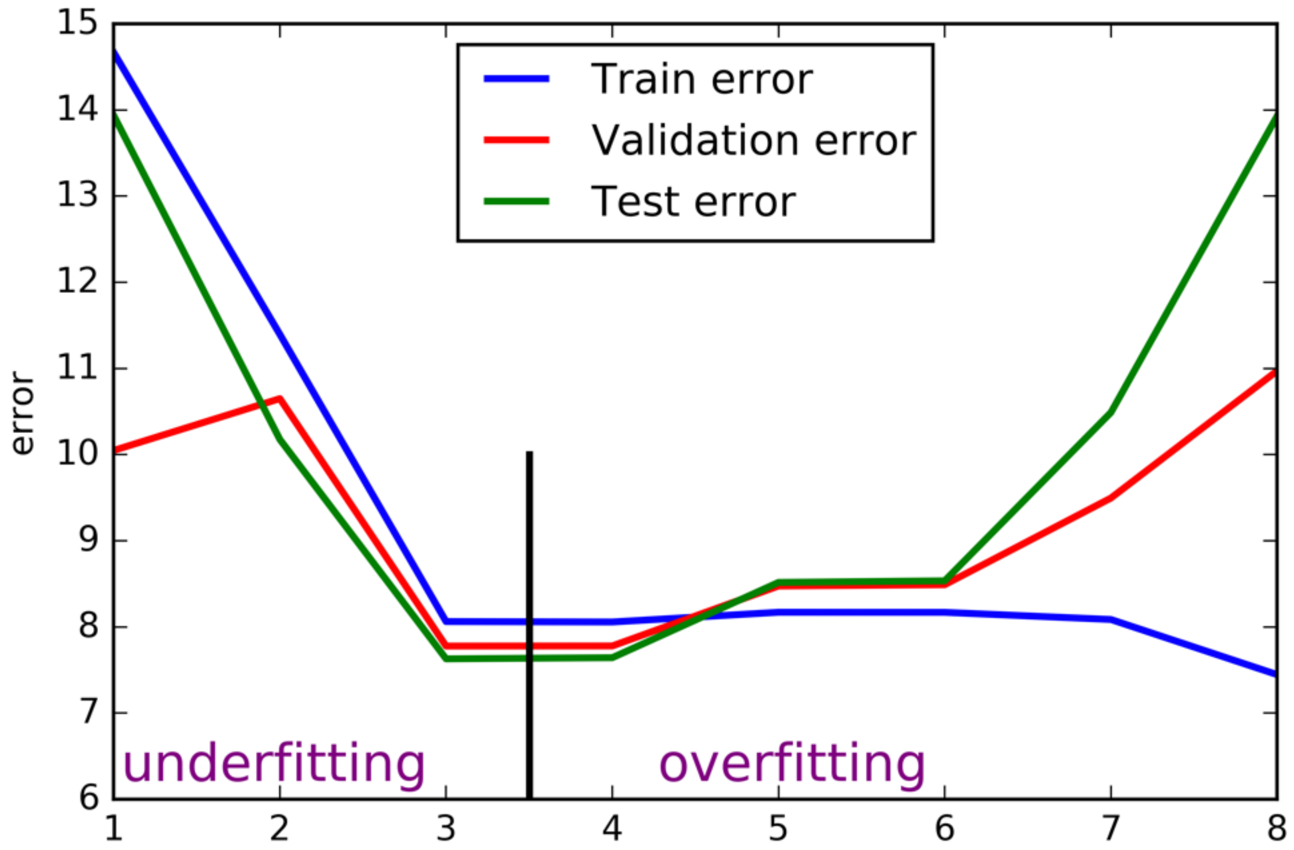
We often divide datasets into two sets:

- training data
- test data (not use in training phase)

How to know the quality of training model with unseen data?

- Extract a small set from training dataset → validation set
- Select the model that provides the small error in both training set and validation set

Validation

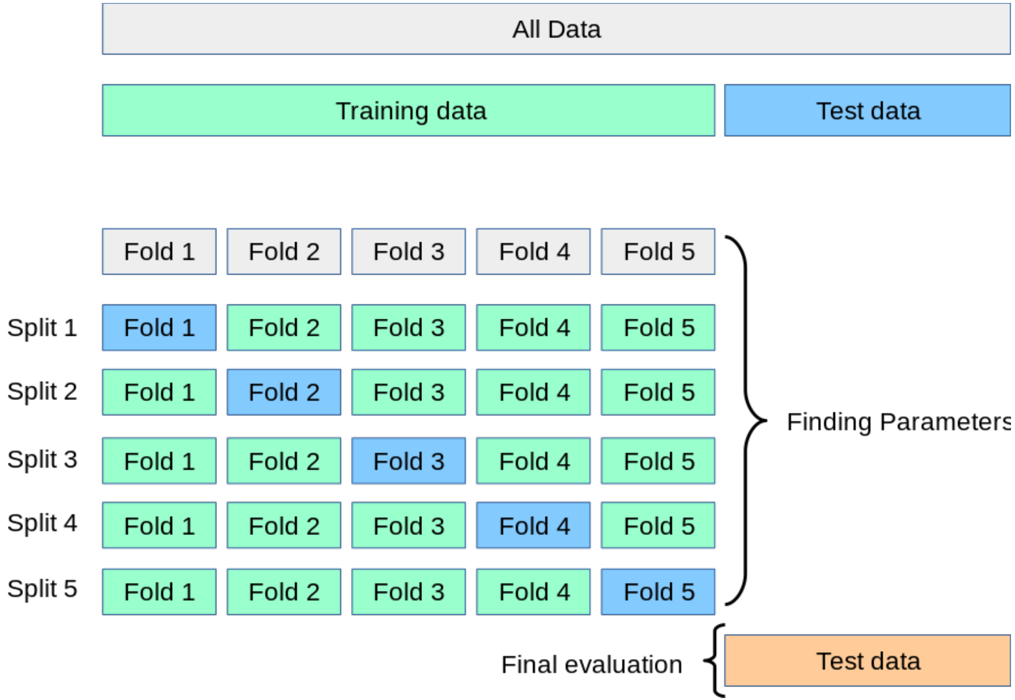


Train error
Validation error
Test error
error
underfitting
overfitting
15
14
13
12
11
10
9
8
7
6
1
2
3
4
5
6
7
8

Cross-Validation

If the training dataset is very small, taking too much data from it to form validation set will affect the performance → k-fold cross val.

- Split the training set into k sub-sets
- At each run, one of k sub-sets is treated as validation set
- The rest (k-1) sub-sets are treated as training set



Các bạn sử dụng tập dữ liệu của bài tập 5, mô hình dữ liệu x theo dạng bậc 6 và làm các công việc sau:

1. Chia dữ liệu ra thành 2 tập: training (70%) và validation (30%). Phải đảm bảo việc chia dữ liệu là ngẫu nhiên và tỷ lệ positive và negative cân bằng.
2. Viết chương trình cho phép học các tham số của mô hình phân loại phi tuyến trên có sử dụng regularization L2 và L1.
3. Tính J ở mỗi vòng lặp cho cả hai tập, chọn điểm dừng phù hợp.
4. Thay đổi lambda và tính J cho mỗi lambda tương ứng cho cả hai tập. Vẽ biểu đồ quan hệ giữa J và lambda từ đó lựa chọn lambda phù hợp.