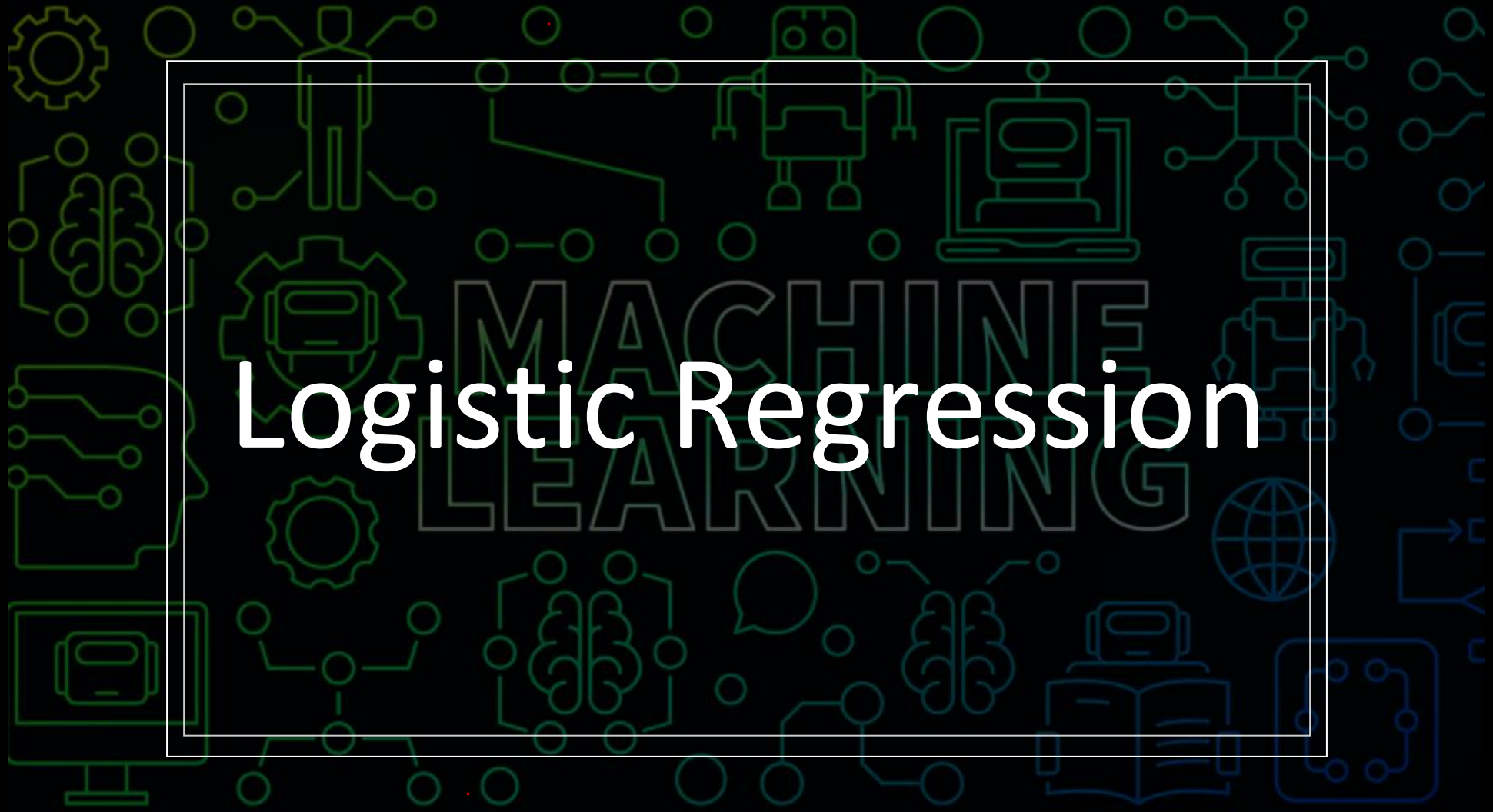


Logistic Regression

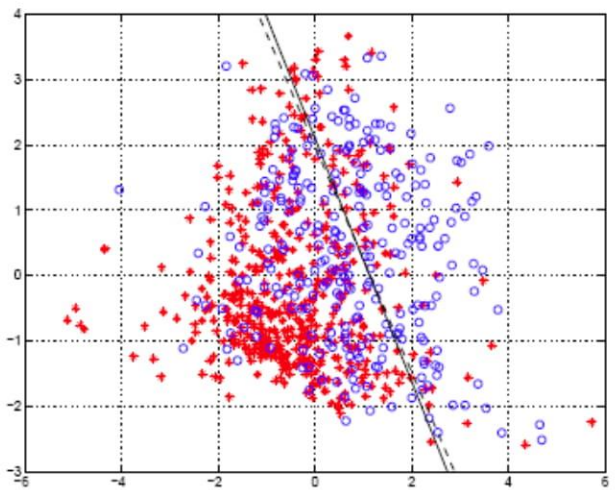
Classification Based on Probability

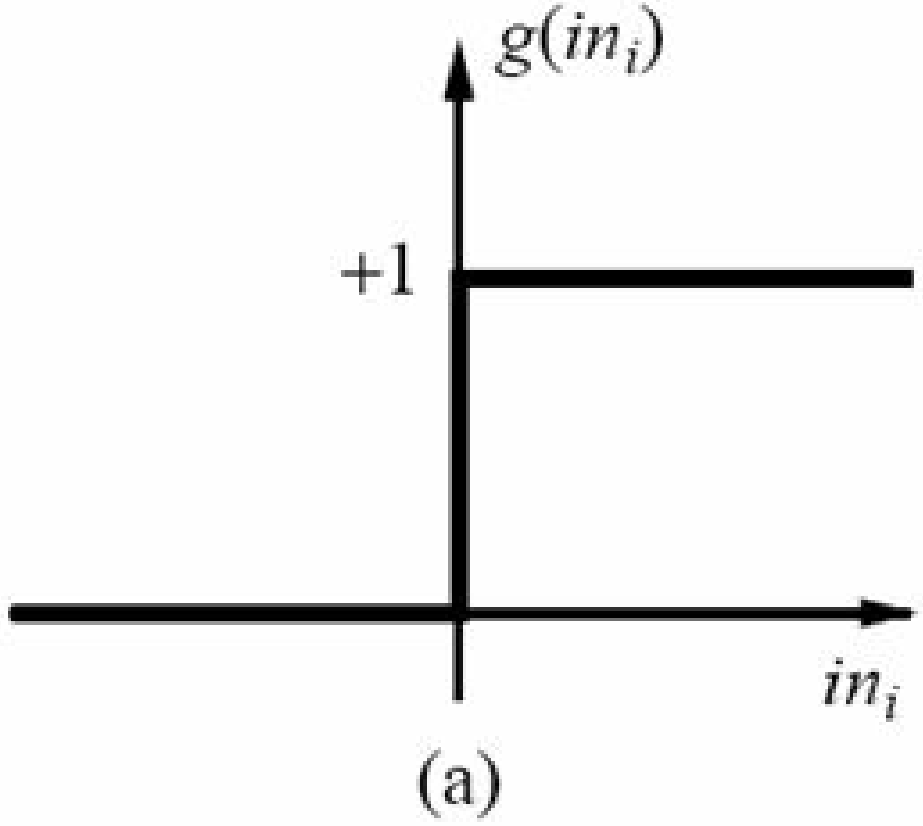
- Instead of just predicting the class, give the probability of the instance being that class
  - i.e., learn $p(y \mid \boldsymbol{x})$
- Comparison to perceptron:
  - Perceptron doesn't produce probability estimate

- Recall that:

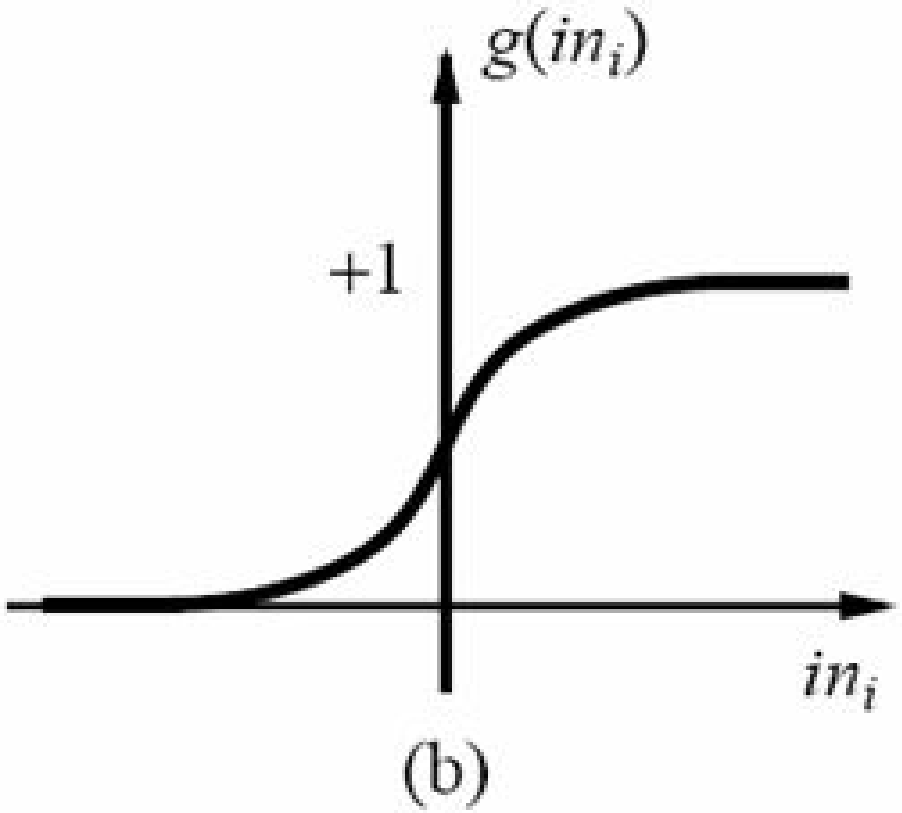
$$0 \leq p(\text{event}) \leq 1$$

$$p(\text{event}) + p(\neg\text{event}) = 1$$





step function



sigmoid function

Logistic Regression

- Takes a probabilistic approach to learning discriminative functions (i.e., a classifier)
- $h_{\boldsymbol{\theta}}(\boldsymbol{x})$ should give $p(y = 1 \mid \boldsymbol{x}; \boldsymbol{\theta})$
  - Want $0 \leq h_{\boldsymbol{\theta}}(\boldsymbol{x}) \leq 1$

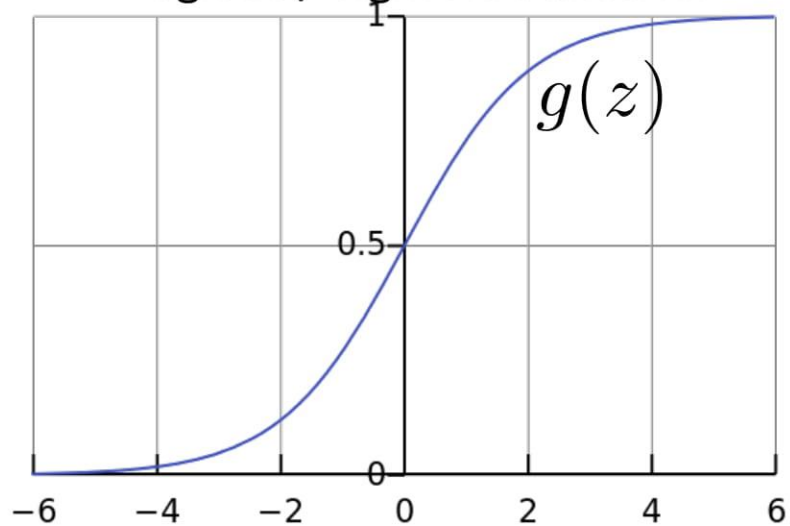
- Logistic regression model:

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = g\left(\boldsymbol{\theta}^{\mathsf{T}}\boldsymbol{x}\right)$$

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

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

Logistic / Sigmoid Function



Interpretation of Hypothesis Output

$h_{\boldsymbol{\theta}}(\boldsymbol{x})$ = estimated $p(y = 1 \mid \boldsymbol{x}; \boldsymbol{\theta})$

Example: Cancer diagnosis from tumor size

$$\boldsymbol{x} = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 1 \\ \text{tumorSize} \end{bmatrix}$$

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = 0.7$$

→ Tell patient that 70% chance of tumor being malignant

Note that: $p(y = 0 \mid \boldsymbol{x}; \boldsymbol{\theta}) + p(y = 1 \mid \boldsymbol{x}; \boldsymbol{\theta}) = 1$

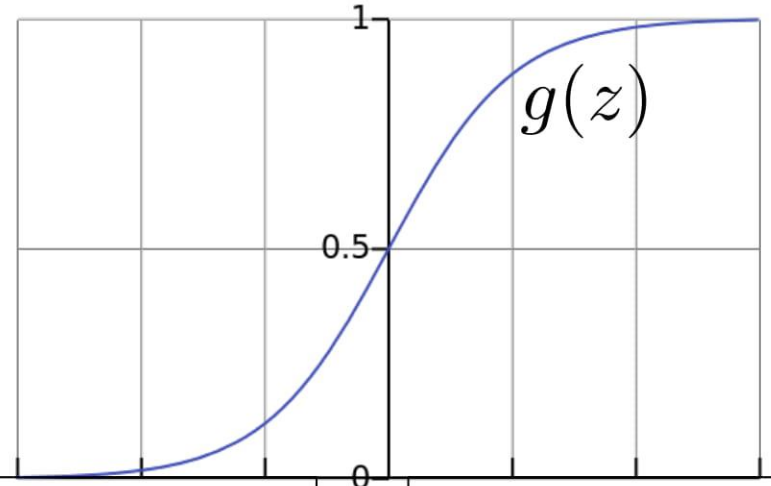
Therefore, $p(y = 0 \mid \boldsymbol{x}; \boldsymbol{\theta}) = 1 - p(y = 1 \mid \boldsymbol{x}; \boldsymbol{\theta})$

Based on example by Andrew Ng

Logistic Regression

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = g\left(\boldsymbol{\theta}^{\mathsf{T}}\boldsymbol{x}\right)$$

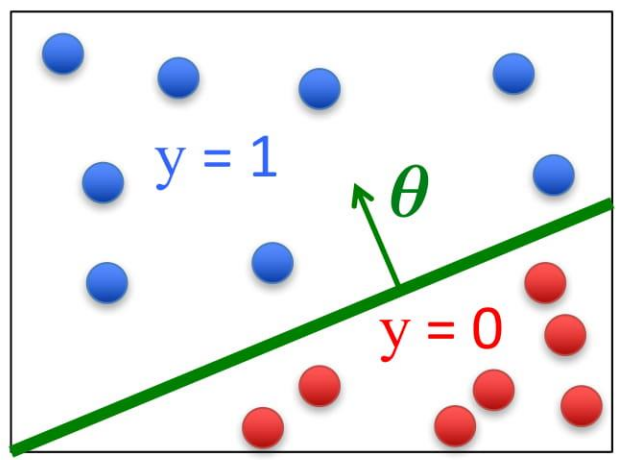
$$g(z) = \frac{1}{1+e^{-z}}$$



$\boldsymbol{\theta}^{\mathsf{T}}\boldsymbol{x}$ should be large <u>negative</u> values for negative instances

$\boldsymbol{\theta}^{\mathsf{T}}\boldsymbol{x}$ should be large <u>positive</u> values for positive instances

- Assume a threshold and...
  - Predict y = 1 if $h_{\boldsymbol{\theta}}(\boldsymbol{x}) \geq 0.5$
  - Predict y = 0 if $h_{\boldsymbol{\theta}}(\boldsymbol{x}) < 0.5$



Logistic Regression

- Given $\left\{ \left(\boldsymbol{x}^{(1)}, y^{(1)} \right), \left(\boldsymbol{x}^{(2)}, y^{(2)} \right), \ldots, \left(\boldsymbol{x}^{(n)}, y^{(n)} \right) \right\}$

  where $\boldsymbol{x}^{(i)} \in \mathbb{R}^d, \; y^{(i)} \in \{0, 1\}$

- Model: $h_{\boldsymbol{\theta}}(\boldsymbol{x}) = g\left(\boldsymbol{\theta}^\intercal \boldsymbol{x}\right)$

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

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_d \end{bmatrix} \qquad \boldsymbol{x}^\intercal = \begin{bmatrix} 1 & x_1 & \ldots & x_d \end{bmatrix}$$

Logistic Regression Objective Function

- Can't just use squared loss as in linear regression:

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

  - Using the logistic regression model

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

    results in a non-convex optimization

Logistic regression objective:

$$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$$

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

Intuition Behind the Objective

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

- Cost of a single instance:

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

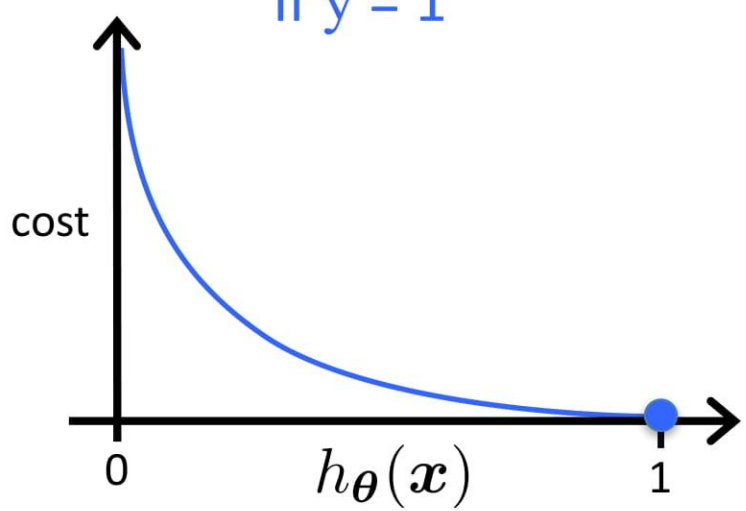
- Can re-write objective function as

$$J(\boldsymbol{\theta}) = \sum_{i=1}^{n} \text{cost}\left(h_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}), y^{(i)}\right)$$

Compare to linear regression: $J(\boldsymbol{\theta}) = \frac{1}{2n}\sum_{i=1}^{n}\left(h_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right) - y^{(i)}\right)^2$

Intuition Behind the Objective

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



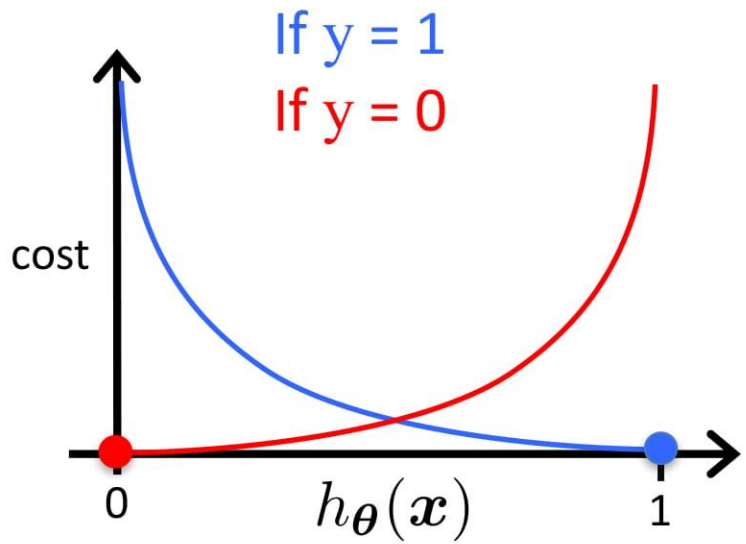
If y = 1

- Cost = 0 if prediction is correct
- As $h_{\boldsymbol{\theta}}(\boldsymbol{x}) \to 0, \text{cost} \to \infty$
- Captures intuition that larger mistakes should get larger penalties
  - e.g., predict $h_{\boldsymbol{\theta}}(\boldsymbol{x}) = 0$, but y = 1

Based on example by Andrew Ng

Intuition Behind the Objective

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



If y = 0

- Cost = 0 if prediction is correct
- As $(1 - h_{\boldsymbol{\theta}}(\boldsymbol{x})) \to 0, \text{cost} \to \infty$
- Captures intuition that larger mistakes should get larger penalties

Based on example by Andrew Ng

$$J(\hat{y}, y) = -((y \log \hat{y}) + (1-y)(\log(1-\hat{y})))$$

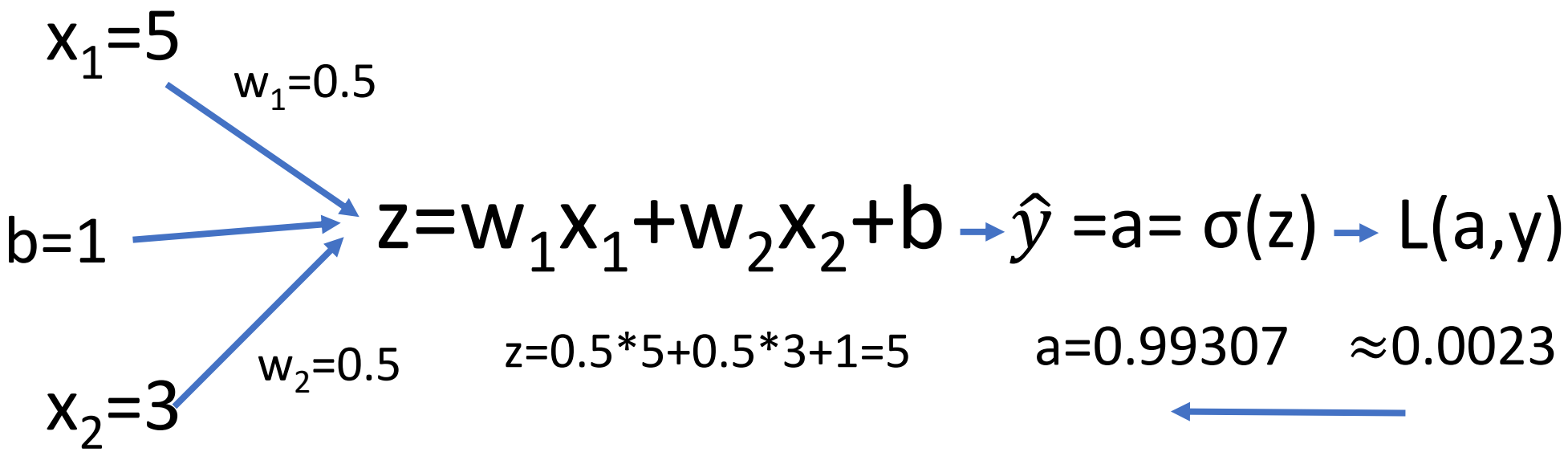
$$J(\hat{y}, y) = -((y \log \hat{y}) + (1-y)(\log(1-\hat{y})))$$

$$\hat{y} = \frac{1}{1+e^{-z}}$$

$$z = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2$$

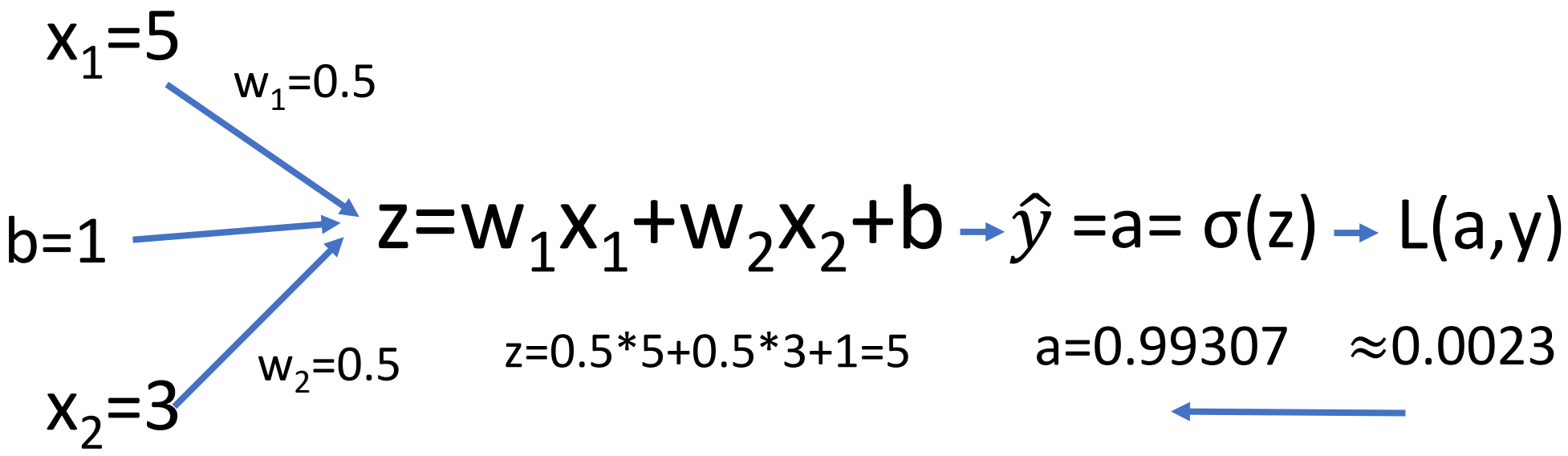
Derivatives in logistic regression

Original Value of y=1



Derivatives in logistic regression

Original Value of y=1

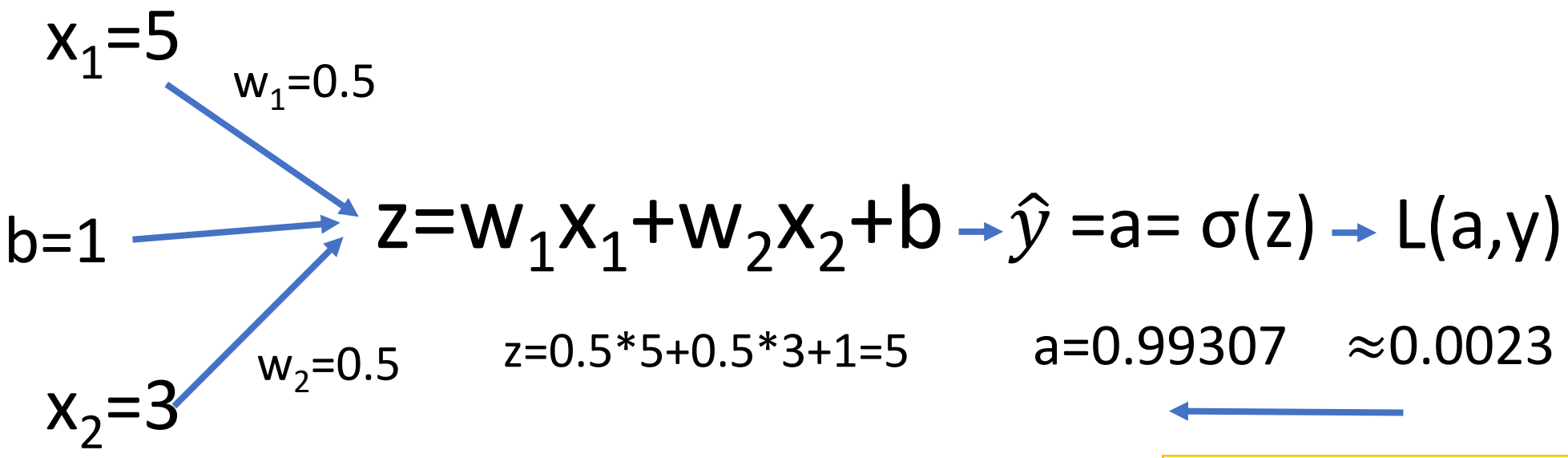


db=dz

$$da=\frac{dL(a,y)}{da}=\frac{-y}{a}+\frac{1-y}{1-a}=-1.006978$$

Derivatives in logistic regression

Original Value of y=1



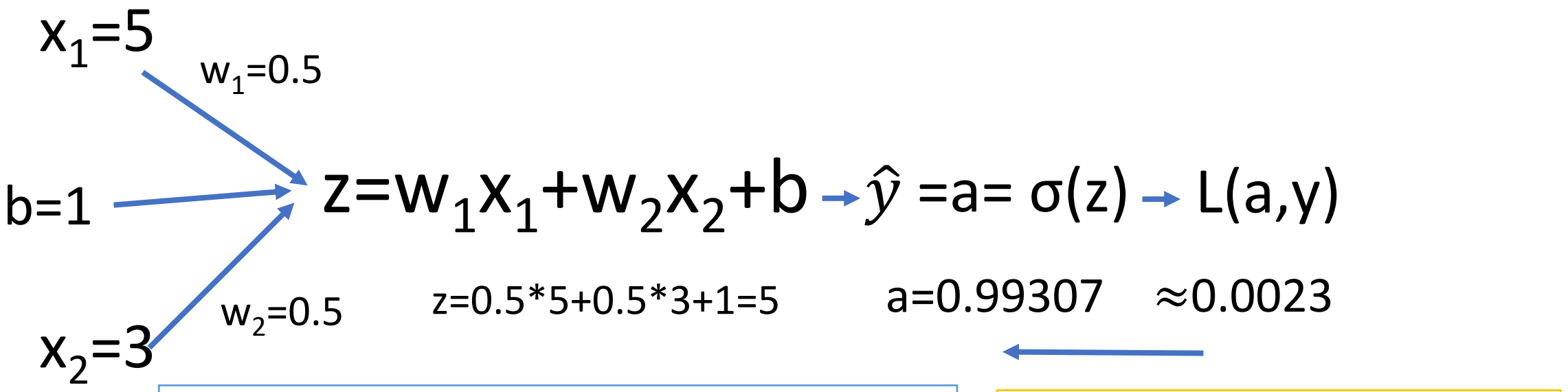
db=dz

$da=\frac{dL(a,y)}{da}=\frac{-y}{a}+\frac{1-y}{1-a}$ = -1.006978

$dz=\frac{dL(a,y)}{dz}=\frac{dL}{da}\frac{da}{dz}=(\frac{-y}{a}+\frac{1-y}{1-a})$*a(1-a)=a-y=-0.00693

Derivatives in logistic regression

Original Value of y=1



dz=$\frac{dL(a,y)}{dz}$ = $\frac{dL}{da}\frac{da}{dz}$ = ($\frac{-y}{a}$ + $\frac{1-y}{1-a}$)*a(1-a)=a-y=-0.00693

da=$\frac{dL(a,y)}{da}$ = $\frac{-y}{a}$ + $\frac{1-y}{1-a}$ = -1.006978

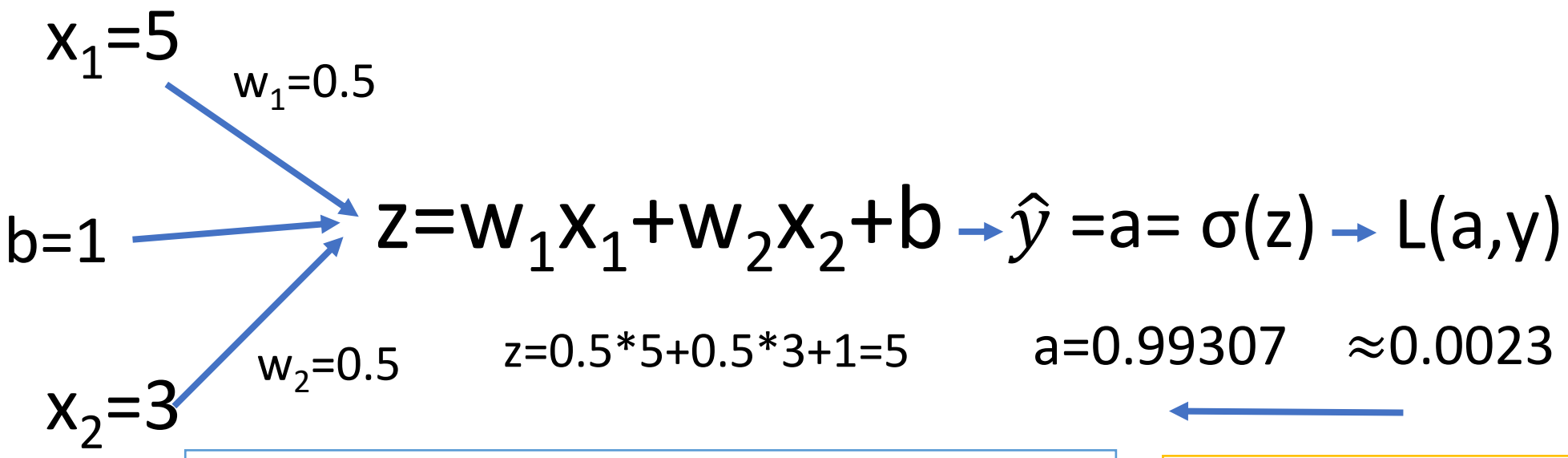
$dw_1 = \frac{dL(a,y)}{dw_1} = x_1.dz = 5*-0.00695 = -0.03475$

db=dz

$dw_2 = \frac{dL(a,y)}{dw_2} = x2.dz = 3*-0.00695 = -0.02085$

Derivatives in logistic regression

Original Value of y=1



dz=$\frac{dL(a,y)}{dz}$ = $\frac{dL}{da}\frac{da}{dz}$ = ($\frac{-y}{a}$ + $\frac{1-y}{1-a}$)*a(1-a)=a-y=-0.00693

da=$\frac{dL(a,y)}{da}$ = $\frac{-y}{a}$ + $\frac{1-y}{1-a}$ = -1.006978

dw_1= $\frac{dL(a,y)}{dw_1}$ =x_1.dz=5*-0.00695=-0.03475

dw_2= $\frac{dL(a,y)}{dw_2}$ =x2.dz=3*-0.00695=-0.02085

db=dz

$w_1=w_1-\alpha dw_1$ = 0.5-(0.01*-0.03475)=0.503475

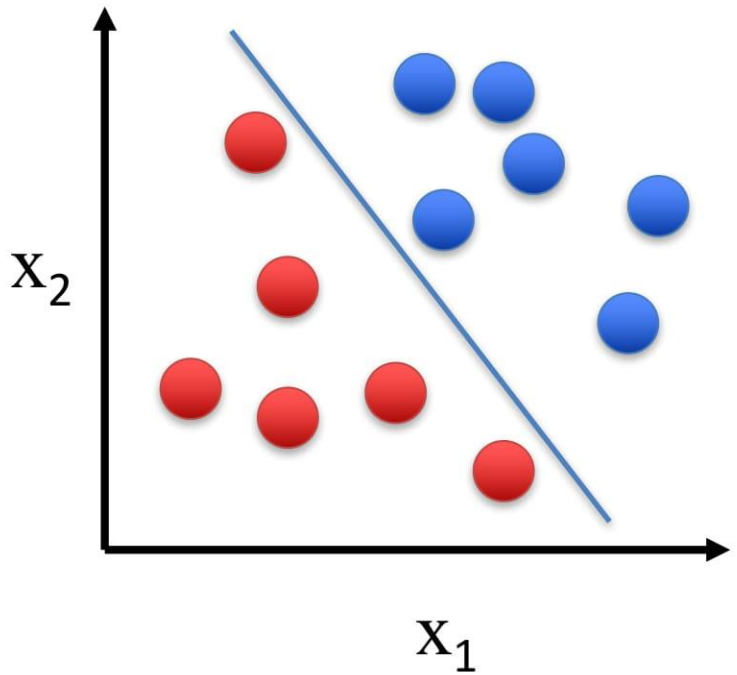
α= 0.01

$w_2=w_2-\alpha dw_2$ = 0.5-(0.01*-0.02085)=0.502085

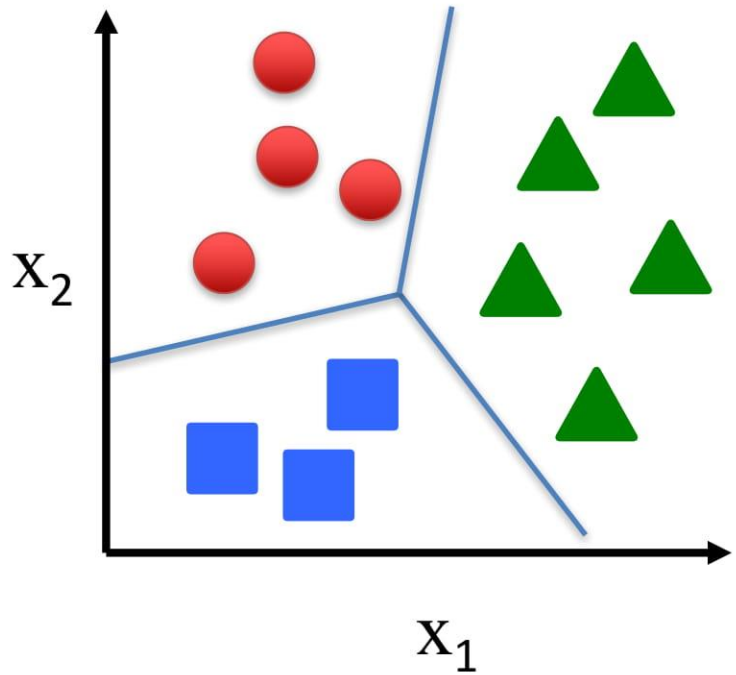
b=b- αdb =1-(0.01)*(-0.00693)=1.000693

Multi-Class Classification

Binary classification:



Multi-class classification:

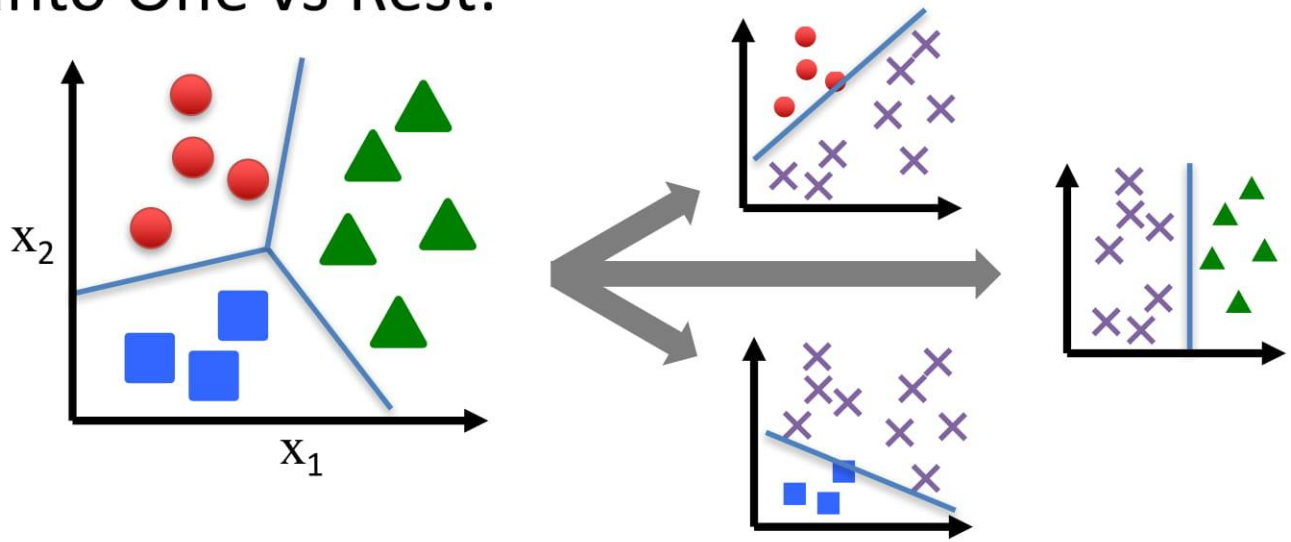


Disease diagnosis: healthy / cold / flu / pneumonia

Object classification: desk / chair / monitor / bookcase

Multi-Class Logistic Regression

Split into One vs Rest:



- Train a logistic regression classifier for each class i to predict the probability that $\mathrm{y} = \mathrm{i}$ with

$$h_c(\boldsymbol{x}) = \frac{\exp(\boldsymbol{\theta}_c^\mathsf{T} \boldsymbol{x})}{\sum_{c=1}^{C} \exp(\boldsymbol{\theta}_c^\mathsf{T} \boldsymbol{x})}$$

Softmax Function

When we have to classify in multiple categories then softmax function is useful. For example if you want to categorize pictures into

A) scene b) Animals c) Humans d) Vehicles then in that case we will have four outputs from the softmax function which will give us the probabilities of each of these categories.

Sum of the probabilities will be one and that with the highest probability will be shown as the answer.

Understanding softmax

$$z^{[L]} = \begin{pmatrix} 5 \\ 2 \\ -1 \\ 3 \end{pmatrix} \qquad t = \begin{pmatrix} e^5 \\ e^2 \\ e^{-1} \\ e^3 \end{pmatrix}$$

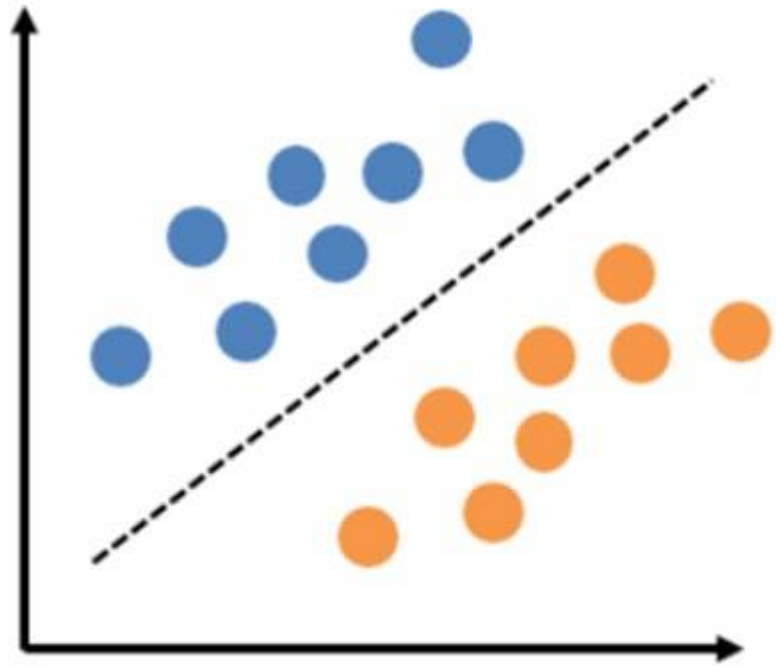
$$a^{[l]} = g^{[L]}(z^{[L]}) = \begin{pmatrix} e^5/(e^5+e^2+e^{-1}+e^3) \\ e^2/(e^5+e^2+e^{-1}+e^3) \\ e^{-1}/(e^5+e^2+e^{-1}+e^3) \\ e^3/(e^5+e^2+e^{-1}+e^3) \end{pmatrix} = \begin{pmatrix} 0.842 \\ 0.042 \\ 0.002 \\ 0.114 \end{pmatrix}$$

"Hard Max"

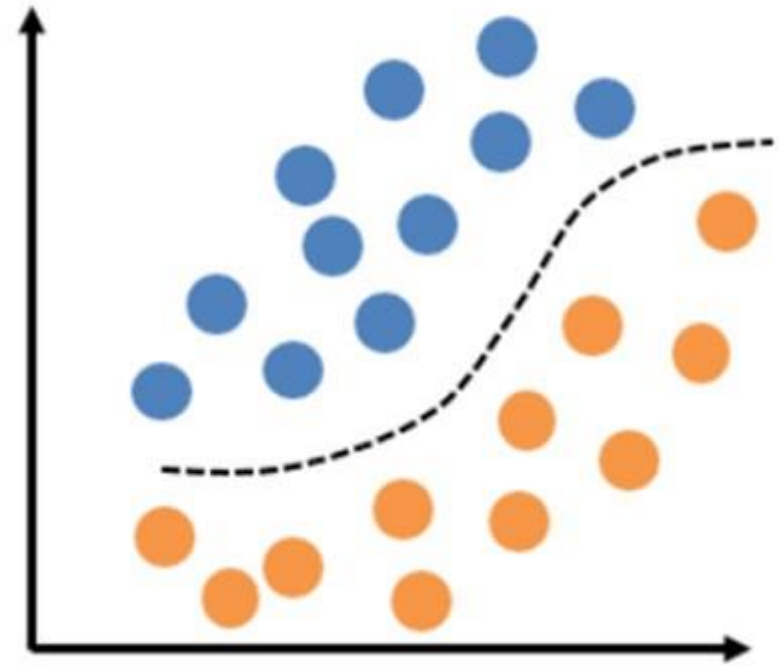
$$\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Softmax regression generalizes logistic regression to C classes. If c=2, softmax reduces to logistic regression.

Linear



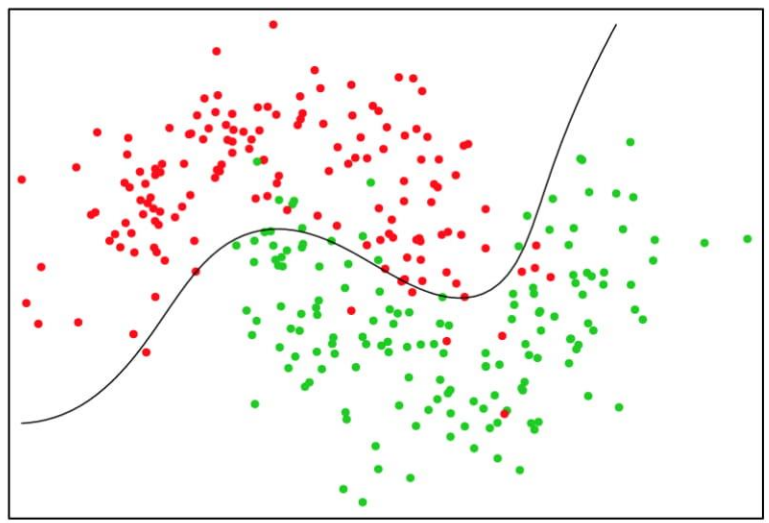
Nonlinear



Non-Linear Decision Boundary

- Can apply basis function expansion to features, same as with linear regression

$$\boldsymbol{x} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix} \rightarrow \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_1 x_2 \\ x_1^2 \\ x_2^2 \\ x_1^2 x_2 \\ x_1 x_2^2 \\ \vdots \end{bmatrix}$$



$$J(\hat{y}, y) = -((y \log \hat{y}) + (1-y)(\log(1-\hat{y})))$$

$$
\begin{aligned}
\frac{d}{dx}\sigma(x) &= \frac{d}{dx}\left[\frac{1}{1+e^{-x}}\right] \\
&= \frac{d}{dx}\left(1+\mathrm{e}^{-x}\right)^{-1} \\
&= -(1+e^{-x})^{-2}(-e^{-x}) \\
&= \frac{e^{-x}}{(1+e^{-x})^2} \\
&= \frac{1}{1+e^{-x}} \cdot \frac{e^{-x}}{1+e^{-x}}
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{1}{1+e^{-x}} \cdot \frac{(1+e^{-x})-1}{1+e^{-x}} \\
&= \frac{1}{1+e^{-x}} \cdot \left(\frac{1+e^{-x}}{1+e^{-x}} - \frac{1}{1+e^{-x}}\right) \\
&= \frac{1}{1+e^{-x}} \cdot \left(1 - \frac{1}{1+e^{-x}}\right) \\
&= \sigma(x) \cdot (1-\sigma(x))
\end{aligned}
$$