6

Logistic regression

6.1> Classification

⇒ Logistic Regression is a classification algorithm.

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

6.2> hypothesis - representation

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

where $g(z) = \frac{1}{1+e^{-z}}$ → Sigmoid function or logistic function

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

→ estimated probability that $y=1$ on input x

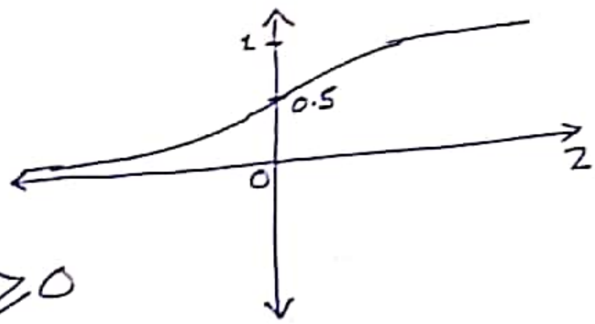
6.3> Decision boundary

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

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

$$h_\theta(z) = \frac{1}{1-e^{-z}}$$

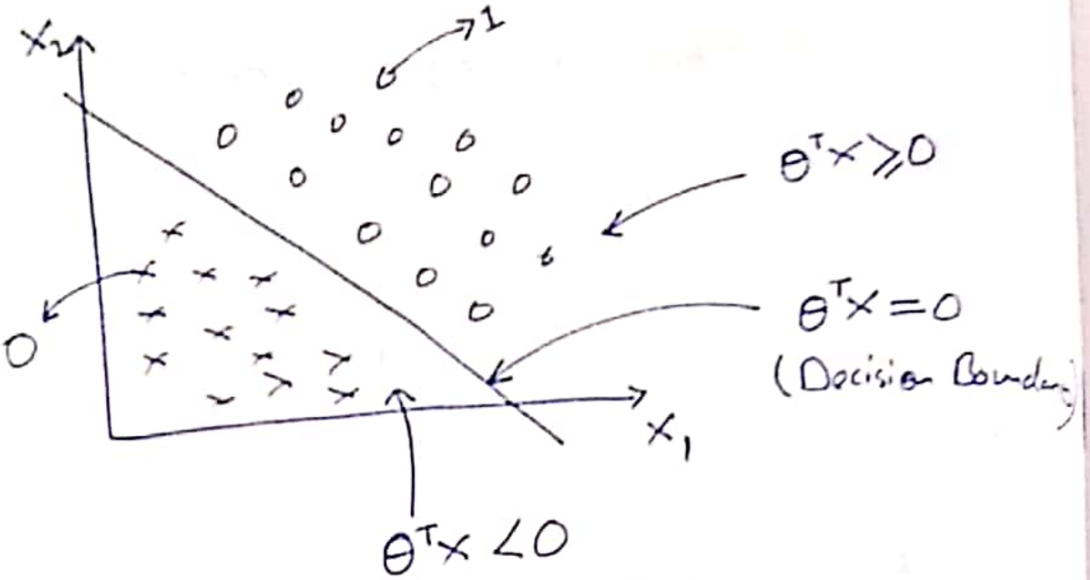
where $z = \theta^T x$



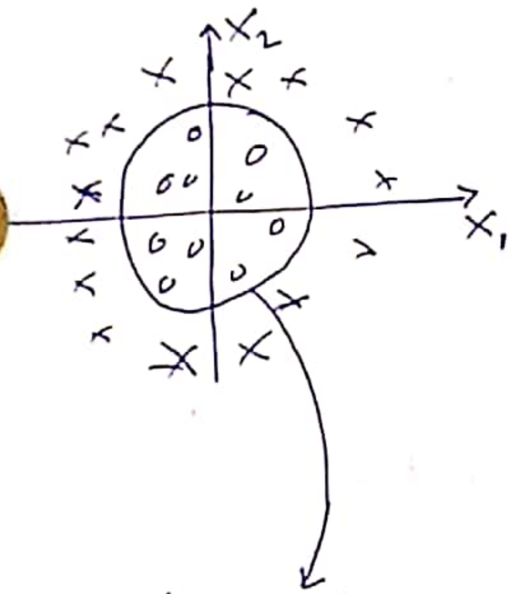
$$h_\theta(z) \geq 0.5 \Rightarrow z \geq 0$$

&

$$h_\theta(z) < 0.5 \Rightarrow z < 0$$



Non linear decision boundaries



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

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

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

$\Rightarrow x_1^2 + x_2^2 \geq 0$

decision boundaries

$$\boxed{x_1^2 + x_2^2 = 1}$$

6.4) Cost function

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

{m examples}

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

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

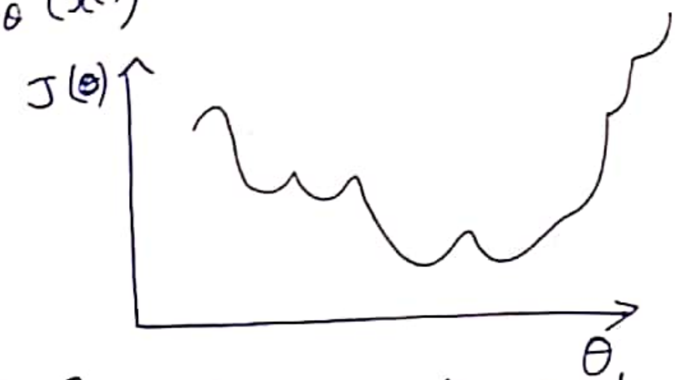
★ <u>Cost function</u>

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

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

{Same as Linear regression}

⇒ <u>then</u>, $J(\theta)$ is not a convex function because of highly non linear term $h_\theta(x^{(i)})$



⇒ So using gradient descent will not guarantee global minima.

⇒ So to avoide this problem we will use the following as cost function.

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

→ It give very high penalty if prediction is wrong and zero penalty if prediction is write.

→ It guarantee a convex function.
(Proving this is out of scope of this course)

Scanned by CamScanner

6.5) Simplified cost function and gradient descent

⇒ Simplified cost function:

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

&

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

* **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 {

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

} {Simultaneously update all θ_j}

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

6.6> Advanced Optimization

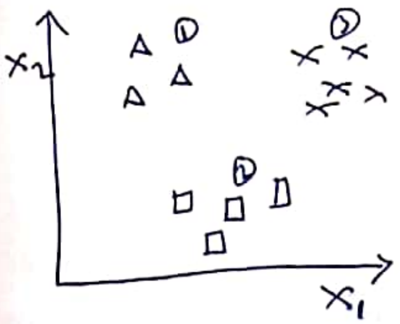
⇒ Other optimization algorithms: {other than <u>gradient descent</u>}

- Conjugate gradient
- BFGS
- L-BFGS

Advantages	Disadvantages
→ No need to manually pick α.	→ More complex
→ Often faster than gradient descent	

6.7> Multiclass-classification (one-vs-all)

⇒ Classification problem with more than two classes.



⇒ Turn this into three separate binary classification problem.

$h_\theta^{(1)}(x) \to \triangle$ vs rest

$h_\theta^{(2)}(x) \to \square$ vs rest

$h_\theta^{(3)}(x) \to \times$ vs rest

① Train a logistic regression classifier $h_\theta^{(i)}(x)$ for each class i to predict the probability that $y = i$.

② On a new input x, to make a prediction, pick the class i that maximizes

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