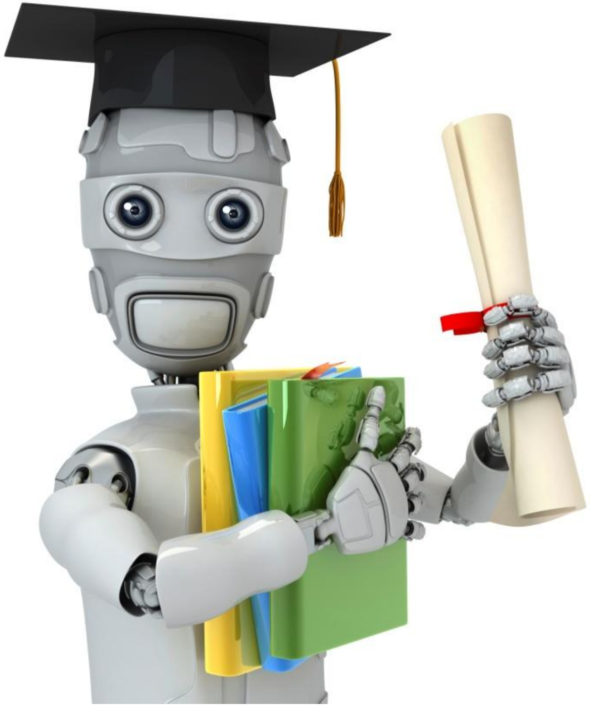


Machine Learning

# Logistic Regression

## Classification

## Classification

Email: Spam / Not Spam?

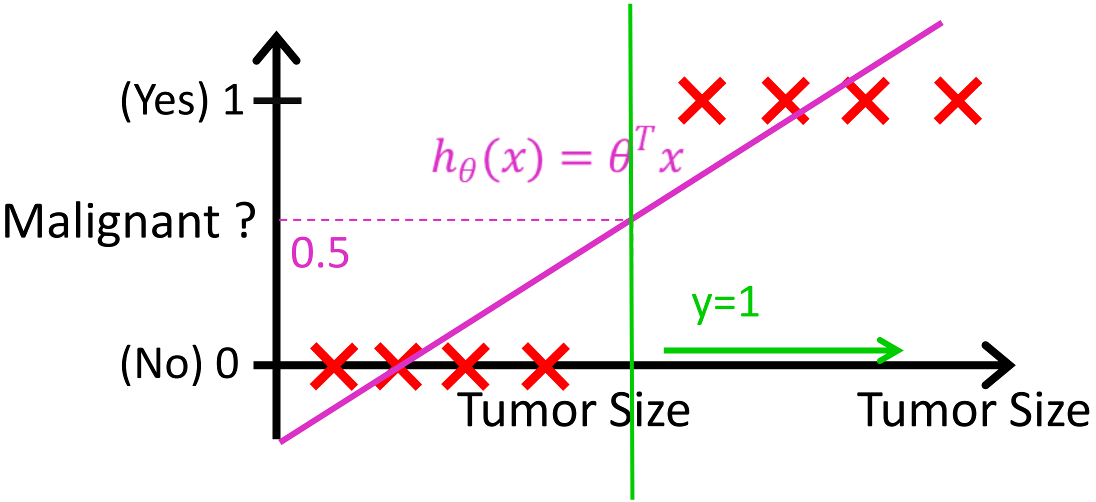
Online Transactions: Fraudulent (Yes / No)?

Tumor: Malignant / Benign ?

$$y \in \{0, 1\}$$

0: “Negative Class” (e.g., benign tumor)

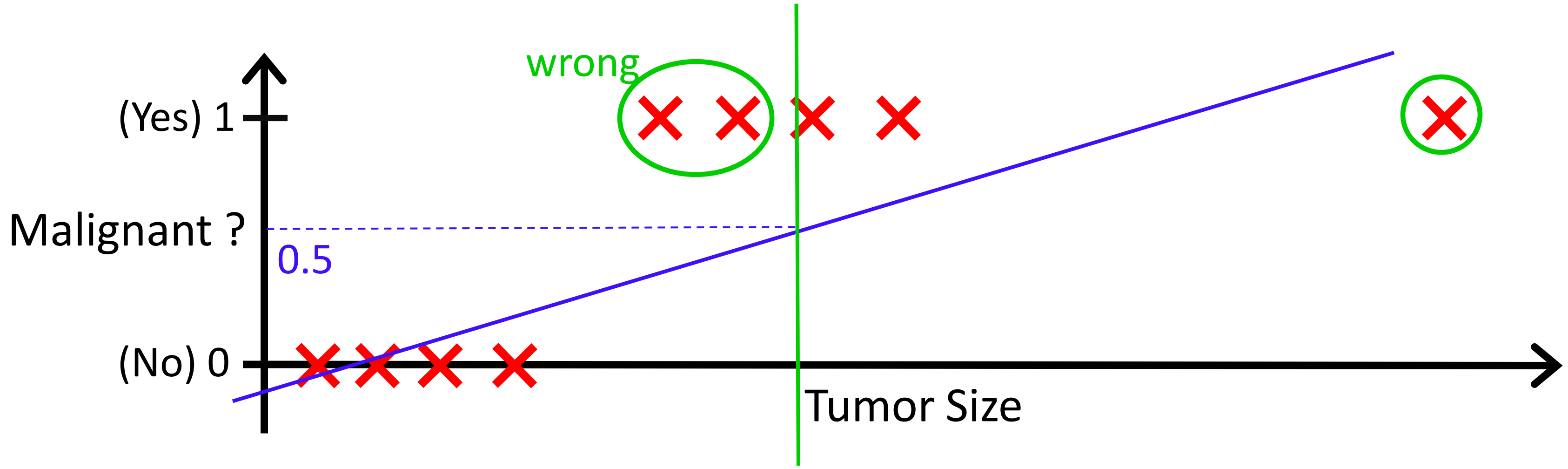
1: “Positive Class” (e.g., malignant tumor)




Threshold classifier output $h_\theta(x)$ at 0.5:

If $h_\theta(x) \geq 0.5$, predict "y = 1"

If $h_\theta(x) < 0.5$, predict "y = 0"




Threshold classifier output $h_\theta(x)$ at 0.5:

If $h_\theta(x) \geq 0.5$, predict "y = 1"

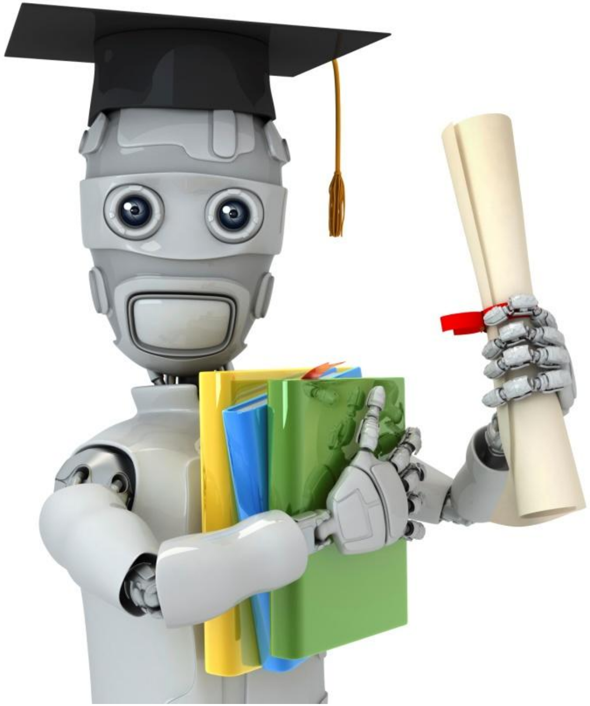
If $h_\theta(x) < 0.5$, predict "y = 0"

Classification: y = 0 or 1

$h_\theta(x)$ can be > 1 or < 0

Logistic Regression: $0 \leq h_\theta(x) \leq 1$

(a classification algorithm)



Machine Learning

# Logistic Regression

## Hypothesis Representation

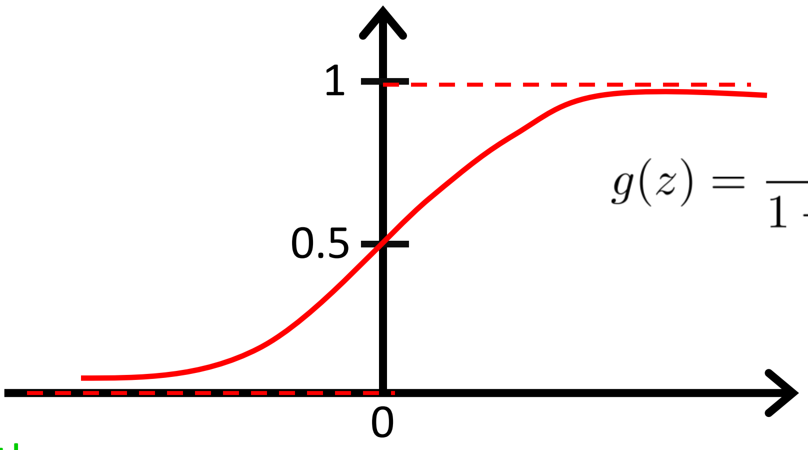
## Logistic Regression Model

Want $0 \leq h_\theta(x) \leq 1$

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

$$h_\theta(x) = g(\ \theta^T x\ )$$

$$g(z) = \frac{1}{1+e^{-z}}$$




$$g(z) = \frac{1}{1+e^{-z}}$$

Sigmoid function

Logistic function

Mean the same thing

# Interpretation of Hypothesis Output

$h_\theta(x)$ = estimated probability that y = 1 on input x

Example: If $x = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 1 \\ \text{tumorSize} \end{bmatrix}$

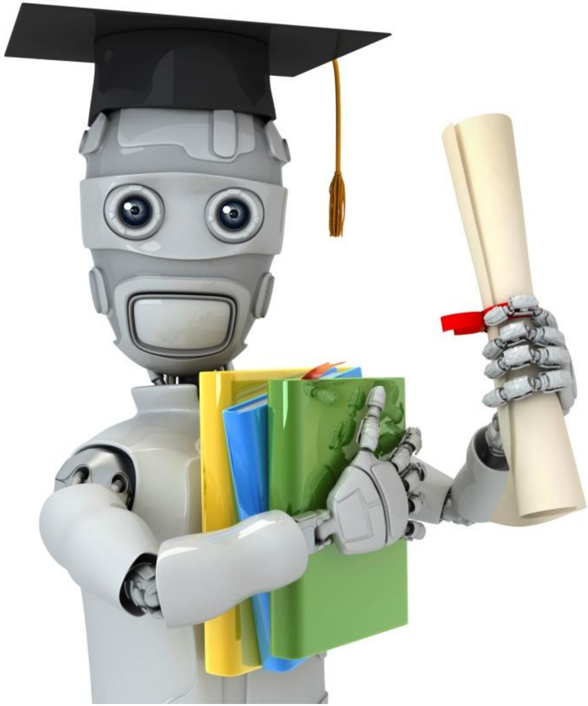
$$h_\theta(x) = 0.7$$

Tell patient that 70% chance of tumor being malignant

$$h_\theta(x) = P(y = 1|x;\theta)$$

"probability that y = 1, given x, parameterized by $\theta$"

$$P(y = 0|x;\theta) + P(y = 1|x;\theta) = 1$$
$$P(y = 0|x;\theta) = 1 - P(y = 1|x;\theta)$$



Machine Learning

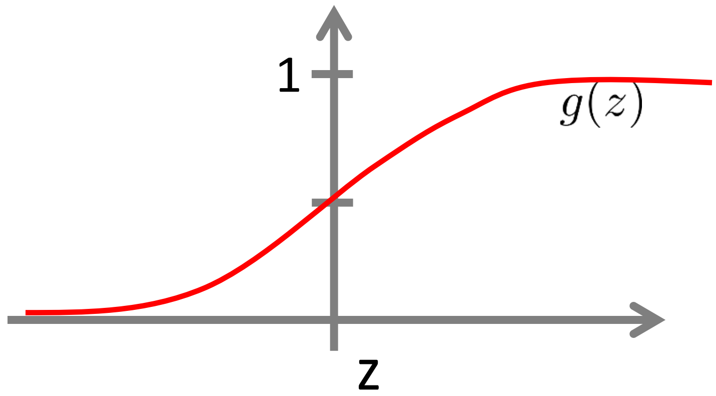
# Logistic Regression

## Decision boundary

# Logistic regression

$$h_\theta(x) = g(\theta^T x)$$

$$g(z) = \frac{1}{1+e^{-z}}$$




Suppose predict "$y = 1$" if $h_\theta(x) \geq 0.5$

$g(z) \geq 0.5$ When $z \geq 0$

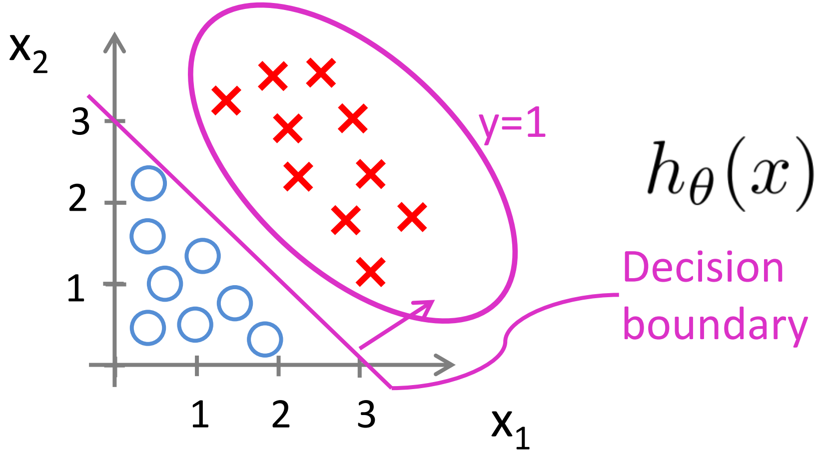
So $h_\theta(x) = g(\theta^T x) \geq 0.5$ When $\theta^T x \geq 0$

predict "$y = 0$" if $h_\theta(x) < 0.5$

$g(z) \leq 0.5$ When $z<0$

So $h_\theta(x) = g(\theta^T x) < 0.5$ When $\theta^T x < 0$

# Decision Boundary

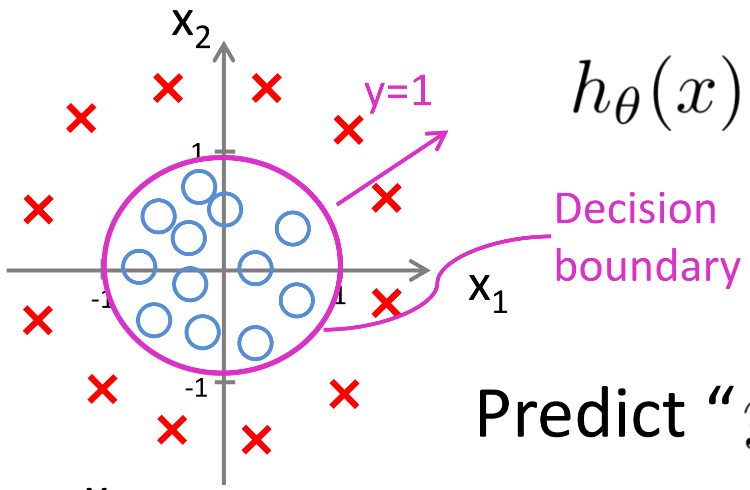



$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

$\theta_0 = -3$, $\theta_1 = 1$, $\theta_2 = 1$

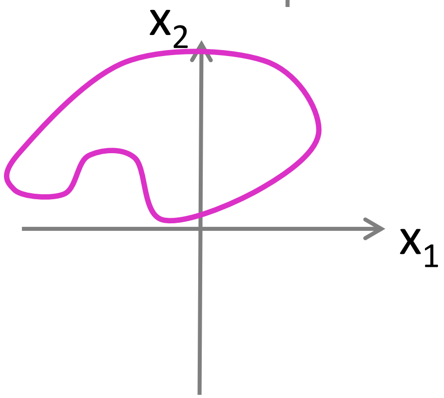
Predict "$y = 1$" if $-3 + x_1 + x_2 \geq 0$

# Non-linear decision boundaries

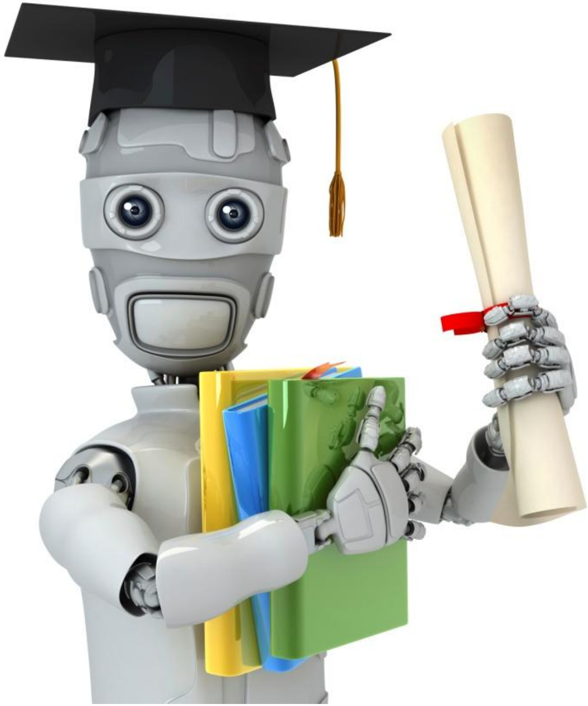



$$\begin{array}{l} \qquad\qquad\qquad\quad -1 \qquad\quad 0 \qquad\qquad\quad 0 \\ h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 \\ \qquad\qquad\qquad\qquad + \theta_3 x_1^2 + \theta_4 x_2^2) \\ \qquad\qquad\qquad\qquad\quad 1 \qquad\qquad\quad 1 \end{array}$$

Predict "$y = 1$" if $-1 + x_1^2 + x_2^2 \geq 0$




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



# Logistic Regression

## Cost function

Machine Learning

Training set: $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \cdots, (x^{(m)}, y^{(m)})\}$

m examples $\qquad x \in \begin{bmatrix} x_0 \\ x_1 \\ \cdots \\ x_n \end{bmatrix} \qquad x_0 = 1, y \in \{0, 1\}$

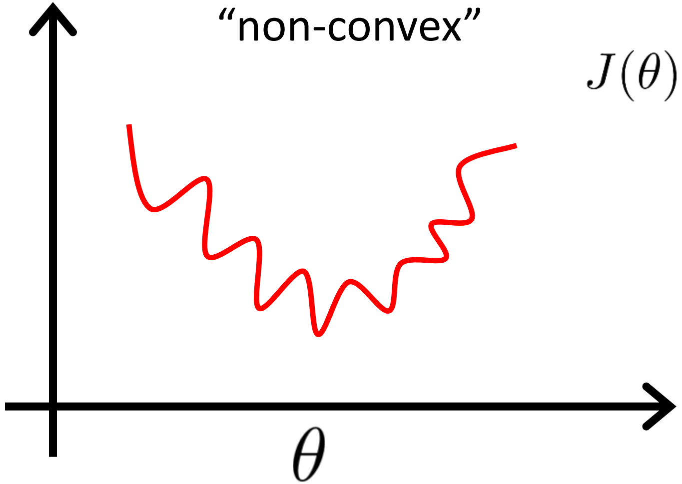
$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

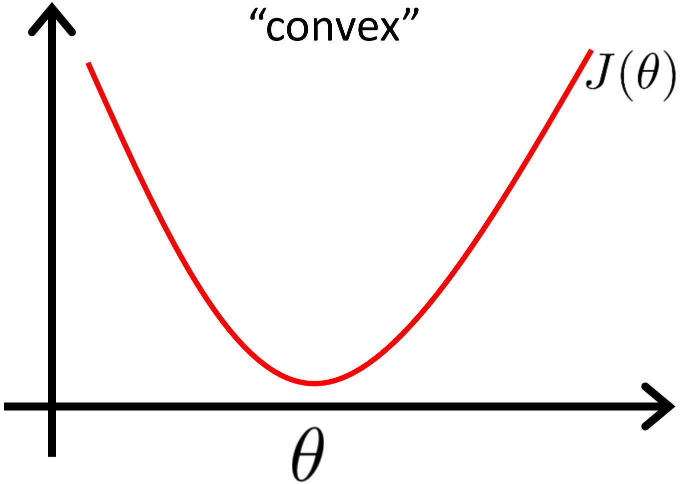
How to choose parameters $\theta$ ?

## Cost function

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

$$\mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)}) = \frac{1}{2} \left( h_\theta(x^{(i)}) - y^{(i)} \right)^2$$

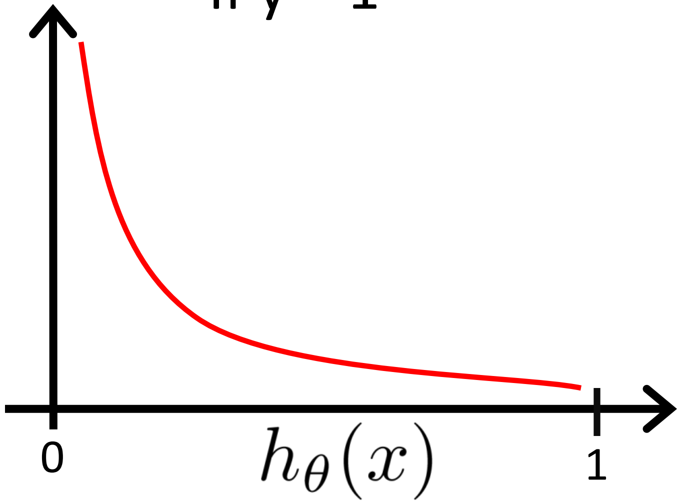


# Logistic regression cost function

$$\mathrm{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

If y = 1




$\mathrm{Cost} = 0$ if $y = 1, h_\theta(x) = 1$

But as $h_\theta(x) \to 0$

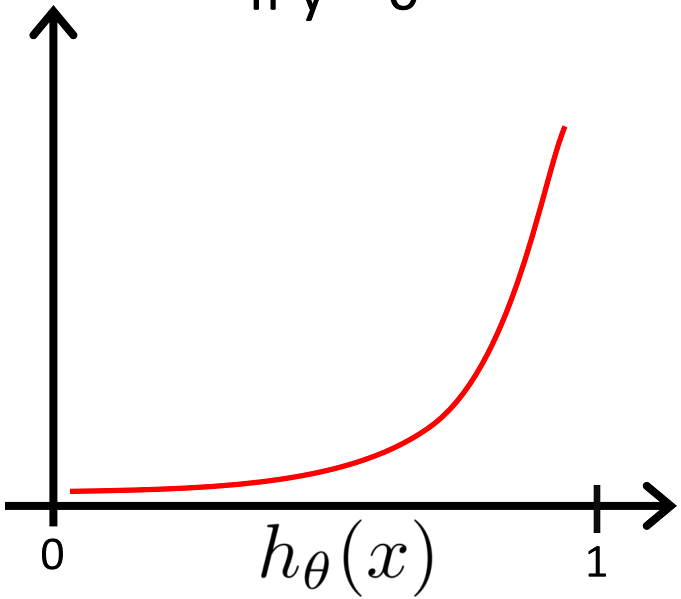
$Cost \to \infty$

Captures intuition that if $h_\theta(x) = 0$, (predict $P(y = 1|x;\theta) = 0$), but $y = 1$, we'll penalize learning algorithm by a very large cost.

# Logistic regression cost function

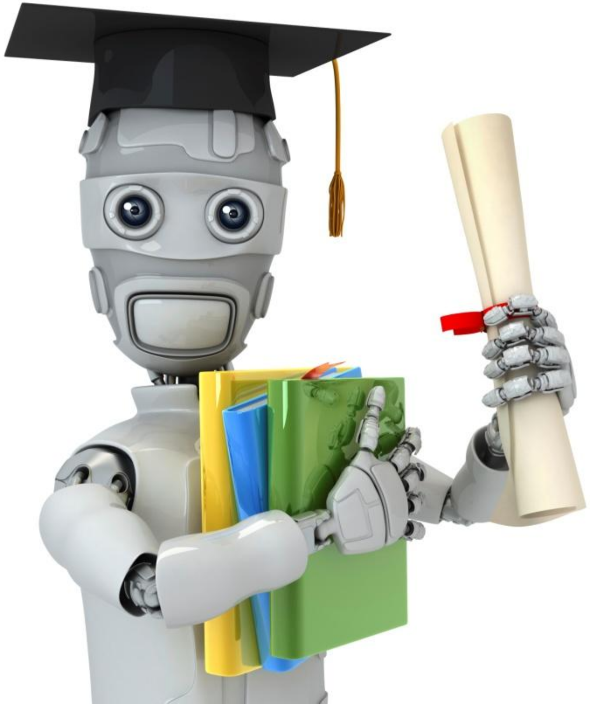
$$\mathrm{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

If y = 0




If y=0 and theta->0, then cost->0.
If y=0 and theta->1, then cost->infinity.

This is the motivation of using a cost function in the form.



# Logistic Regression

## Simplified cost function and gradient descent

Machine Learning

# Logistic regression cost function

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)})$$

$$\mathrm{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

Note: $y = 0$ or $1$ always

$$Cost(h_\theta(x), y) = -y\log(h_\theta(x)) - (1-y)\log(1-h_\theta(x))$$

if y=1 $\quad Cost(h_\theta(x), y) = -\log(h_\theta(x))$

if y=0 $\quad Cost(h_\theta(x), y) = -\log(1-h_\theta(x))$

# Logistic regression cost function

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} \mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)})$$

$$= -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log (1 - h_\theta(x^{(i)}))]$$

To fit parameters $\theta$:

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

To make a prediction given new $x$:

Output $h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$ $\rightarrow$ $p(y = 1|x;\theta)$

Andrew Ng

# Gradient Descent

$$J(\theta) = -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1-y^{(i)}) \log (1-h_\theta(x^{(i)}))]$$

Want $\min_\theta J(\theta)$:

Repeat $\{$

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

(simultaneously update all $\theta_j$)

$\}$

# Gradient Descent

$$J(\theta) = -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log (1 - h_\theta(x^{(i)}))]$$

Want $\min_\theta J(\theta)$:

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

Repeat $\{$

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

(simultaneously update all $\theta_j$)

$\}$

Algorithm looks identical to linear regression!

# Optimization algorithm

Cost function $J(\theta)$. Want $\min_\theta J(\theta)$.

Given $\theta$, we have code that can compute

- $J(\theta)$
- $\frac{\partial}{\partial\theta_j} J(\theta)$ (for $j = 0, 1, \ldots, n$)

Gradient descent:

Repeat $\{$

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial\theta_j} J(\theta)$$

$\}$

## Optimization algorithm

Given $\theta$, we have code that can compute

- $J(\theta)$
- $\frac{\partial}{\partial\theta_j} J(\theta)$ (for $j = 0, 1, \ldots, n$)

Optimization algorithms:

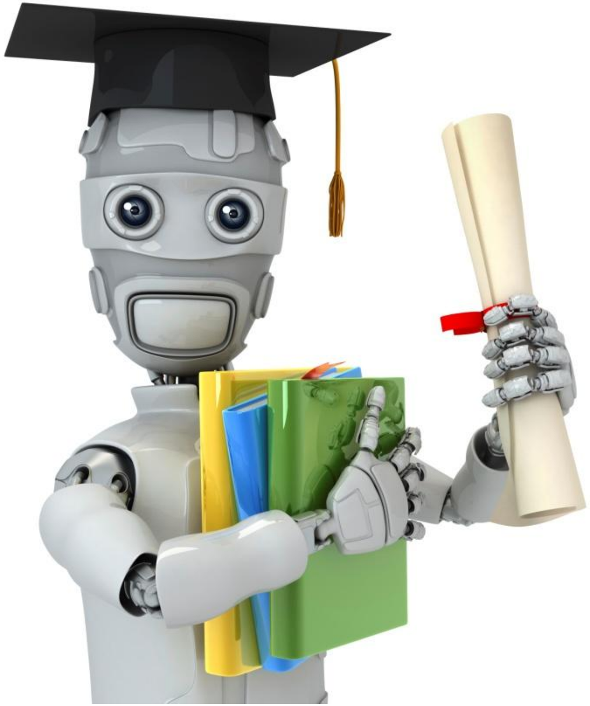
- Gradient descent
- Conjugate gradient
- BFGS
- L-BFGS
- Coordinate descent

Advantages:

- No need to manually pick $\alpha$
- Often faster than gradient descent.

Disadvantages:

- More complex



# Logistic Regression

## Multi-class classification: One-vs-all

Machine Learning

# Multiclass classification

Email foldering/tagging: Work, Friends, Family, Hobby

$y=1$ $y=2$ $y=3$ $y=4$

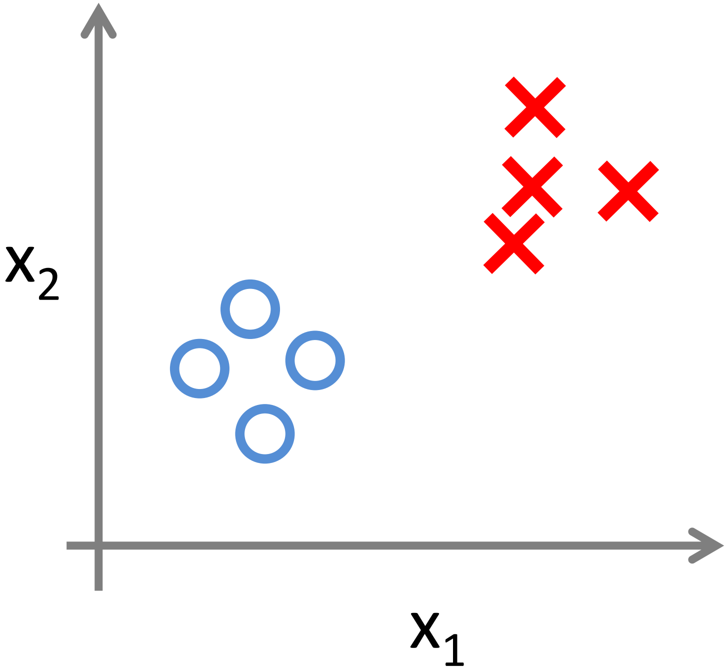
Medical diagrams: Not ill, Cold, Flu

$y=1$ $y=2$ $y=3$

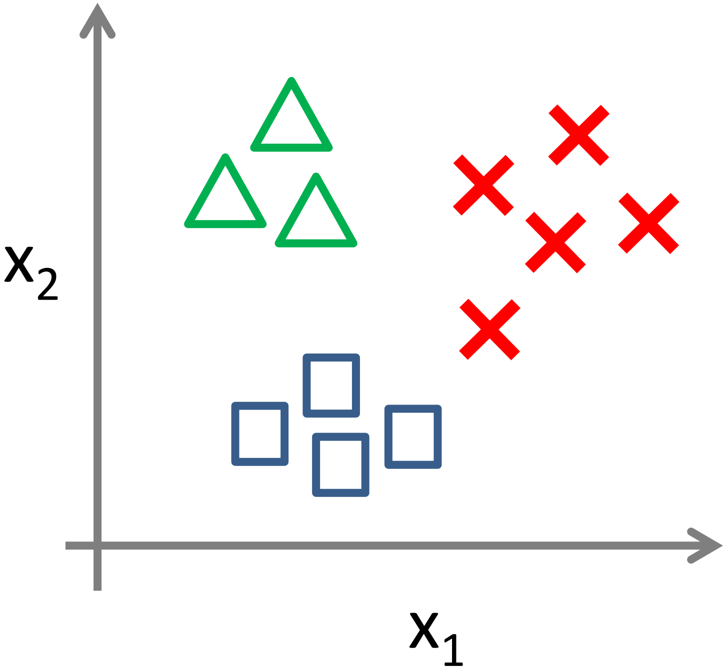
Weather: Sunny, Cloudy, Rain, Snow

$y=1$ $y=2$ $y=3$ $y=4$

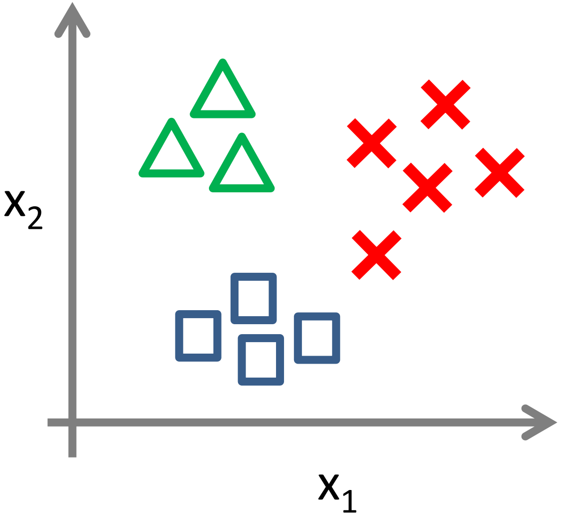
Binary classification:

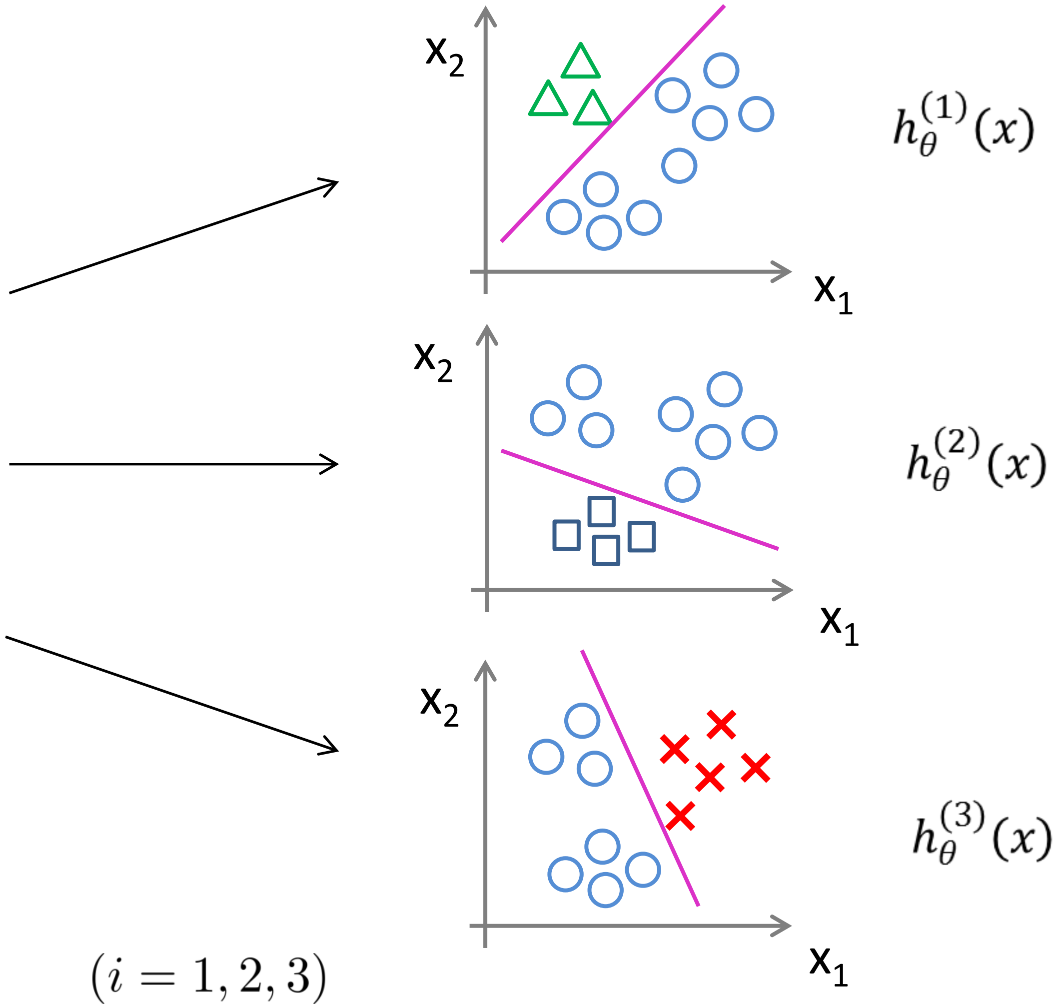



Multi-class classification:



# One-vs-all (one-vs-rest):




Class 1: △

Class 2: □

Class 3: ✕

$$h_\theta^{(i)}(x) = P(y = i|x;\theta) \qquad (i = 1, 2, 3)$$

# One-vs-all

Train a logistic regression classifier $h_\theta^{(i)}(x)$ for each class $i$ to predict the probability that $y = i$.

On a new input $x$, to make a prediction, pick the class $i$ that maximizes

$$\max_i h_\theta^{(i)}(x)$$