Machine Learning
Logistic Regression

DR. BHARGAVI R

SCOPE

VIT CHENNAI

Classification - Applications

Binary Classification

- Online transactions – Fraudulent / Not Fraudulent
- Email – Spam/ Not spam ?
- Tumor classification – Malignant/Benign

Multi-class Classification

- Optical Character Recognition
- Face classification

Multi-Label Classification

A variant of the classification problem where multiple nonexclusive labels may be assigned to each instance.

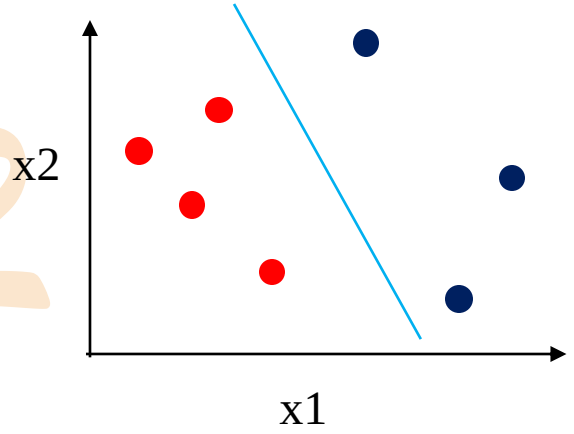
Binary classification

$y \in \{0,1\}$: 0 - Negative class (Not spam)

: 1 – Positive class (Spam)

Logistic Regression - Introduction

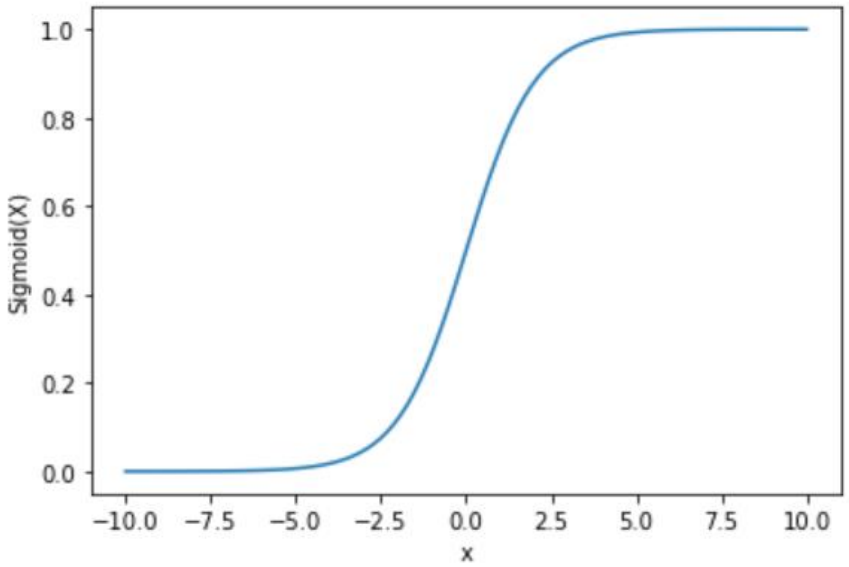
- Linear model.
- Used for binary classification
- Can be extended to handle multiclass as well
- Computationally inexpensive
- Easy to implement.
- Logistic Regression models the response/prediction as probability that y (output variable) belongs to a particular category.



Hypothesis Function

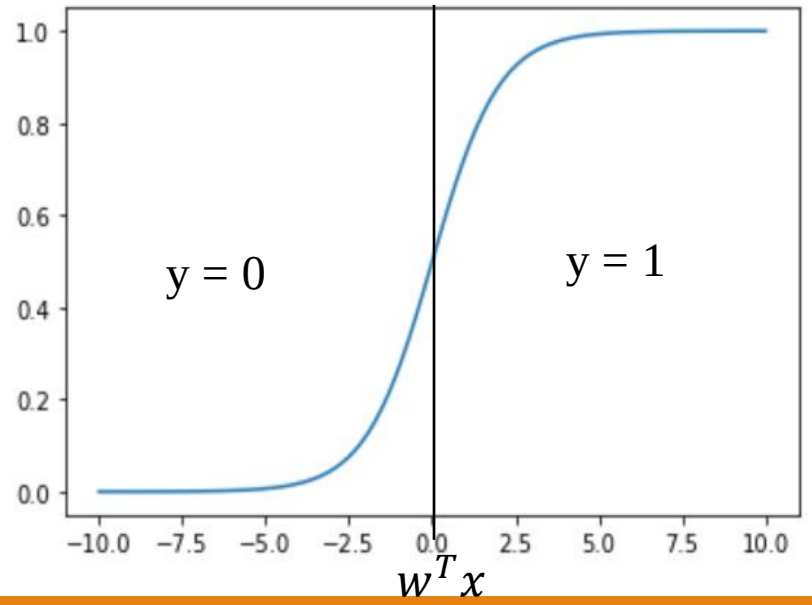
- Can the hypothesis function $g(x, w) = \sum_{i=1}^{m} w_i x_i$ be used for classification?
- $g(x, w)$ results in a real value ($-\infty < g(w,x) < +\infty$).
- For classification problems we need the result to be finite discrete values representing different classes.
- $Sigmoid(x) = \frac{1}{1+e^{-x}}$
- $h_w(x) = \frac{1}{1+e^{-g(x,w)}} = \frac{1}{1+e^{-\sum_{i=1}^{m} w_i x_i}}$
- $h_w(x)$ can also be written as $\frac{1}{1+e^{-w^T X}}$
- $h_w(x)$ is called as Sigmoid/ logistic function
- $0 \leq h_w(x) \leq 1$

(Note: $\log(\text{odds}) = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_m x_m$)

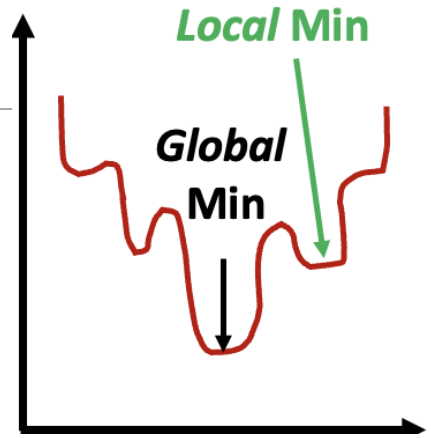


Hypothesis Function (cont...)

- Value of $h_w(x)$ is the estimated probability that y =1, for input x with given w's
- $h_w(x)$ = P(y=1 /x; w) i.e probability that y =1, for input x with given w's
- And P(y=0 /x; w) = 1 - P(y=1 /x; w) (since P(y=1 /x; w) + P(y=0 /x; w) = 1)
- If $h_w(x) \geq 0.5$ i.e $w^T x \geq 0$ then y = 1
- If $h_w(x) < 0.5$ i.e $w^T x < 0$ then y = 0
- For fixed w's, $\boldsymbol{w^T x}$ represent a linear decision boundary
- x_i's that results in $w^T x \geq 0$ are predicted as 1
- x_i's that results in $w^T x < 0$ are predicted as 0



Cost Function



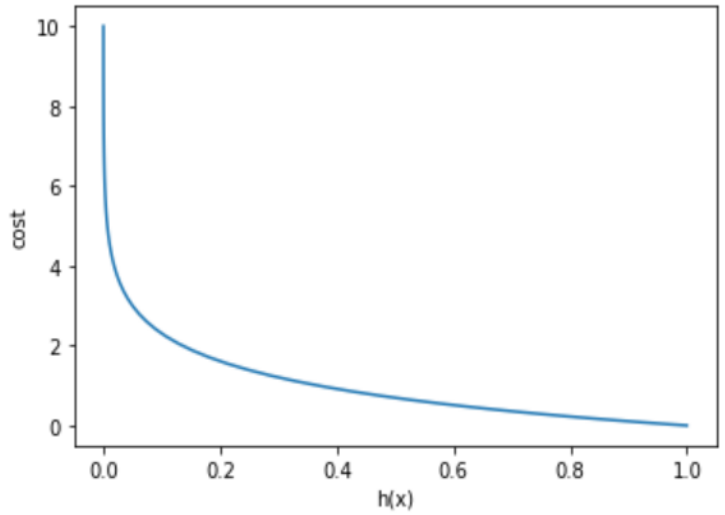
- Can we use RSS/Mean Squared error as the cost function?
- Ans: No
- Why?
- Because of the nonlinear nature of the hypothesis function, RSS, or MSE will not be result in smooth convex functions.
- Hence the solution may stuck in Local Minima.
- **In order to ensure the cost function is convex (and therefore ensure convergence to the global minimum)**, the cost function is transformed using the logarithm of the sigmoid function.

Cost Function (Cont…)

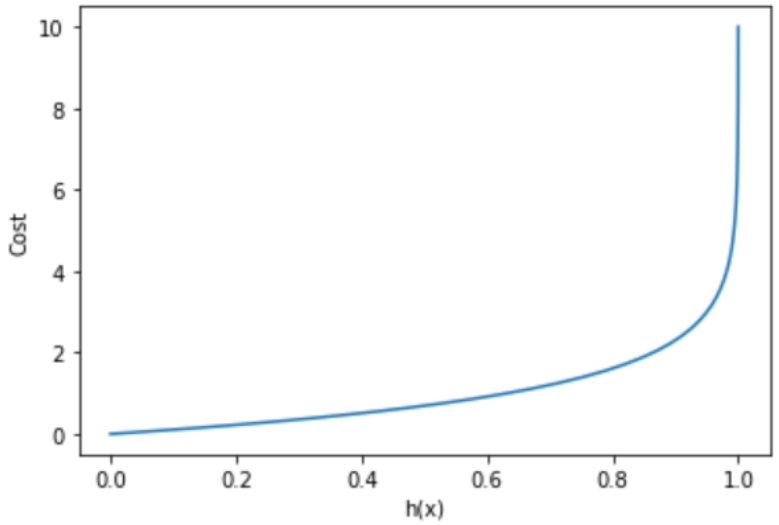
- Cost should be minimum (≈ 0) for the correct predictions and maximum(≈ ∞ or very high value) for the wrong predictions.

For a single observation : Cost ($h_w(x)$, y) = -log($h_w(x)$) for y = 1 (i.e for +ve sample)

-log(1 - $h_w(x)$) for y = 0 (i.e for -ve sample)



-log($h_w(x)$) plot



-log(1-$h_w(x)$) plot

Cost Function (cont…)

- Combining the cost for positive and negative predictions into a single equation

$$\text{Cost}\,(h_w(x), y) = -y log\big(h_w(x)\big) - (1-y)\log\big(1 - h_w(x)\big)$$

- Cost function for n data points can be written as

$$J(w) = \frac{1}{n}\sum_{i=1}^{n} cost(h_w(\boldsymbol{x}_i, y_i))$$

$$= \frac{1}{n}\sum_{i=1}^{n} -y_i log\big(h_w(x_i)\big) - (1-y_i)\log(1 - h_w(x_i))$$

$$= -\frac{1}{n}\sum_{i=1}^{n} y_i log\big(h_w(x_i)\big) + (1-y_i)\log(1 - h_w(x_i))$$

Where $h_w(\boldsymbol{x}_i) = \frac{1}{1+ e^{-W^T x_i}}$

Gradient of Cost Function

Gradient of J(w):

Derivative of hypothesis function

$$\frac{dh_w}{dw} = \frac{-1}{\left(1+ e^{-\boldsymbol{w}^T \boldsymbol{x}_i}\right)^2} \times e^{-\boldsymbol{w}^T \boldsymbol{x}_i} \times (-\boldsymbol{x}_i)$$

$$= \frac{1}{1+ e^{-\boldsymbol{w}^T \boldsymbol{x}_i}} \frac{e^{-\boldsymbol{w}^T \boldsymbol{x}_i}}{1+ e^{-\boldsymbol{w}^T \boldsymbol{x}_i}} \boldsymbol{x}_i = h_w(1 - h_w)x_i$$

Substituting partial derivative of the hypothesis in the cost function and simplifying we get

Gradient of J(w) (say w.r.t w_j) = $\frac{1}{n}\sum_{i=1}^{n}(h_w(\boldsymbol{x}_i) - y_i)x_{ij}$

Gradient Descent

Do repeatedly
{

$$w_j = w_j - \alpha \frac{\partial}{\partial w_j}(J(W))$$

}

or

Do repeatedly
{

$$w_j = w_j - \alpha \sum_{i=1}^{n}(h_w(\boldsymbol{x}_i) - y_i)x_{ij}$$ (Simultaneously update all W_j's)

}

Here α is the learning rate

Prediction

The marketing department of a credit card company wants to organize a campaign to convince existing holders of the company's standard credit card to upgrade to the company's premium card for a nominal annual fee. The marketing department begins with the question "Which of the existing standard credit cardholders should be the target for the campaign?"

Dataset - 30 cardholders data that indicates whether the cardholder upgraded to a premium or not (y i.e response)

Two independent variables/features:

1. Total amount of credit card purchases in the prior year(x1)
2. Whether the cardholder ordered additional credit cards (at extra cost) for other members of the household (x2 : 0 no, 1 yes).

The regression coefficient vales are w_0 = -6.940, w_1 = 0.13947, w_2 = 2.774

Prediction (cont…)

Consider a cardholder who charged $36,000 last year and possesses additional cards for members of the household. What is the probability the cardholder will upgrade to the premium card during the marketing campaign.

Substitute the w_is and x_is in the function $h_w(\boldsymbol{x}_i) = \frac{1}{1+ e^{-W^T \boldsymbol{x_i}}}$ to the predicted probability.

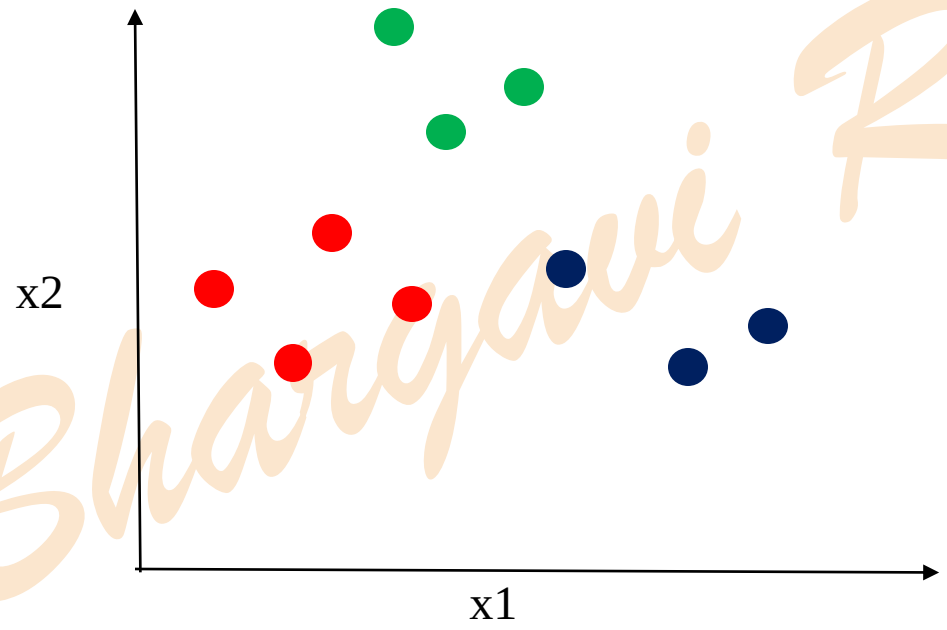
-6.94 + (0.13947)(36) + (2.774)(1) = 0.85492

$e^{-(0.85492)} = 0.423$

Estimated probability of purchasing premium card = 1 / (1 + 0.423) = 0.702

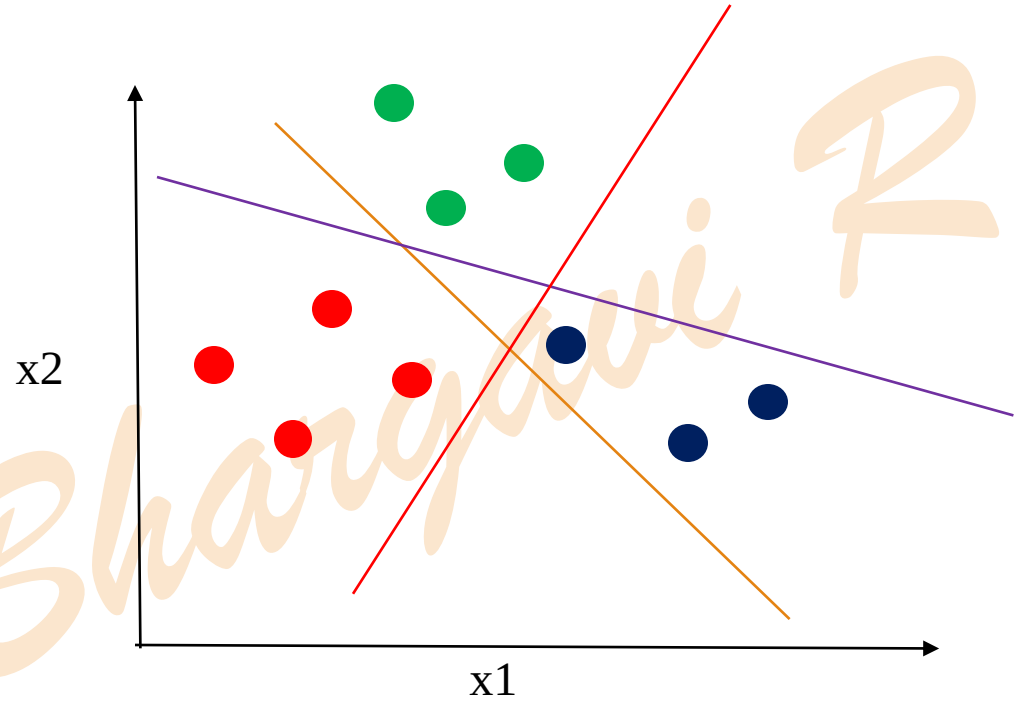
Multiclass Classification

$y \in \{1, 2, 3, \ldots.\}$



Multiclass Classification (cont…)

One-vs-all (or) one-vs-rest



Multiclass Classification (cont…)

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

Step1 : Modify the training data such that only one specific class has y=1 and rest all have y=0.

Step 2: Train the classifier.

Step3: Repeat Step 1 and 2 for remaining all classes each time making one class as y=1 and remaining all as y=0 and training individual models.

Step 4: Prediction : For a new test input x_t pick a class that maximizes the $h_w(x_t)$

Numerical Example

Gradient Descent for parameter updation:

Repeat until convergence{

$$\theta_j = \theta_j - \alpha \sum_{i=1}^{n} \left(\frac{1}{1 + e^{-\theta^T x^{(i)}}} - y^{(i)} \right) x_j{}^{(i)}$$

}

Training

Apply logistic regression on the spam email recognition problem, assuming $\alpha = 0.5$ and starting with $\theta = [0, 0, 0, 0, 0, 0]$

	and	vaccine	the	of	nigeria	y
Email **a**	1	1	0	1	1	1
Email **b**	0	0	1	1	0	0
Email **c**	0	1	1	0	0	1
Email **d**	1	0	0	1	0	0
Email **e**	1	0	1	0	1	1
Email **f**	1	0	1	1	0	0

Training (cont…)

	$x_0 = 1$	$x_1 =$ and	$x_2 =$ vaccine	$x_3 =$ the	$x_4 =$ of	$x_5 =$ nigeria	y
Email **a**	1	1	1	0	1	1	1
Email **b**	1	0	0	1	1	0	0
Email **c**	1	0	1	1	0	0	1
Email **d**	1	1	0	0	1	0	0
Email **e**	1	1	0	1	0	1	1
Email **f**	1	1	0	1	1	0	0

Training (cont…)

x	y	$\theta^T x$	$(\frac{1}{1+e^{-\theta^T x}} - y)x_0$
[1,1,1,0,1,1]	1	[0,0,0,0,0,0]×[1,1,1,0,1,1]=0	$(\frac{1}{1+e^{-0}} - 1) \times 1 =$ -0.5
[1,0,0,1,1,0]	0	[0,0,0,0,0,0]×[1,0,0,1,1,0]=0	$(\frac{1}{1+1} - 0) \times 1 = 0.5$
[1,0,1,1,0,0]	1	[0,0,0,0,0,0]×[1,0,1,1,0,0]=0	$(\frac{1}{1+1} - 1) \times 1 =$ -0.5
[1,1,0,0,1,0]	0	[0,0,0,0,0,0]×[1,1,0,0,1,0]=0	$(\frac{1}{1+1} - 0) \times 1 = 0.5$
[1,1,0,1,0,1]	1	[0,0,0,0,0,0]×[1,1,0,1,0,1]=0	$(\frac{1}{1+1} - 1) \times 1 =$ -0.5
[1,1,0,1,1,0]	0	[0,0,0,0,0,0]×[1,1,0,1,1,0]=0	$(\frac{1}{1+1} - 0) \times 1 = 0.5$

$$\sum_{i=1}^{n} \left(\frac{1}{1 + e^{-\boldsymbol{\theta}^T x^{(i)}}} - y^{(i)} \right) x_0{}^{(i)} = \mathbf{0}$$

Then,

$$\theta_0 = \theta_0 - \alpha \times \mathbf{0}$$

$$= 0 - 0.5 \times \mathbf{0} = \mathbf{0}$$

Training (cont…)

$$\sum_{i=1}^{n}\left(\frac{1}{1+e^{-\boldsymbol{\theta}^T x^{(i)}}}-y^{(i)}\right)x_0{}^{(i)}=\mathbf{0}$$

Now,

$$\theta_0=\theta_0-\alpha\times\mathbf{0}$$

$$=0-0.5\times\mathbf{0}=\mathbf{0}$$

Training (cont…)

$\boldsymbol{x}$	$\boldsymbol{y}$	$\boldsymbol{\theta^T x}$	$(\frac{1}{1+e^{-\theta^T x}} - \boldsymbol{y})\boldsymbol{x_1}$
[1,1,1,0,1,1]	1	[0,0,0,0,0,0]×[1,1,1,0,1,1]=0	-0.5
[1,0,0,1,1,0]	0	[0,0,0,0,0,0]×[1,0,0,1,1,0]=0	0
[1,0,1,1,0,0]	1	[0,0,0,0,0,0]×[1,0,1,1,0,0]=0	0
[1,1,0,0,1,0]	0	[0,0,0,0,0,0]×[1,1,0,0,1,0]=0	0.5
[1,1,0,1,0,1]	1	[0,0,0,0,0,0]×[1,1,0,1,0,1]=0	-0.5
[1,1,0,1,1,0]	0	[0,0,0,0,0,0]×[1,1,0,1,1,0]=0	0.5

$$\sum_{i=1}^{n}\left(\frac{1}{1+e^{-\boldsymbol{\theta}^T x^{(i)}}} - y^{(i)}\right) x_1{}^{(i)} = \mathbf{0}$$

Now,

$$\theta_1 = \theta_1 - \alpha \times \mathbf{0}$$

$$= 0 - 0.5 \times \mathbf{0} = \mathbf{0}$$

Training (cont…)

$\boldsymbol{x}$	$\boldsymbol{y}$	$\boldsymbol{\theta^T x}$	$(\frac{1}{1+e^{-\theta^T x}} - \boldsymbol{y})\boldsymbol{x_2}$
[1,1,1,0,1,1]	1	[0,0,0,0,0,0]×[1,1,1,0,1,1]=0	-0.5
[1,0,0,1,1,0]	0	[0,0,0,0,0,0]×[1,0,0,1,1,0]=0	0
[1,0,1,1,0,0]	1	[0,0,0,0,0,0]×[1,0,1,1,0,0]=0	-0.5
[1,1,0,0,1,0]	0	[0,0,0,0,0,0]×[1,1,0,0,1,0]=0	0
[1,1,0,1,0,1]	1	[0,0,0,0,0,0]×[1,1,0,1,0,1]=0	0
[1,1,0,1,1,0]	0	[0,0,0,0,0,0]×[1,1,0,1,1,0]=0	0

$$\sum_{i=1}^{n}\left(\frac{1}{1+e^{-\boldsymbol{\theta}^T x^{(i)}}} - y^{(i)}\right) {x_2}^{(i)} = \boldsymbol{-1}$$

$\boldsymbol{Now,}$

$$\theta_2 = \theta_2 - \alpha \times (\boldsymbol{-1})$$

$$= 0 - 0.5 \times (\boldsymbol{-1}) = \boldsymbol{0.5}$$

Training (cont…)

x	y	$\theta^T x$	$(\frac{1}{1+e^{-\theta^T x}} - y)x_3$
[1,1,1,0,1,1]	1	[0,0,0,0,0,0]×[1,1,1,0,1,1]=0	0
[1,0,0,1,1,0]	0	[0,0,0,0,0,0]×[1,0,0,1,1,0]=0	0.5
[1,0,1,1,0,0]	1	[0,0,0,0,0,0]×[1,0,1,1,0,0]=0	-0.5
[1,1,0,0,1,0]	0	[0,0,0,0,0,0]×[1,1,0,0,1,0]=0	0
[1,1,0,1,0,1]	1	[0,0,0,0,0,0]×[1,1,0,1,0,1]=0	-0.5
[1,1,0,1,1,0]	0	[0,0,0,0,0,0]×[1,1,0,1,1,0]=0	0.5

$$\sum_{i=1}^{n}\left(\frac{1}{1+e^{-\boldsymbol{\theta}^T x^{(i)}}} - y^{(i)}\right) {x_3}^{(i)} = \mathbf{0}$$

Now,

$$\theta_3 = \theta_3 - \alpha \times \mathbf{0}$$

$$= 0 - 0.5 \times 0 = \mathbf{0}$$

Training (cont…)

$\boldsymbol{x}$	$\boldsymbol{y}$	$\boldsymbol{\theta^T x}$	$(\frac{1}{1+e^{-\theta^T x}} - \boldsymbol{y})\boldsymbol{x_4}$
[1,1,1,0,1,1]	1	[0,0,0,0,0,0]×[1,1,1,0,1,1]=0	-0.5
[1,0,0,1,1,0]	0	[0,0,0,0,0,0]×[1,0,0,1,1,0]=0	0.5
[1,0,1,1,0,0]	1	[0,0,0,0,0,0]×[1,0,1,1,0,0]=0	0
[1,1,0,0,1,0]	0	[0,0,0,0,0,0]×[1,1,0,0,1,0]=0	0.5
[1,1,0,1,0,1]	1	[0,0,0,0,0,0]×[1,1,0,1,0,1]=0	0
[1,1,0,1,1,0]	0	[0,0,0,0,0,0]×[1,1,0,1,1,0]=0	0.5

$$\sum_{i=1}^{n}\left(\frac{1}{1+e^{-\boldsymbol{\theta}^T x^{(i)}}} - y^{(i)}\right) x_4{}^{(i)} = \mathbf{1}$$

Now,

$$\theta_4 = \theta_4 - \alpha \times \mathbf{1}$$

$$= 0 - 0.5 \times 1 = -\mathbf{0.5}$$

Training (cont…)

x	y	$\theta^T x$	$(\frac{1}{1+e^{-\theta^T x}} - y)x_5$
[1,1,1,0,1,1]	1	[0,0,0,0,0,0]×[1,1,1,0,1,1]=0	-0.5
[1,0,0,1,1,0]	0	[0,0,0,0,0,0]×[1,0,0,1,1,0]=0	0
[1,0,1,1,0,0]	1	[0,0,0,0,0,0]×[1,0,1,1,0,0]=0	0
[1,1,0,0,1,0]	0	[0,0,0,0,0,0]×[1,1,0,0,1,0]=0	0
[1,1,0,1,0,1]	1	[0,0,0,0,0,0]×[1,1,0,1,0,1]=0	-0.5
[1,1,0,1,1,0]	0	[0,0,0,0,0,0]×[1,1,0,1,1,0]=0	0

$$\sum_{i=1}^{n}\left(\frac{1}{1+e^{-\theta^T x^{(i)}}} - y^{(i)}\right) x_5{}^{(i)} = -\mathbf{1}$$

Now,

$$\theta_5 = \theta_5 - \alpha \times (-\mathbf{1})$$

$$= 0 - 0.5 \times (-1) = \mathbf{0.5}$$

New Paramter Vector:

$$\theta = [0, 0, 0.5, 0, -0.5, 0.5]$$

Prediction

$(if\ h_\theta(x) \geq 0.5, y' = 1; else\ y' = 0)$

x	y	$\theta^T x$	$h_\theta(x) = (\frac{1}{1+e^{-\theta^T x}})$	Predicted Class (or y')
[1,1,1,0,1,1]	**1**	[0,0,0.5,0,-0.5,0.5]×[1,1,1,0,1,1]=0.5	0.622459331	**1**
[1,0,0,1,1,0]	**0**	[0,0,0.5,0,-0.5,0.5]×[1,0,0,1,1,0]=-0.5	0.377540669	**0**
[1,0,1,1,0,0]	**1**	[0,0,0.5,0,-0.5,0.5]×[1,0,1,1,0,0]=0.5	0.622459331	**1**
[1,1,0,0,1,0]	**0**	[0,0,0.5,0,-0.5,0.5]×[1,1,0,0,1,0]=-0.5	0.377540669	**0**
[1,1,0,1,0,1]	**1**	[0,0,0.5,0,-0.5,0.5]×[1,1,0,1,0,1]=0.5	0.622459331	**1**
[1,1,0,1,1,0]	**0**	[0,0,0.5,0,-0.5,0.5]×[1,1,0,1,1,0]=-0.5	0.377540669	**0**