Logistic Regression
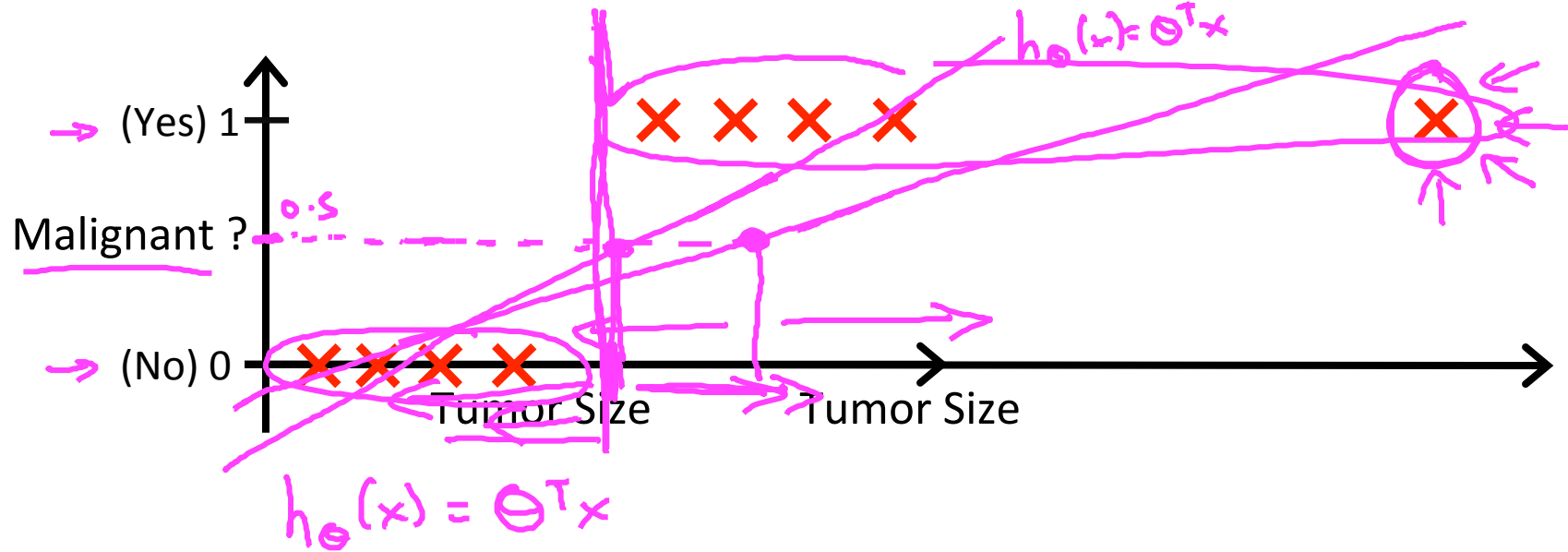
Classification

Machine Learning

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)

$$y \in \{0, 1, 2, 3\}$$

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

Classification: $y = 0$ or 1

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

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

Classification

Logistic Regression

Hypothesis Representation

Machine Learning

Logistic Regression Model

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

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

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

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

$\theta^T x$

Sigmoid function

Logistic function
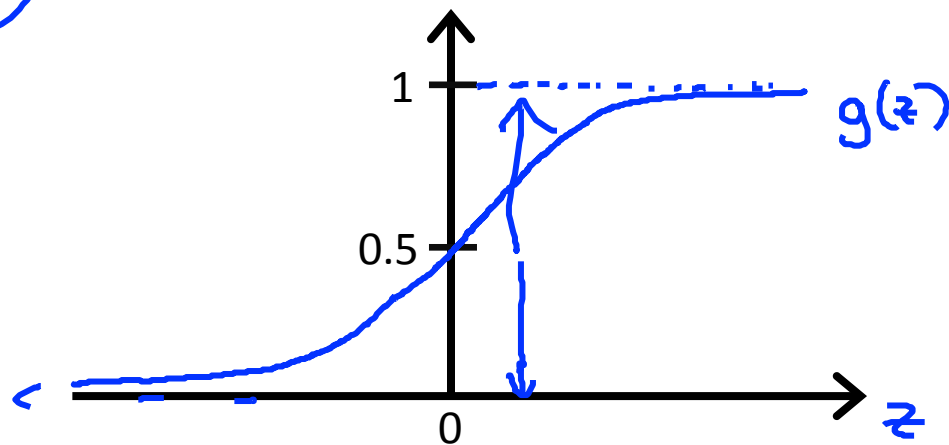
Parameters θ.

Interpretation of Hypothesis Output

$h_\theta(x)$

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

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

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

$$y = 1$$

Tell patient that 70% chance of tumor being malignant

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

$$y = 0 \text{ or } 1$$

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

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

Logistic Regression

Cost function

Machine Learning

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

m examples

$$x \in \begin{bmatrix} x_0 \\ x_1 \\ \cdots \\ x_n \end{bmatrix} \mathbb{R}^{n+1}$$

$x_0 = 1, y \in \{0, 1\}$

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

How to choose parameters θ ?

Cost function

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

Logistic

$\longrightarrow \text{cost}(h_\theta(x^{(i)}), y)$

$\frac{1}{1 + e^{-\theta^T x}}$

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

"non-convex"

$J(\theta)$

"convex"

$J(\theta)$

θ

θ

Andrew Ng

Logistic regression cost function

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

If y = 1

$h_\theta(x)$
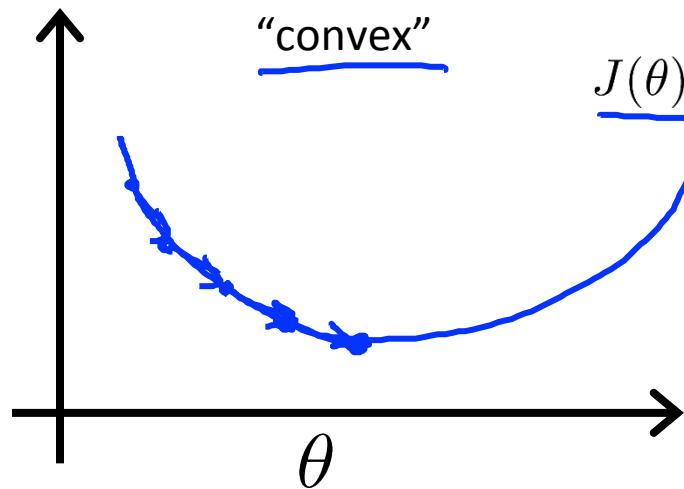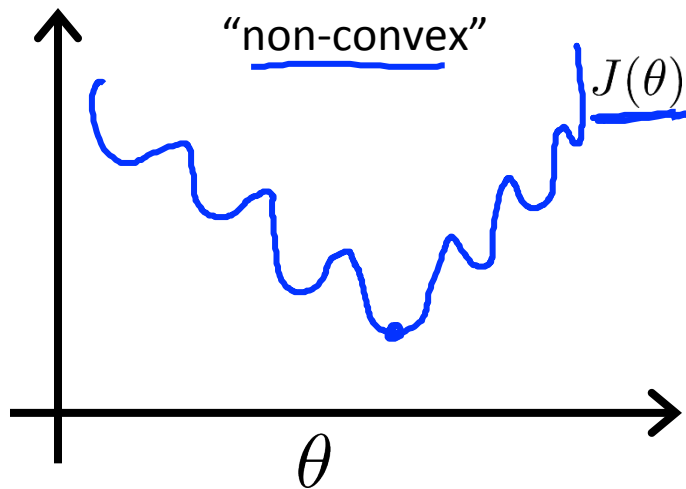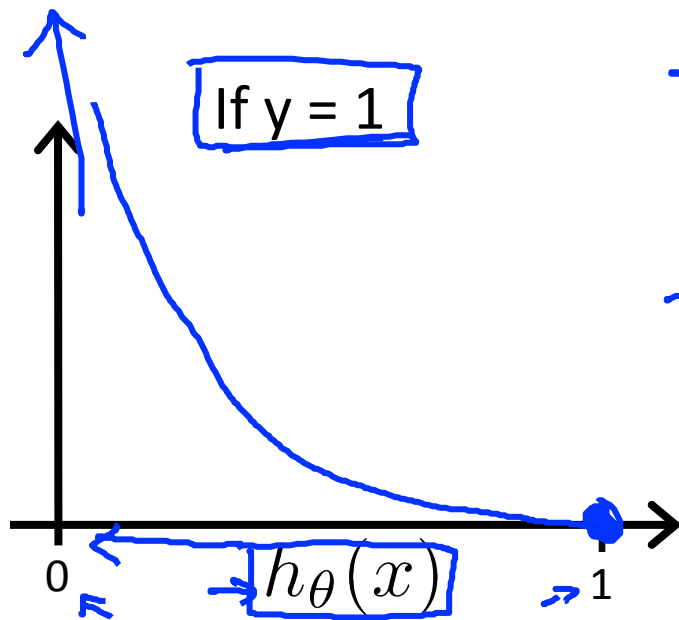
→ Cost $= 0$ if $y = 1$, $h_\theta(x) = 1$

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

$Cost \to \infty$

→ 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

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

If y = 0

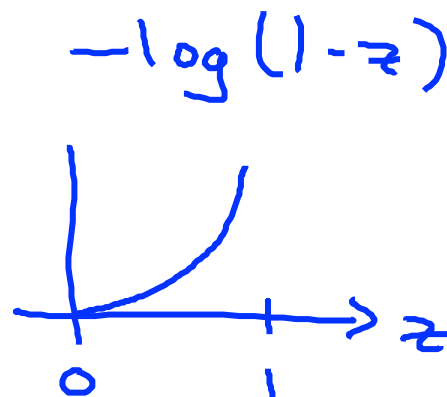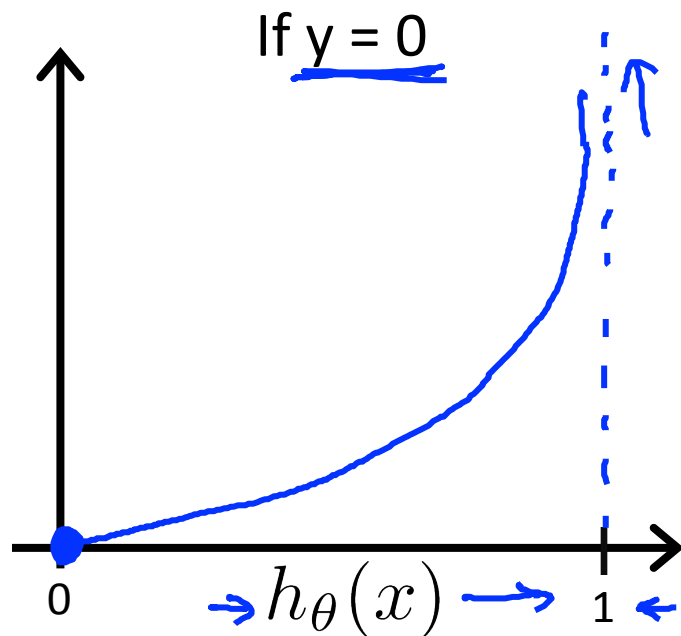$-\log(1-z)$

Logistic Regression

Simplified cost function and gradient descent

Machine Learning

Logistic regression cost function

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

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

Note: $y = 0$ or 1 always

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

If $y = 1$: $\text{Cost}(h_\theta(x), y) = -\log h_\theta(x)$

If $y = 0$: $\text{Cost}(h_\theta(x), y) = -\log(1 - h_\theta(x))$

Logistic regression cost function

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

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

To fit parameters θ :

$$\min_\theta J(\theta) \qquad \text{Get } \theta$$

To make a prediction given new x :

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

Gradient Descent

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

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

Repeat {

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

}

(simultaneously update all θ_j)

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

Gradient Descent

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

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)

}

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix} \leftarrow \quad \text{for } i = 0 \text{ to } n$$

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

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

Algorithm looks identical to linear regression!

Machine Learning

Logistic Regression

Advanced optimization

Optimization algorithm

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

Given θ, 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 θ, 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

Advantages:
- No need to manually pick α
- Often faster than gradient descent.

Disadvantages:
- More complex

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

$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}$ $\theta_1 = 5, \theta_2 = 5.$

$J(\theta) = (\theta_1 - 5)^2 + (\theta_2 - 5)^2$

$\frac{\partial}{\partial \theta_1} J(\theta) = 2(\theta_1 - 5)$

$\frac{\partial}{\partial \theta_2} J(\theta) = 2(\theta_2 - 5)$

```
function [jVal, gradient]
          = costFunction(theta)
  jVal = (theta(1)-5)^2 + ...
          (theta(2)-5)^2;
  gradient = zeros(2,1);
  gradient(1) = 2*(theta(1)-5);
  gradient(2) = 2*(theta(2)-5);
```

```
options = optimset('GradObj', 'on', 'MaxIter', '100');
initialTheta = zeros(2,1);
[optTheta, functionVal, exitFlag] ...
     = fminunc(@costFunction, initialTheta, options);
```

$\theta \in \mathbb{R}^d$ $d \geq 2.$

$$\text{theta} = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix} \begin{matrix} \text{—} & \text{theta(1)} \leftarrow \\ \text{—} & \text{theta(2)} \\ \\ \text{—} & \text{theta(n+1)} \end{matrix}$$

```
function [jVal, gradient] = costFunction(theta)

    jVal = [code to compute J(θ)];

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

    gradient(2) = [code to compute ∂/∂θ₁ J(θ)];
      .
      .
      .
    gradient(n+1) = [code to compute ∂/∂θₙ J(θ)];
```

$$\text{jVal} = [\text{code to compute } J(\theta)];$$

$$\text{gradient(1)} = [\text{code to compute } \frac{\partial}{\partial \theta_0} J(\theta)];$$

$$\text{gradient(2)} = [\text{code to compute } \frac{\partial}{\partial \theta_1} J(\theta)];$$

$$\text{gradient(n+1)} = [\text{code to compute } \frac{\partial}{\partial \theta_n} J(\theta)];$$

Logistic Regression

Multi-class classification: One-vs-all

Machine Learning

Multiclass classification

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

$$y=1 \qquad y=2 \qquad y=3 \qquad y=4$$

Medical diagrams: Not ill, Cold, Flu

$$y=1 \qquad 2 \qquad 3$$

Weather: Sunny, Cloudy, Rain, Snow

$$y=1 \qquad 2 \qquad 3 \qquad 4 \quad \leftarrow$$

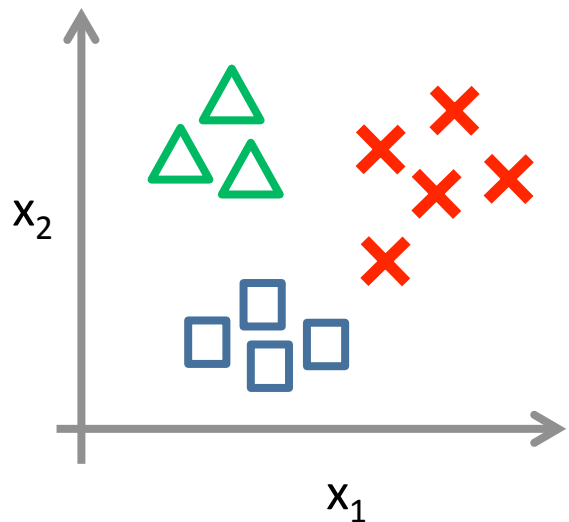$$0 \quad\quad 1 \quad\quad 2 \quad\quad 3$$

Binary classification:
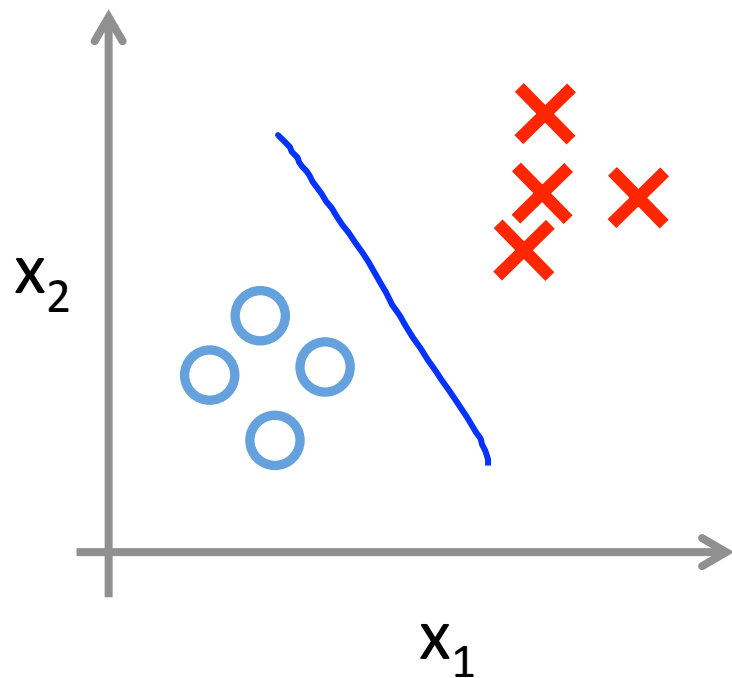
Multi-class classification:
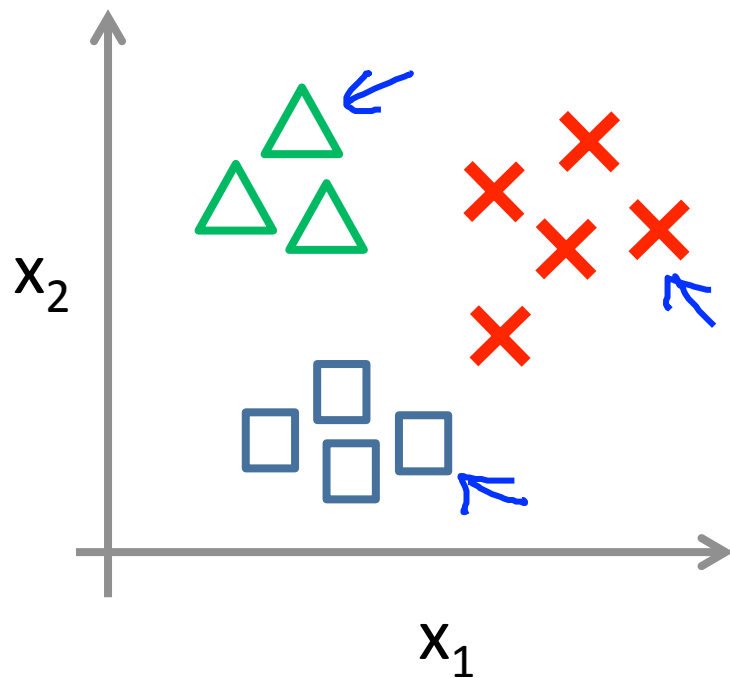
One-vs-all (one-vs-rest):

Class 1: △

Class 2: □

Class 3: ✖

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

$h_\theta^{(1)}(x)$

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

$h_\theta^{(2)}(x)$

$h_\theta^{(3)}(x)$

Andrew Ng
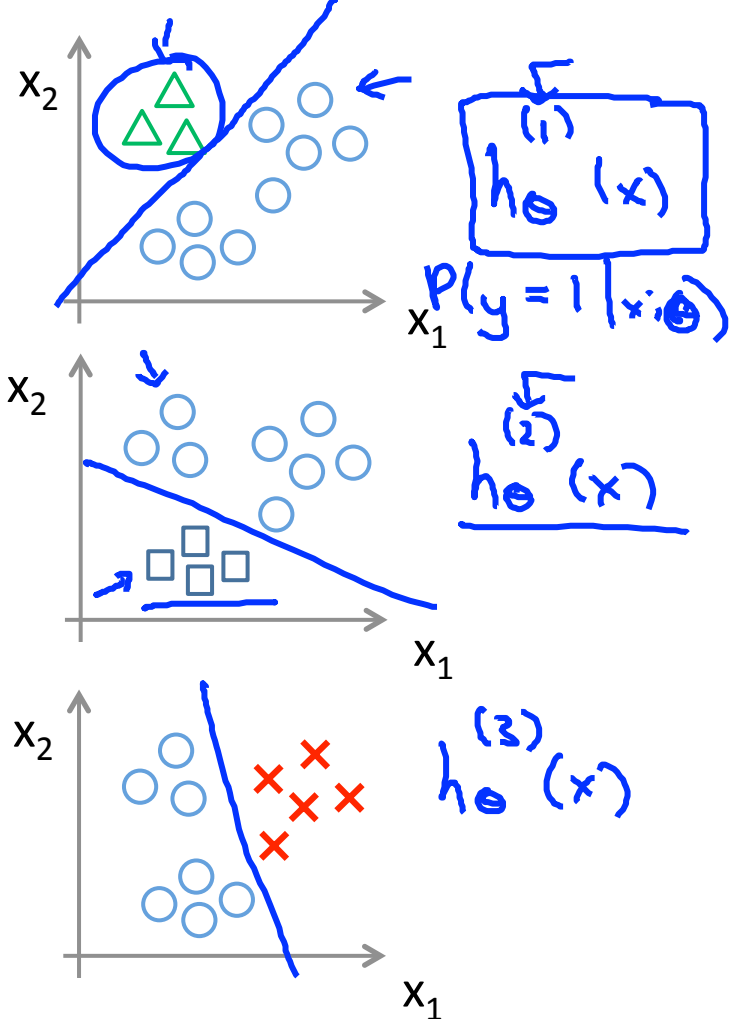
One-vs-all

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

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