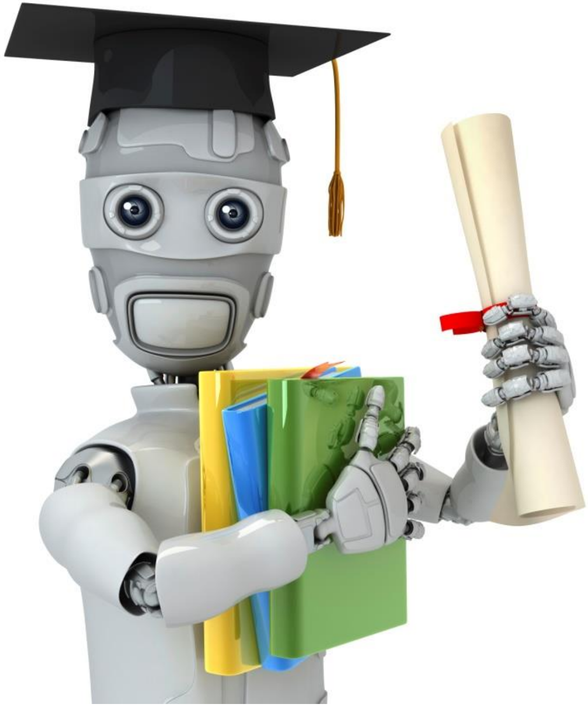


Machine Learning

Logistic Regression

Classification

Classification

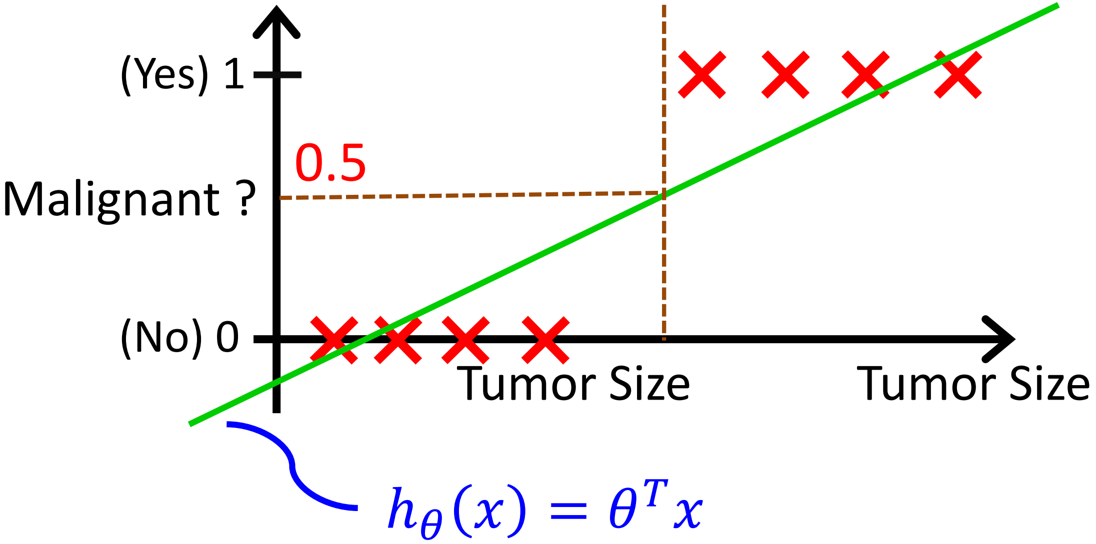
Email: Spam / Not Spam?
Online Transactions: Fraudulent (Yes / No)?
Tumor: Malignant / Benign ?

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

0: "Negative Class" (e.g., benign tumor)
1: "Positive Class" (e.g., malignant tumor)

Binary classification: classification problem with just two categories.

Multi-class classification: classification problem with more than two categories.

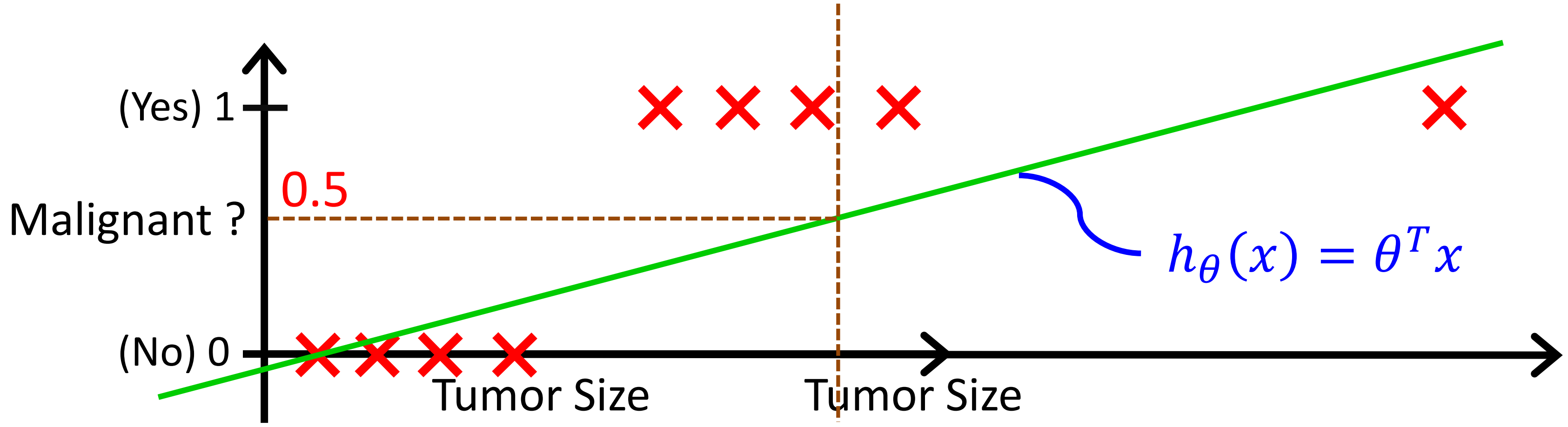


Threshold classifier output $h_\theta(x)$ at 0.5:

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

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

Andrew Ng



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

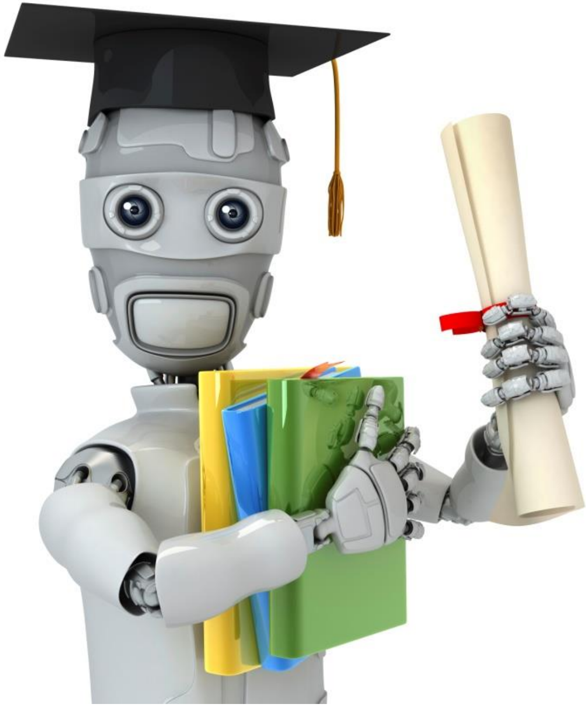
If $h_\theta(x) \geq 0.5$, predict "y = 1"

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

Classification: y = 0 or 1

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

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



Machine Learning

Logistic Regression

Hypothesis Representation

Logistic Regression Model

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

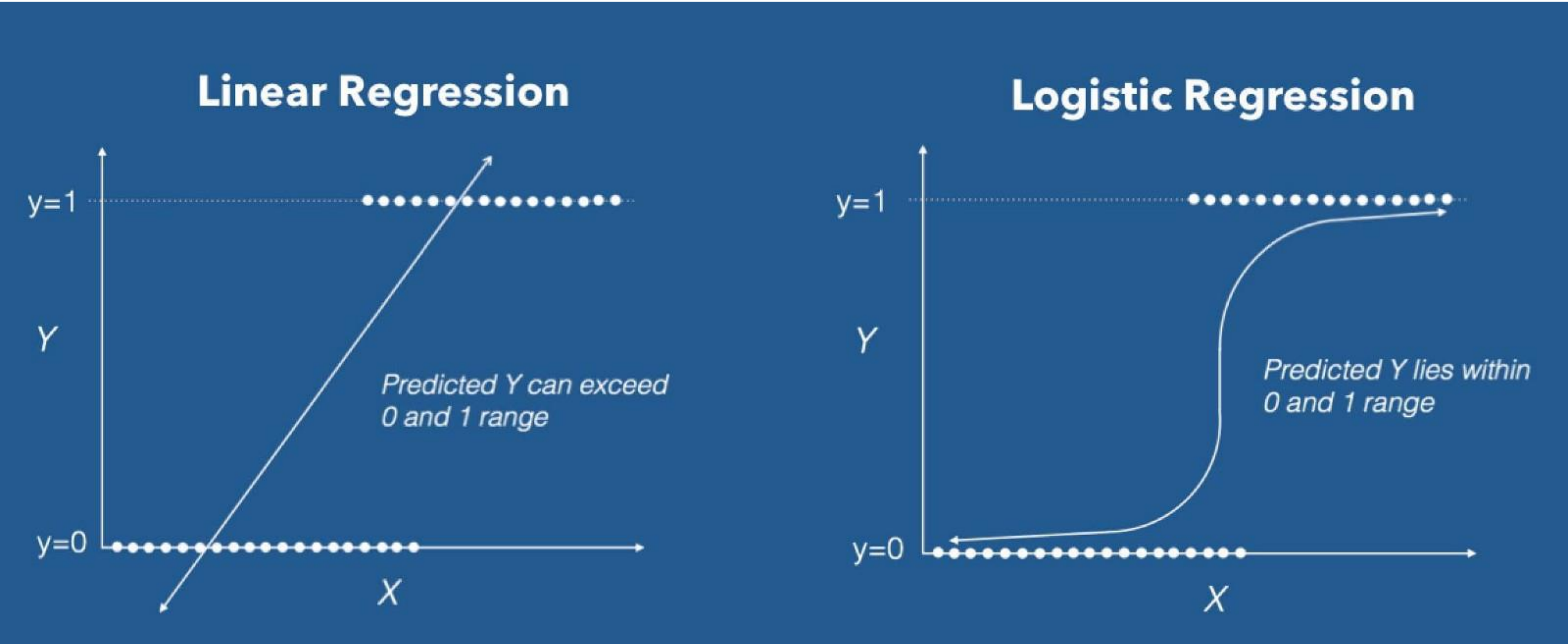
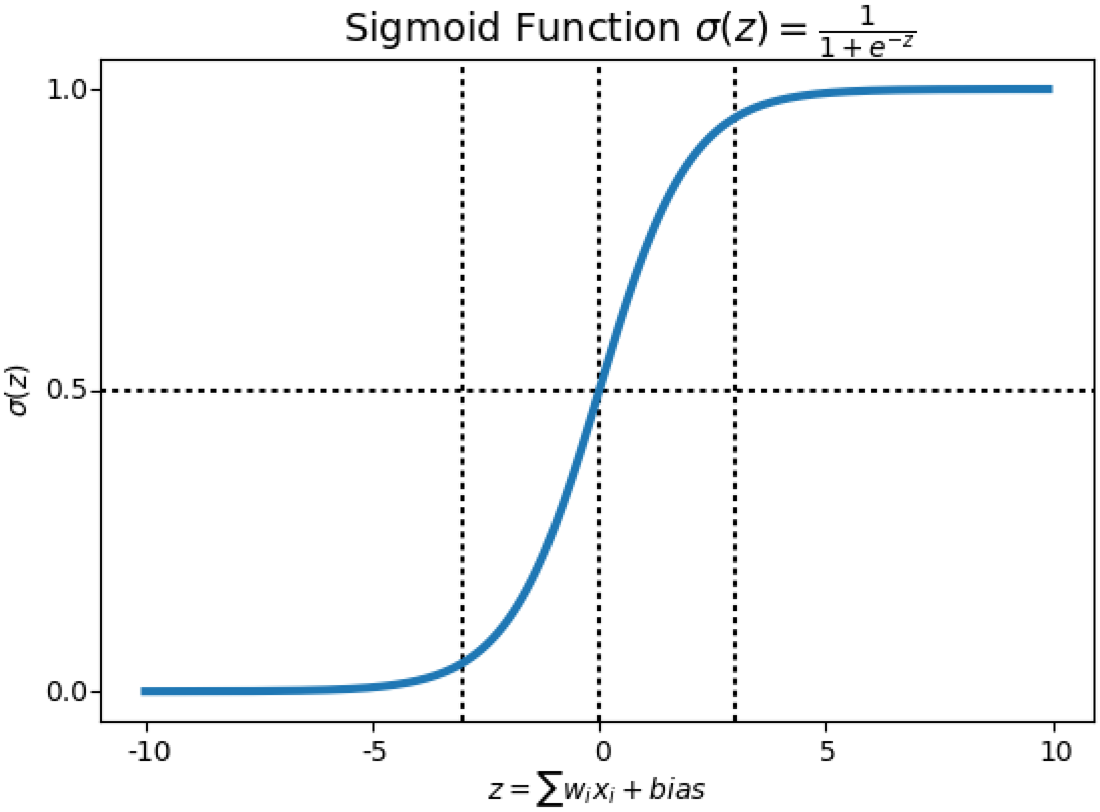


Image: Data Camp

Sigmoid function

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



Logistic Regression Model

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

Linear regression hypothesis: $h_\theta(x) = \theta^T x$

Logistic regression hypothesis: $h_\theta(x) = g(\theta^T x)$

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

Sigmoid function

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

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

Formally,

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

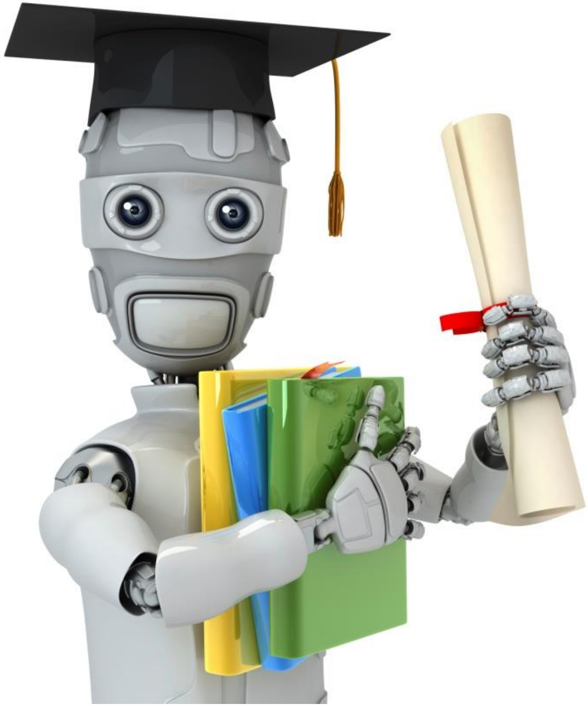
"probability that $y = 1$, given x, parameterized by θ"

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

Andrew Ng

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

The value of hypothesis is interpreted as the probability that the input x belongs to class $y = 1$. i.e. probability that $y = 1$, given x, parametrized by θ.

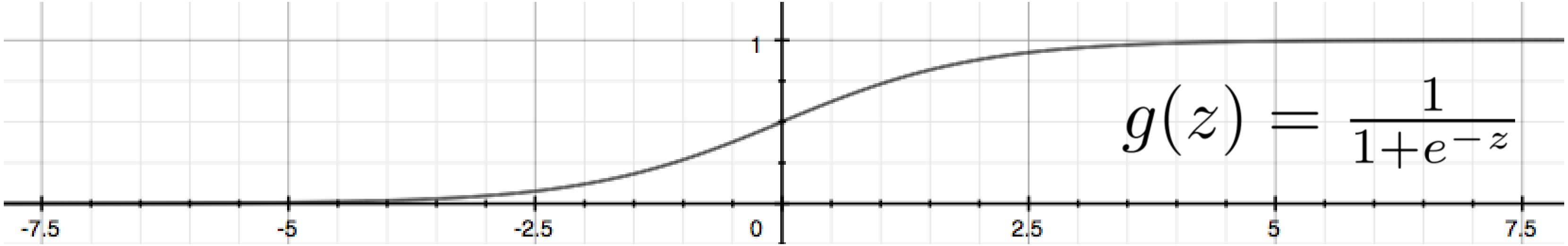


Logistic Regression

Decision boundary

Machine Learning

Logistic regression $\quad h_\theta(x) = g(\theta^T x)$



Suppose:

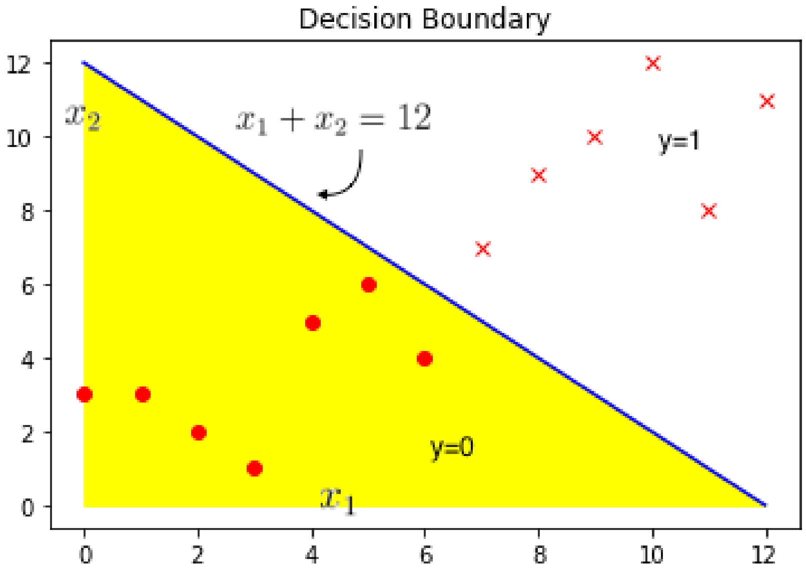
Predict "$y = 1$" if $h_\theta(x) \geq 0.5$ $\qquad$ Predict "$y = 1$" if $\theta^T x \geq 0$

Predict "$y = 0$" if $h_\theta(x) < 0.5$ $\qquad$ Predict "$y = 0$" if $\theta^T x < 0$

$g(z) \geq 0.5$, if $z \geq 0$

$g(z) < 0.5$, if $z < 0$

Decision Boundary



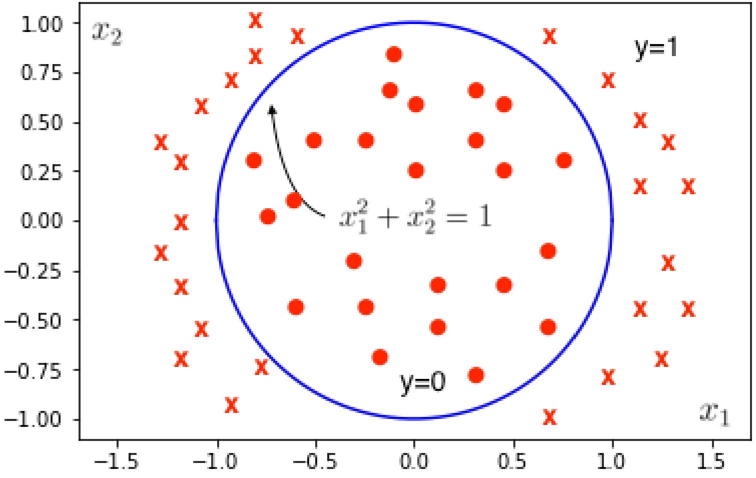
$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

$$\theta = [-12, 1, 1]^T$$

Predict $y = 1$, if $-12 + x_1 + x_2 \geq 0$ or $x_1 + x_2 \geq 12$

Predict $y = 0$, if $-12 + x_1 + x_2 < 0$ or $x_1 + x_2 < 12$

Non-linear Decision Boundary



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

$$\theta = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

As the order of features is increased, more and more complex decision boundaries can be achieved by logistic regression.

Substituting (12) in (11),

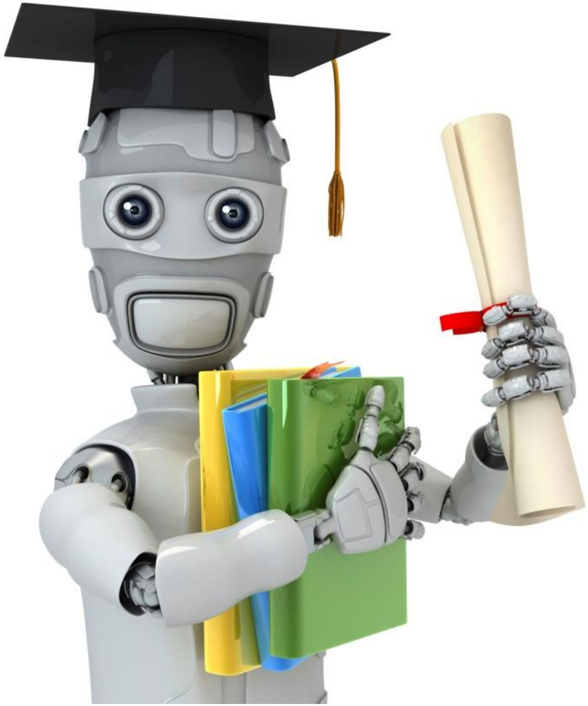
$$\theta^T x = -1 + x_1^2 + x_2^2$$

So, from (7), the decision boundary is given by,

$$\begin{aligned} -1 + x_1^2 + x_2^2 &= 0 \\ x_1^2 + x_2^2 &= 1 \end{aligned}$$

Which the equation of a circle at origin with radius 0, as can be seen in the plot above. And, using the θ from (12) and hypothesis from (11) , (7) can be written as,

$$\begin{aligned} &\text{predict } y = 1, \text{ if } -1 + x_1^2 + x_2^2 \geq 0 \text{ or } x_1^2 + x_2^2 \geq 1 \\ &\text{predict } y = 0, \text{ if } -1 + x_1^2 + x_2^2 < 0 \text{ or } x_1^2 + x_2^2 < 1 \end{aligned} \tag{12}$$



Machine Learning

Logistic Regression

Cost function

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

m examples

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

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

How to choose parameters θ ?

Cost function

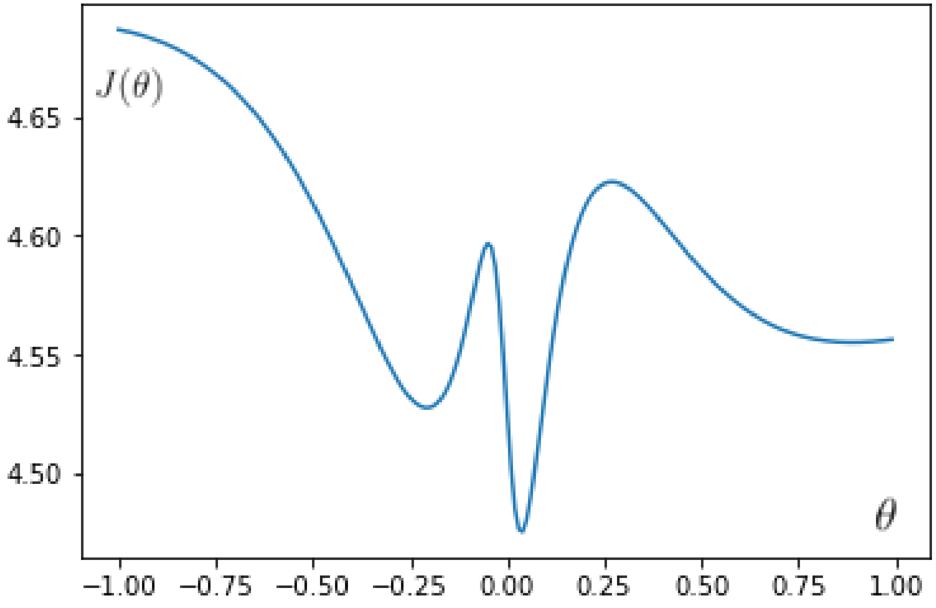
Linear regression:
$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{2} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$$

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

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

Cost function

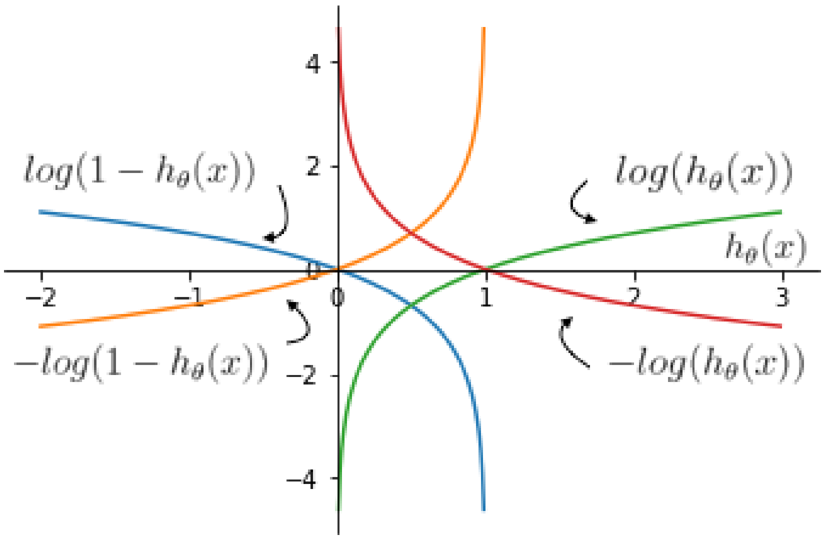
Linear regression: $$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{2} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$



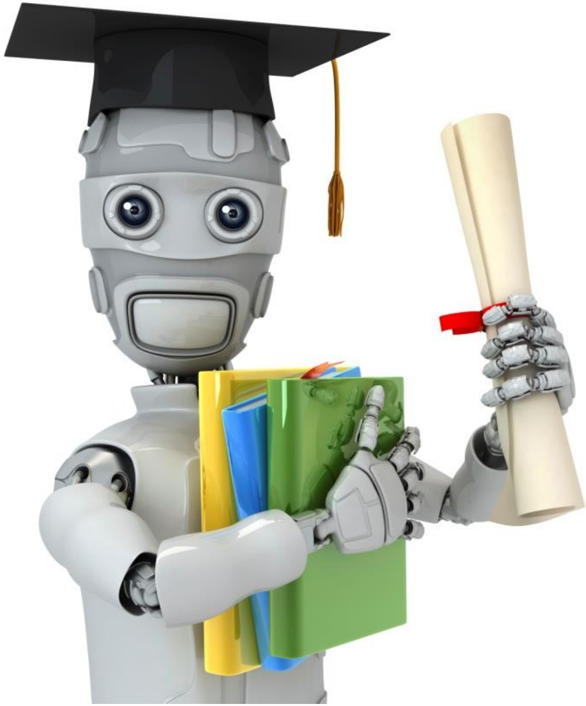
Non convex cost

Logistic regression cost function

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



- If $y = 1$ and
  - $h_\theta(x) = 1$, then $Cost = 0$
  - $h_\theta(x) \to 0$, then $Cost \to \infty$
- If $y = 0$ and
  - $h_\theta(x) = 0$, then $Cost = 0$
  - $h_\theta(x) \to 1$, then $Cost \to \infty$



Machine Learning

Logistic Regression

Simplified cost function and gradient descent

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

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

Logistic regression cost function

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

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

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

Logistic regression cost function

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

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

To fit parameters θ :

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

To make a prediction given new x:

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

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 $\{$

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

$\}$ (simultaneously update all θ_j)

$$\frac{1}{m}\sum_{i=1}^{m}\left(h(x^{(i)}) - y^{(i)}\right) x_j^{(i)}$$

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

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

$$\frac{d}{dz}log(z) = \frac{1}{z}$$

$$\begin{aligned}\frac{d}{dz}h(z) &= \frac{d}{dz}\frac{1}{1+e^{-z}} \\ &= \frac{e^{-z}}{(1+e^{-z})^2} \\ &= \frac{1+e^{-z}-1}{(1+e^{-z})^2} \\ &= \frac{1}{1+e^{-z}} - \frac{1}{(1+e^{-z})^2} \\ &= h(z)(1-h(z))\end{aligned}$$

$$\begin{aligned}\frac{\partial}{\partial\theta_j}J(\theta) &= -\frac{1}{m}\sum_{i=1}^{m} y^{(i)}\,\frac{x_j^{(i)}}{h(x^{(i)})}h(x^{(i)})(1-h(x^{(i)})) + (1-y^{(i)})\frac{x_j^{(i)}}{1-h(x^{(i)})}(-h(x^{(i)})(1-h(x^{(i)}))) \\ &= -\frac{1}{m}\sum_{i=1}^{m}(y^{(i)}\,(1-h(x^{(i)})) - (1-y^{(i)})h(x^{(i)}))x_j^{(i)} \\ &= -\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)} - y^{(i)}h(x^{(i)}) - h(x^{(i)}) + y^{(i)}h(x^{(i)})\right)x_j^{(i)} \\ &= \frac{1}{m}\sum_{i=1}^{m}\left(h(x^{(i)}) - y^{(i)}\right)x_j^{(i)}\end{aligned}$$

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 $\{$

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

(simultaneously update all θ_j)

$\}$

Algorithm looks identical to linear regression!

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

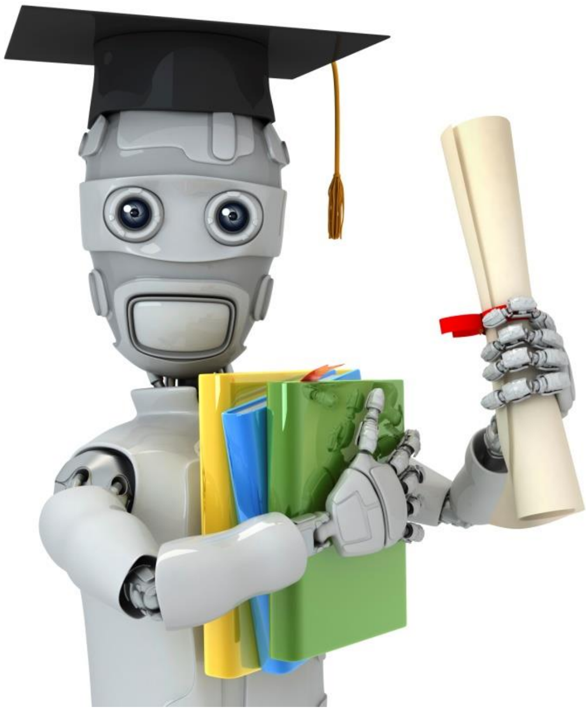
Repeat $\{$

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

$\}$

vectorized implementation

$$\theta := \theta - \alpha \frac{1}{m} X^T (g(X\theta) - y)$$



Machine Learning

Logistic Regression

Multi-class classification: One-vs-all

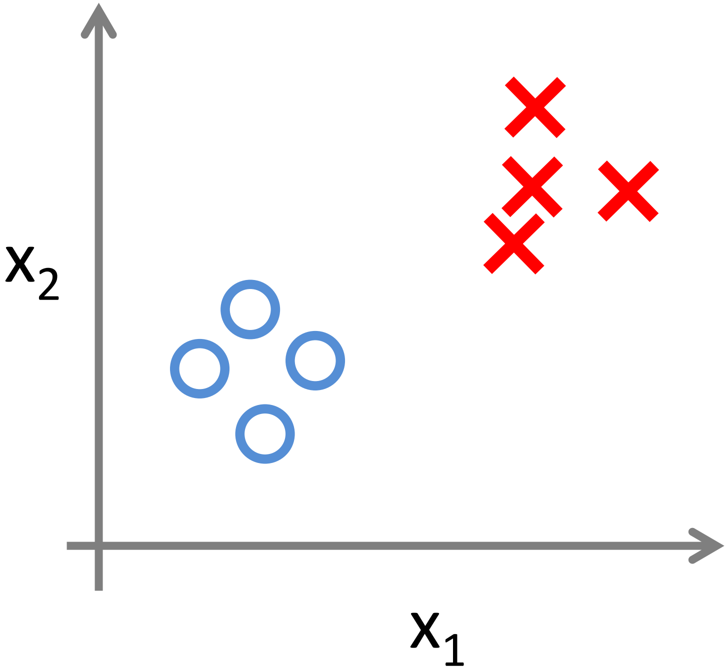
Multiclass classification

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

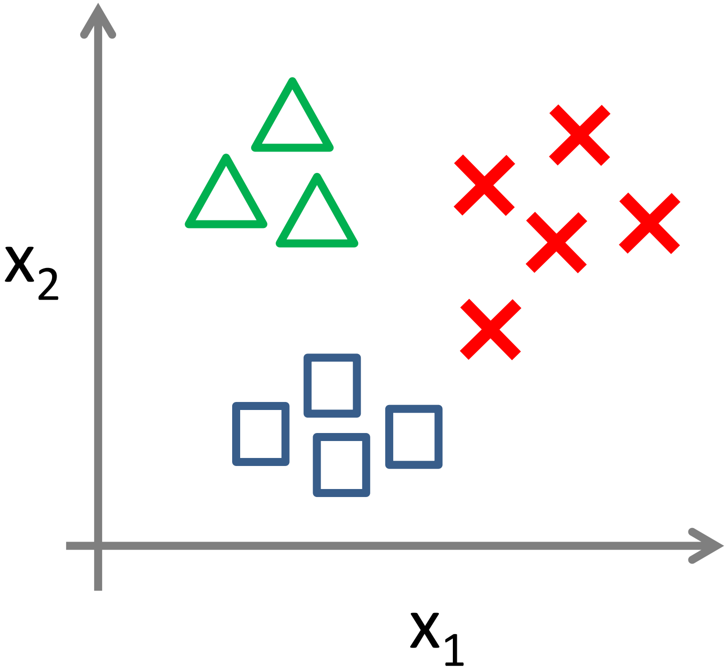
Medical diagrams: Not ill, Cold, Flu

Weather: Sunny, Cloudy, Rain, Snow

Binary classification:

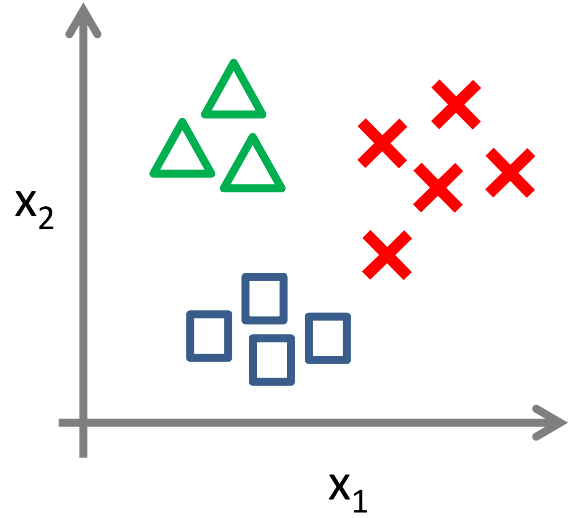


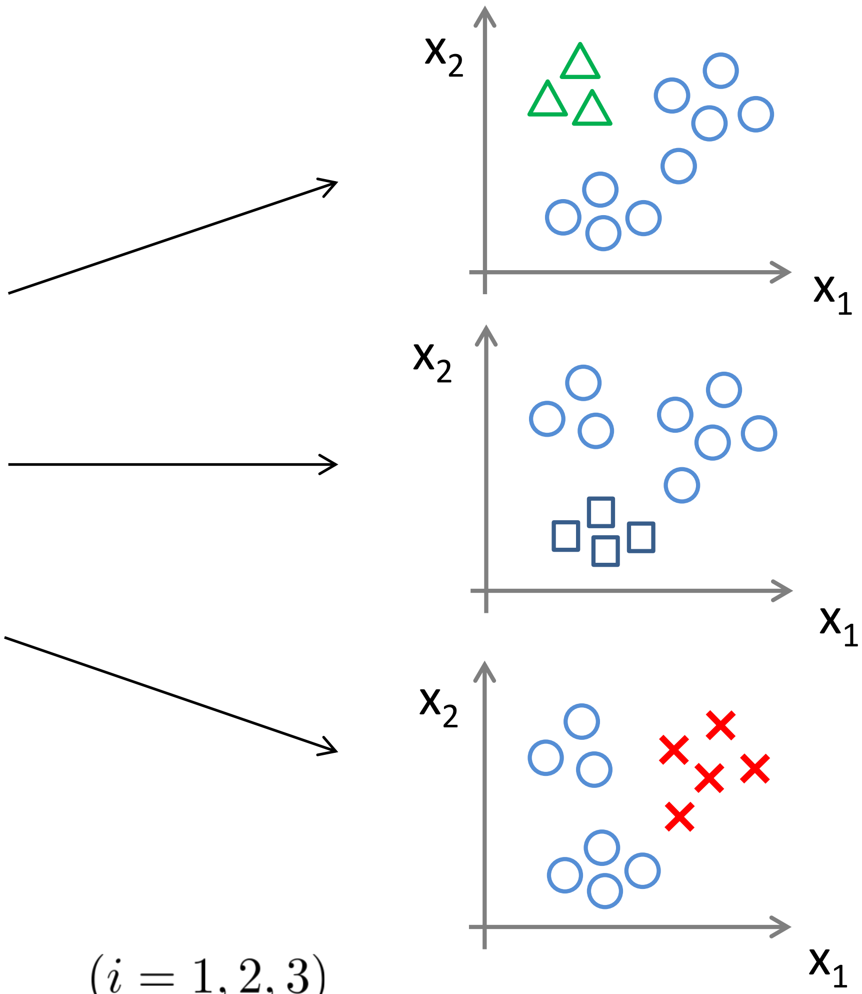
Multi-class classification:



Andrew Ng

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

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