Logistic Regression

Giuseppe Serra

University of Udine

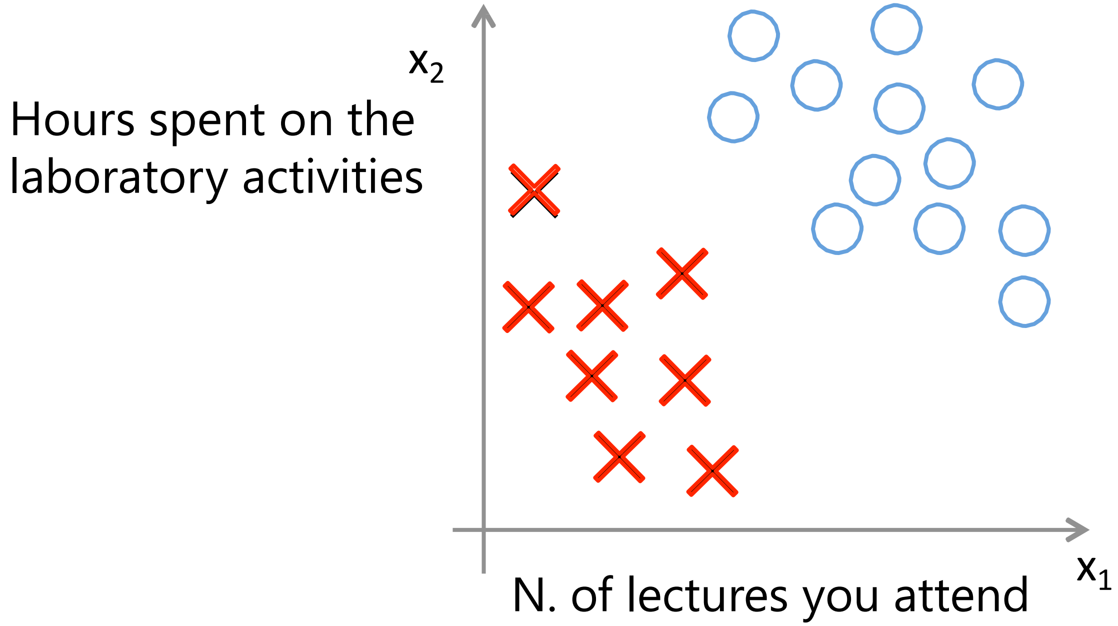
Example Problem: Will I Pass this Class?

Let's start with a simple two feature model:

x_1 = Number of lectures you attend

x_2 = Hours spent on the laboratory's activities

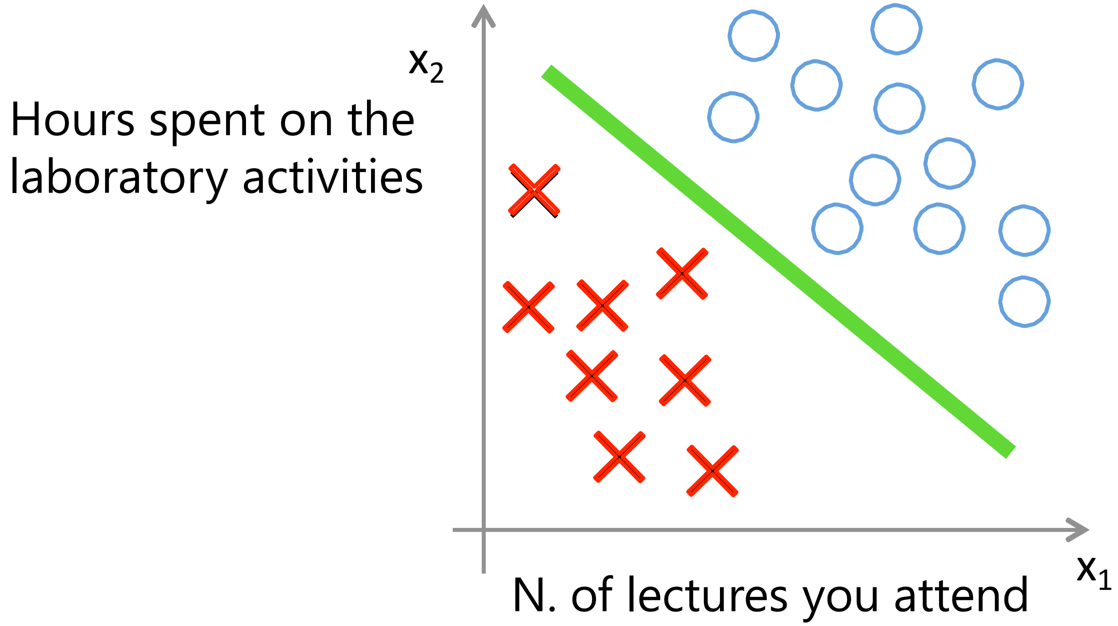
Example Problem: Will I Pass this Class?



O corresponds to “Pass”

X corresponds to “Fail”

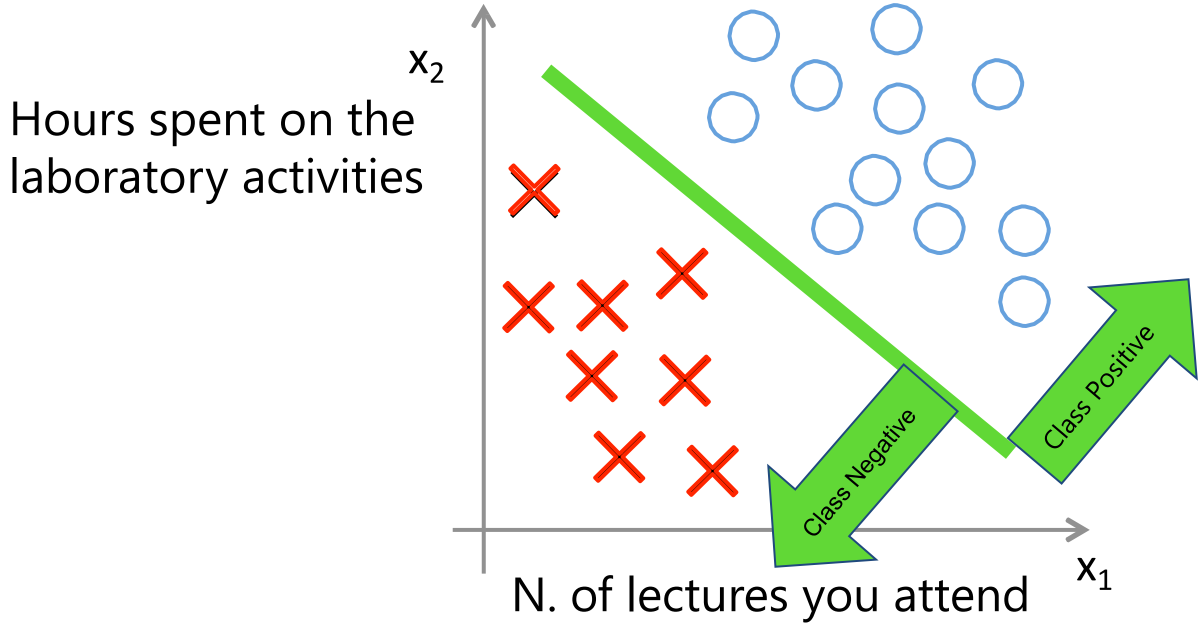
Example Problem: Will I Pass this Class?



O corresponds to “Pass”

X corresponds to “Fail”

Example Problem: Will I Pass this Class?



O corresponds to “Pass”

X corresponds to “Fail”

Logistic Regression

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

In our case $x^{(i)}$ is equal to the i-th John and Mary' ratings and $y^{(i)}$ is the i-th label (positive/negative)

With Logistic Regression we want to learn a probabilistic function that $\hat{y} = P(y = 1|x)$

In particular, **the goal is to find the parameters w and b** of the following function (hypothesis):

$h_{w,b}(x) = g\left(w^T x + b\right) = \frac{1}{1+e^{-(w^T x + b)}}$, where $g(z)$ is the Sigmoid function,

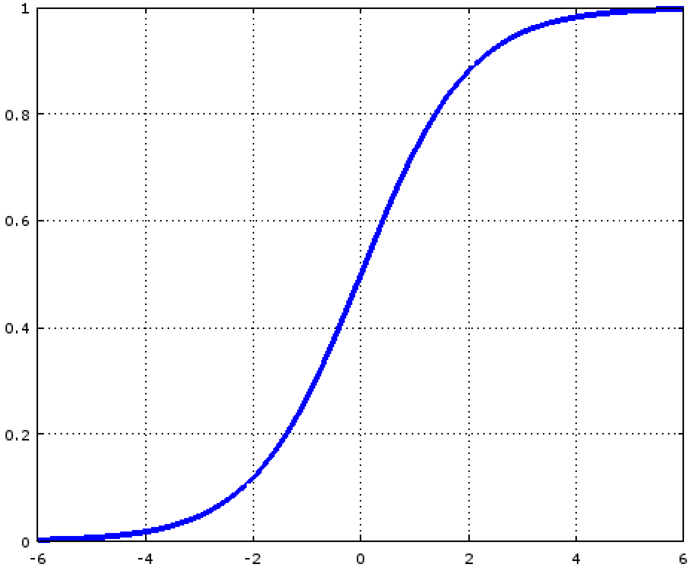
so that $\begin{cases} h_{w,b}(x) \geq 0.5 & if \quad y = 1 \\ h_{w,b}(x) < 0.5 & if \quad y = 0 \end{cases}$

To get our discrete 0 or 1 classification, we map the output of the hypothesis function as follo

$h_{w,b}(x) \geq 0.5 \ \rightarrow " \ 1"$

$h_{w,b}(x) < 0.5 \ \rightarrow \ " \ 0"$

Sigmoid Function



Linear decision boundaries

Let's consider a 2D case $h_{\mathrm{w,b}}(x) = g(b + w_1 x_1 + w_2 x_2)$

with $\mathrm{b}; \mathrm{w} = [w_1, w_2]^T, x = [x_1, x_2]^T$

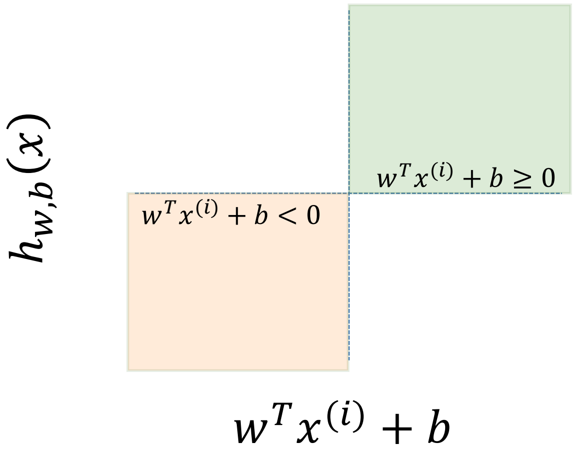
Using the Logic Regression Algorithm, we can obtain

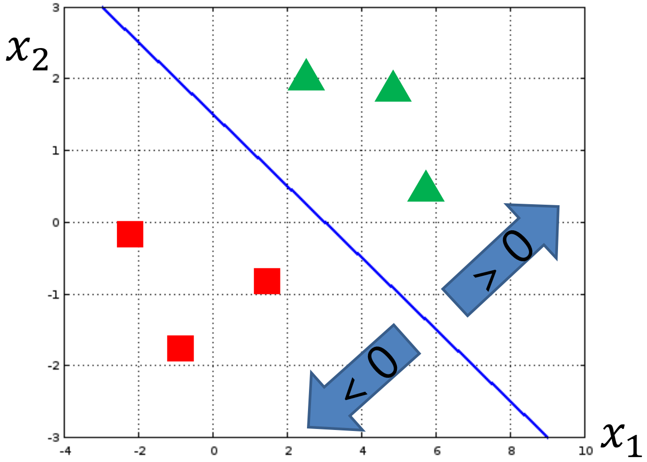
$\mathrm{b} = 3; \mathrm{w} = [\,1, 2]$

Note: $h_{w,b}\ (x) = g\ (w^T x + b) > 0.5$ when $w^T x + b > 0$

Then the decision boundary is

$h_{w,b}\ (x) = 0.5\ \Rightarrow w^T x + b = 0 \Rightarrow -3 + x_1 + 2x_2 = 0$





Decision boundary ($w^T x + b = 0$)

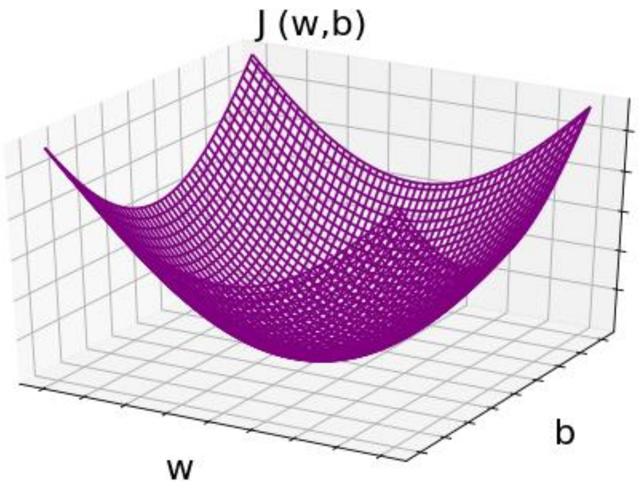
Cost Function

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

To find w and b so $\begin{cases} h_{w,b}(x) \geq 0.5 & if \quad y = 1 \\ h_{w,b}(x) < 0.5 & if \quad y = 0 \end{cases}$

The Logistic Classifier defines the following cost function:

- $J(w, b) = \frac{1}{m} \sum_{i=1}^{m} \text{Cost}\ (h_{w,b}(x^{(i)}), y^{(i)})$
  where $\text{Cost}\ (h_{w,b}(\ x^{(i)}), y^{(i)}) = \ -y^{(i)} \ln\left(h_{w,b}(x^{(i)})\right) - (1 - y^{(i)}) \ln\left(1 - h_{w,b}(x^{(i)})\right)$



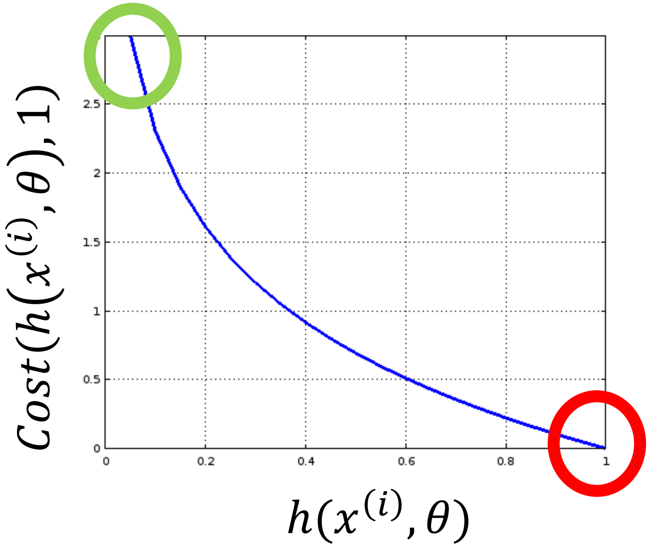
Note: This cost function (or loss) is convex and is derivable respect to w and b

Cost Function

$$\text{Cost}\left(h_{w,b}\left(x^{(i)}\right), y^{(i)}\right) = -y^{(i)}\ln(h_{w,b}\left(x^{(i)}\right)) - \left(1 - y^{(i)}\right)\ln\left(1 - h_{w,b}\left(x^{(i)}\right)\right)$$

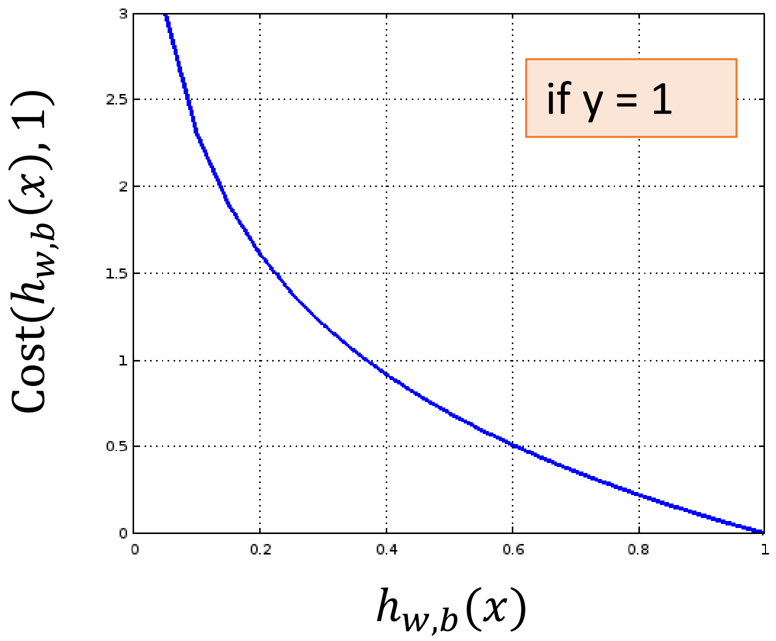
If $\mathrm{y}^{(i)} = 1 \Rightarrow \text{Cost}\left(h_{w,b}\left(x^{(i)}\right), y^{(i)}\right) = -\ln\left(h_{w,b}\left(x^{(i)}\right)\right)$

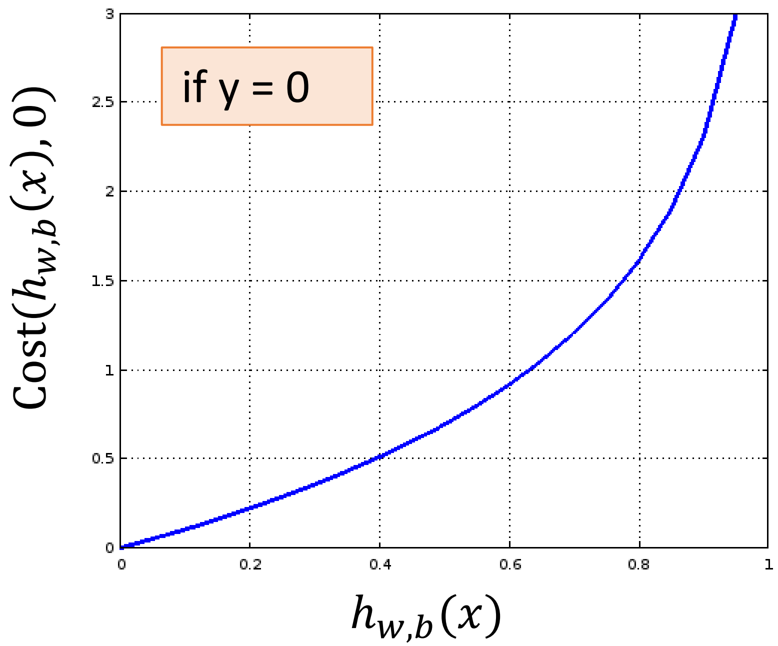
True Label $y^{(i)}$	Prediction $h_{w,b}\left(x^{(i)}, \theta\right)$	Cost $\text{Cost}\left(h_{w,b}\left(x^{(i)}\right), y^{(i)}\right)$
1	~ 1	~ 0
1	~ 0	Inf



Cost Function

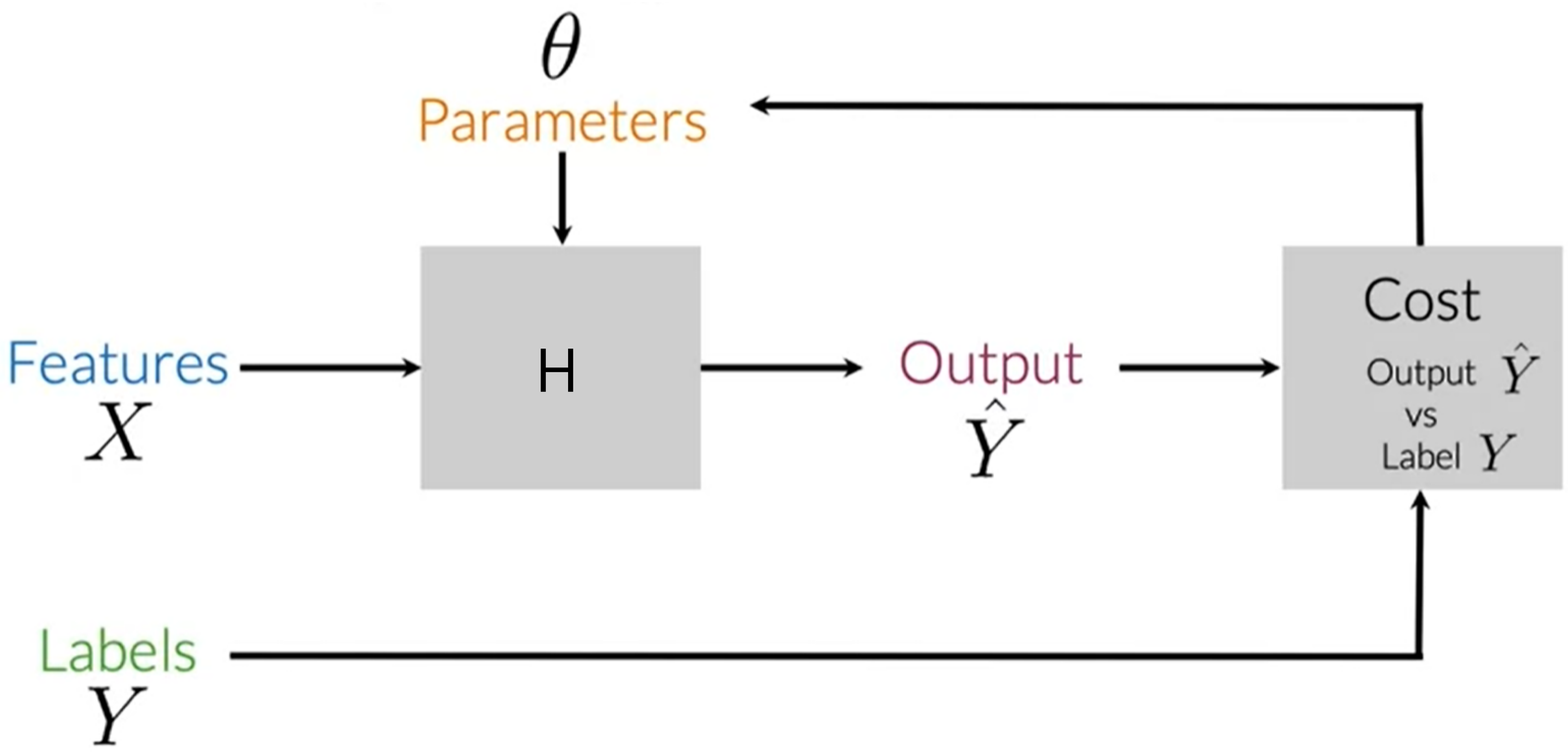
$$\text{Cost}\left(h_{w,b}\left(x^{(i)}\right), y^{(i)}\right) = -y^{(i)} \ln(h_{w,b}\left(x^{(i)}\right)) - \left(1 - y^{(i)}\right) \ln\left(1 - h_{w,b}\left(x^{(i)}\right)\right)$$





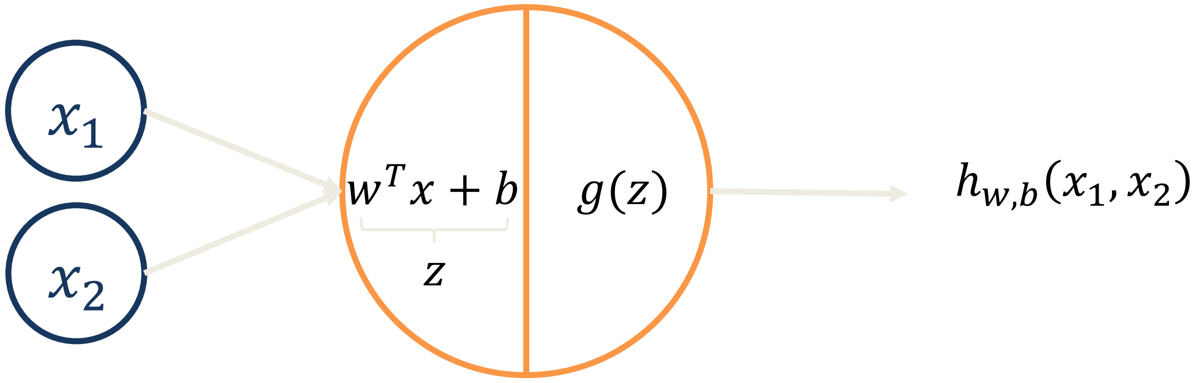
Case y=1, the cost function will be 0 if our hypothesis function $h_{w,b}(x)$ outputs 1. if our hypothesis approaches 0, then the cost function will approach infinity.

Classification: Logistic Regression



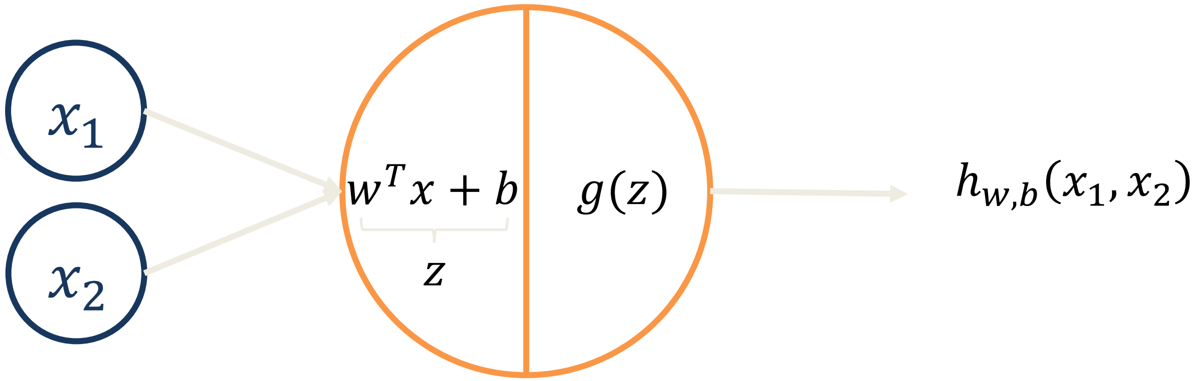
Logistic Regression Representation

This model means that to compute the value of the decision function, we need to multiply first input x_1 with θ_1, second input x_2 with θ_2, then add two values and b, then apply the sigmoid function



Neural Network Representation

This network means that to compute the value of the decision function, we need to multiply first input x_1 with θ_1, second input x_2 with θ_2, then add two values and b, then apply the sigmoid function

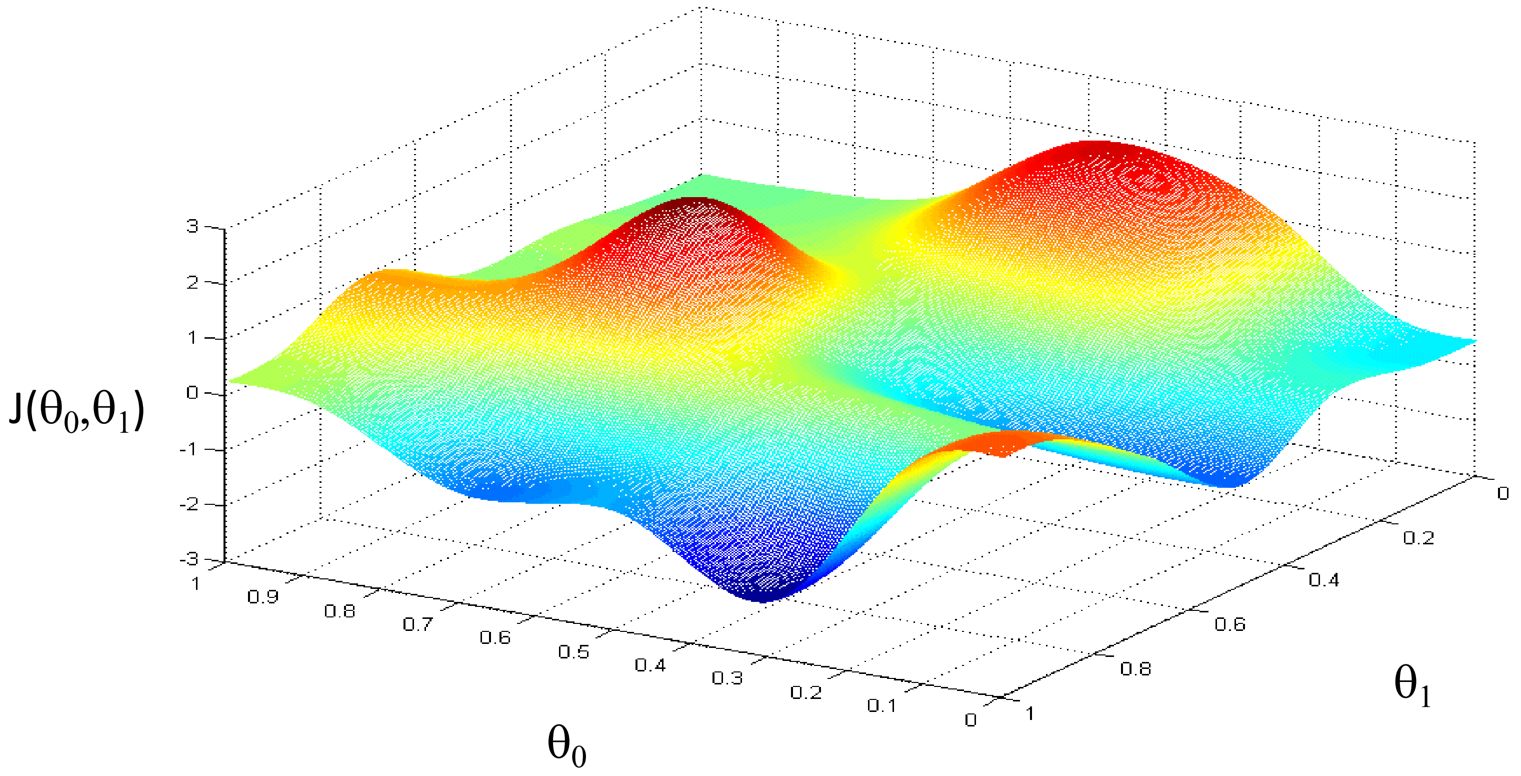


Gradient Descent

Optimization

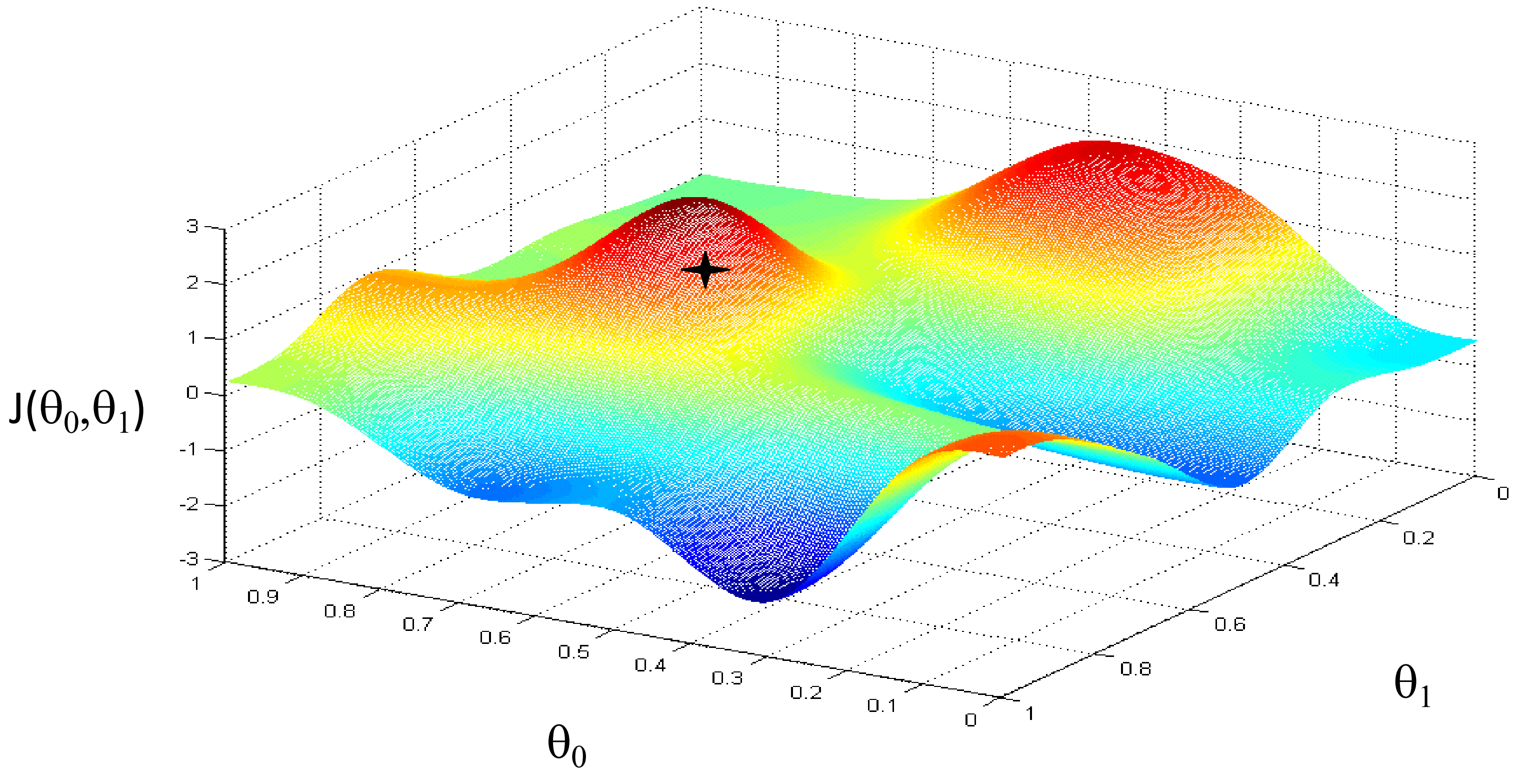
We want to find the parameters that achieve the lowest cost (or loss).

Here an example of a general and no convex cost function



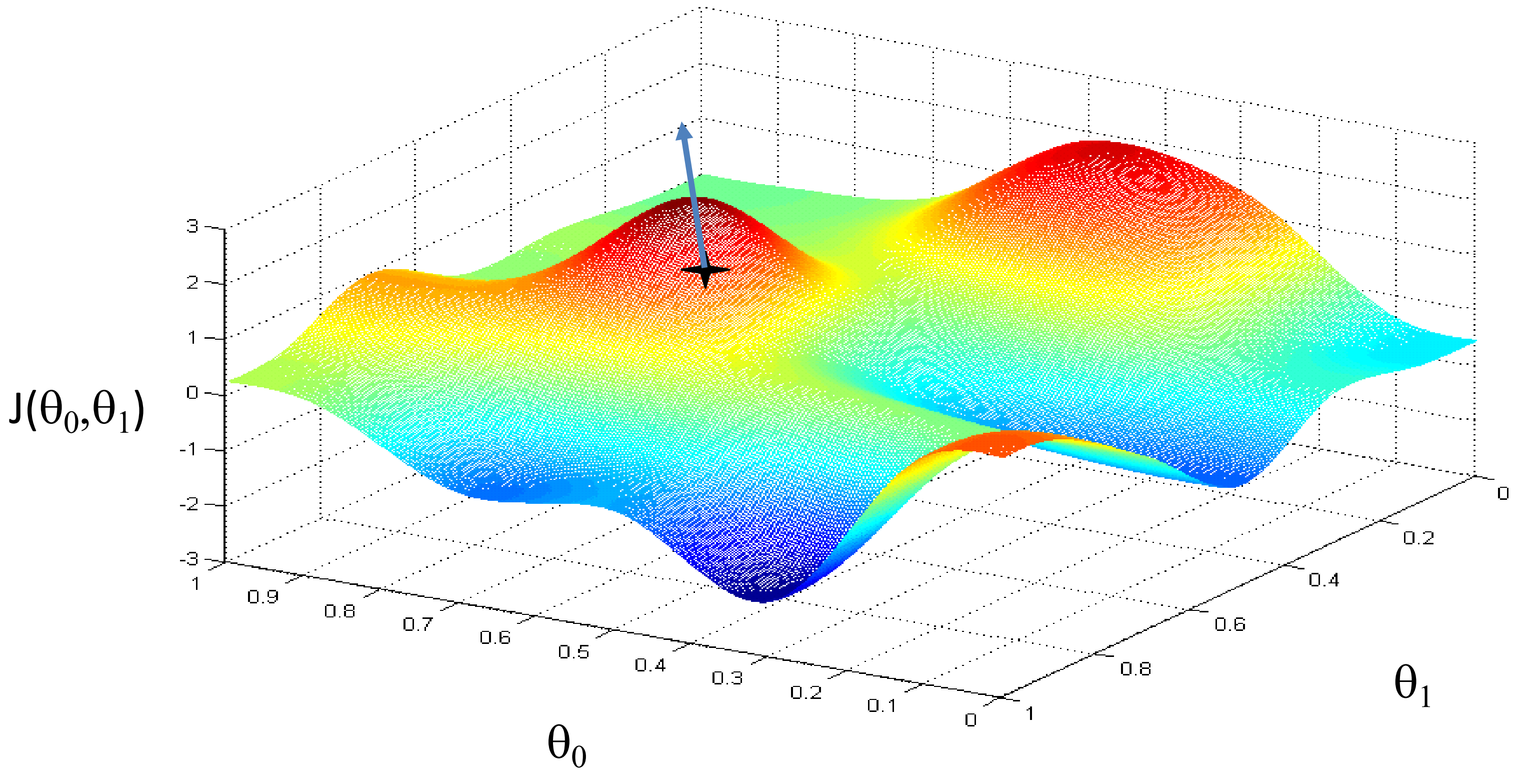
Optimization

Randomly pick initial values θ_0 e θ_1



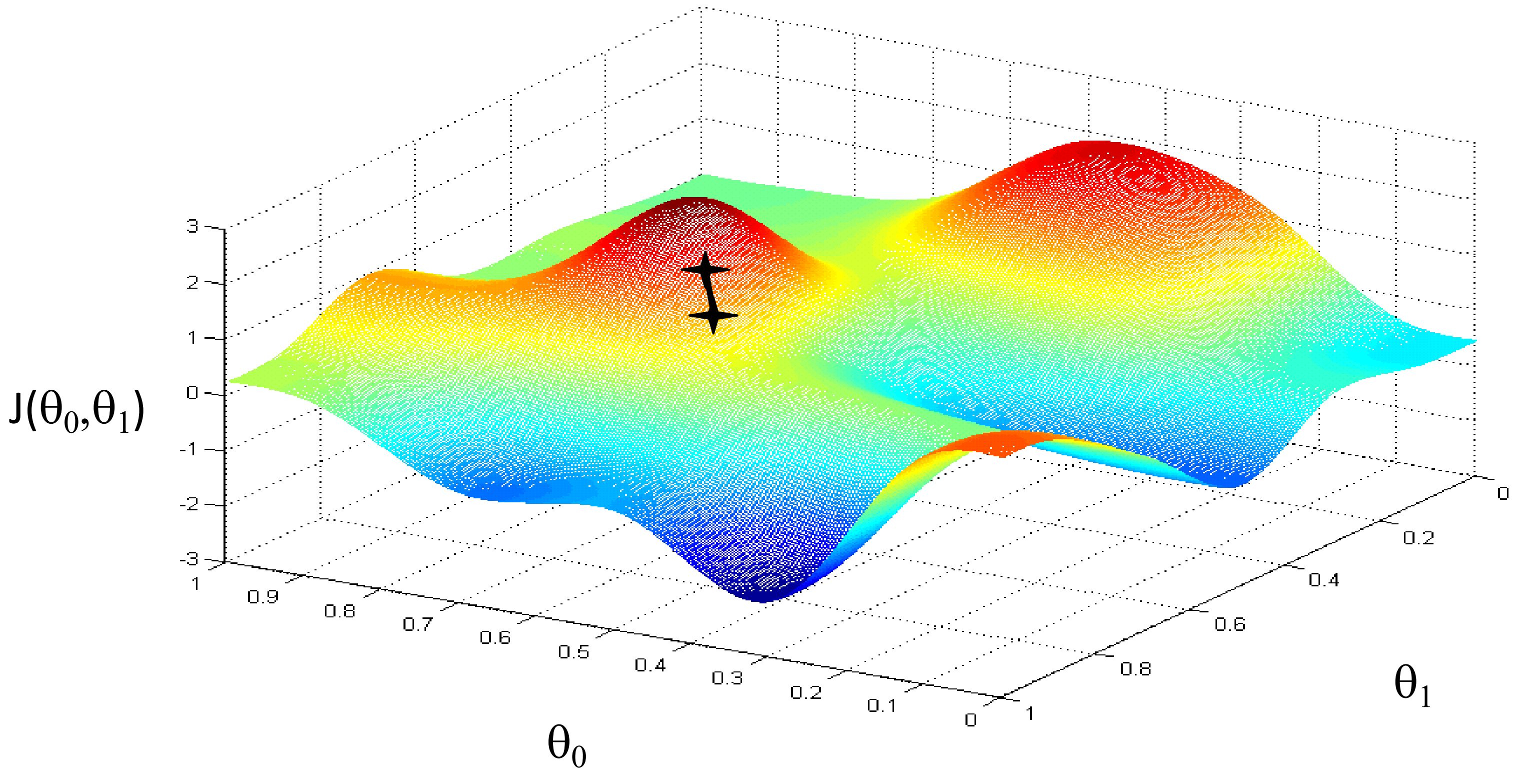
Optimization

Compute the gradient: $\frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$



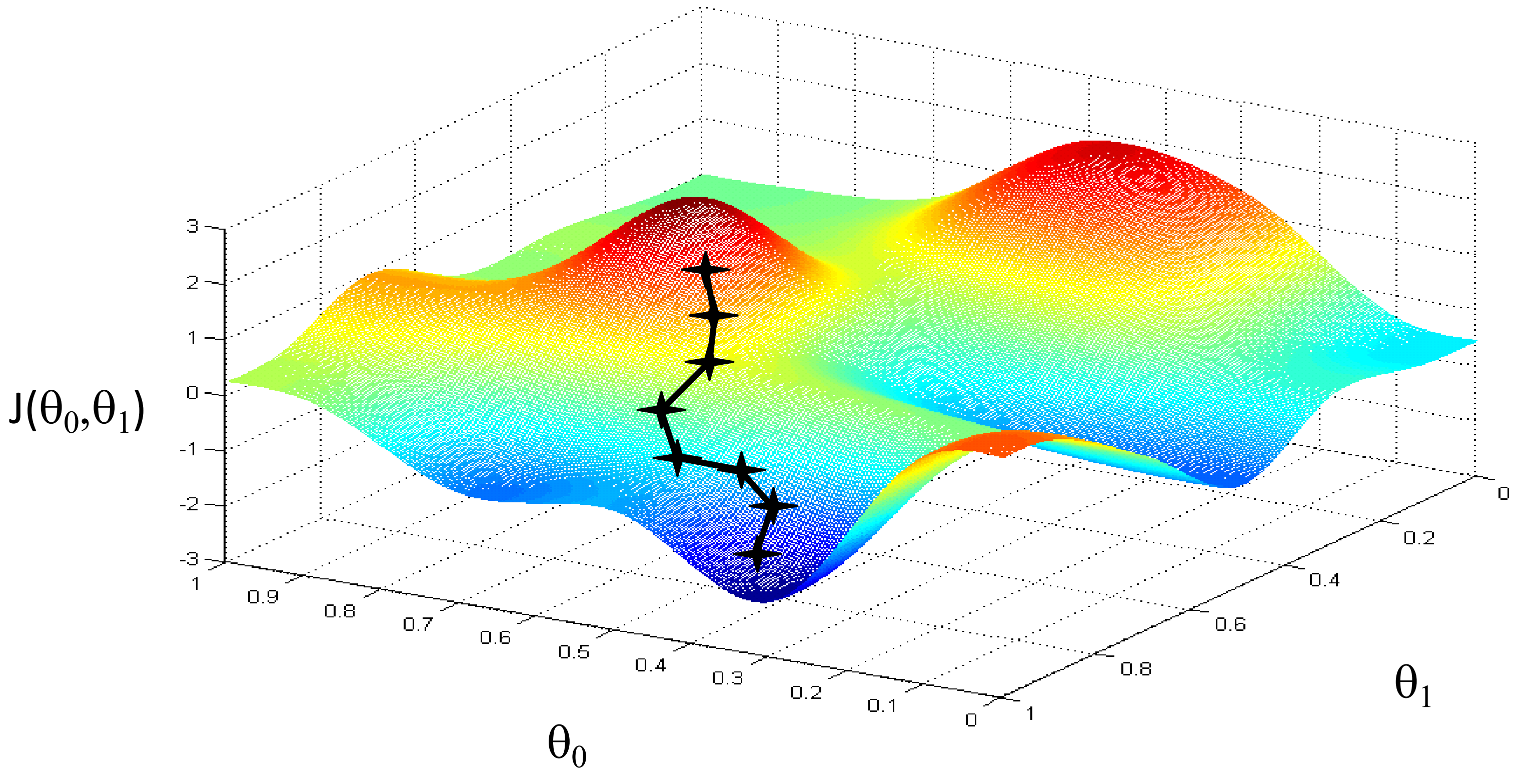
Optimization

Take small step in opposite direction of gradient

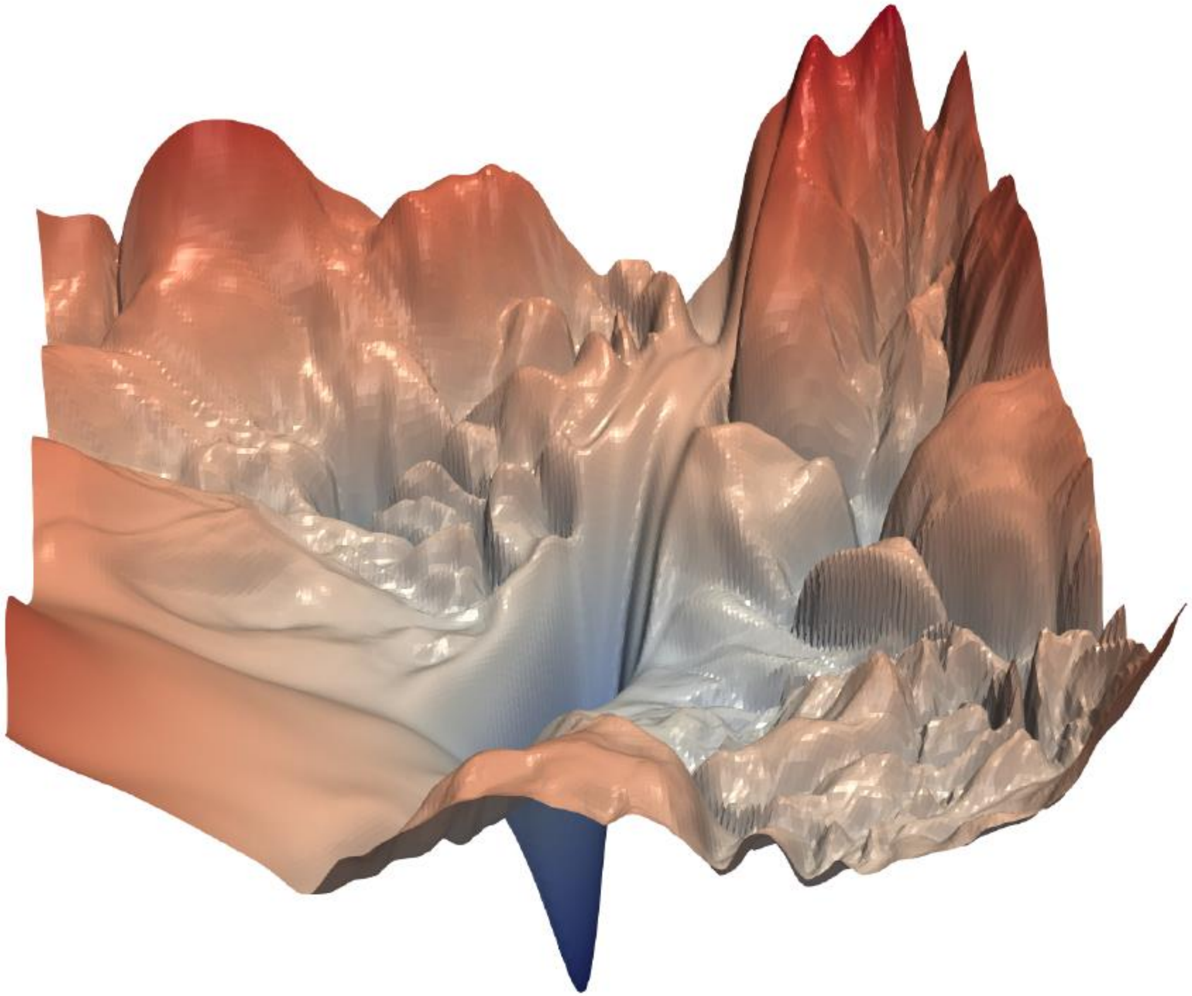


Optimization

Repeat until convergence



Training Neural Networks is Difficult



"Visualizing the loss landscape of neural networks". NIPS Dec 2017.

Gradient Descent Algorithm

GD is a general algorithm to minimize derivable function. Here we are using to linear regression.

Have some function $J(\theta_0, \dots, \theta_n)$

Want $\min\limits_{\theta_0, \dots \theta_n} \; J(\theta_0, \dots, \theta_n)$

Outline:

- Start with some $\theta_0, \dots, \theta_n$ (common choice: random)
- Keep changing $\theta_0, \dots, \theta_n$ to reduce $J(\theta_0, \dots, \theta_n)$
- Until we hopefully end up at a minimum

Gradient Descent Algorithm

$$
\begin{aligned}
&\text{repeat until convergence } \{ \\
&\quad \theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \qquad (\text{for } j = 0 \text{ and } j = 1) \\
&\}
\end{aligned}
$$

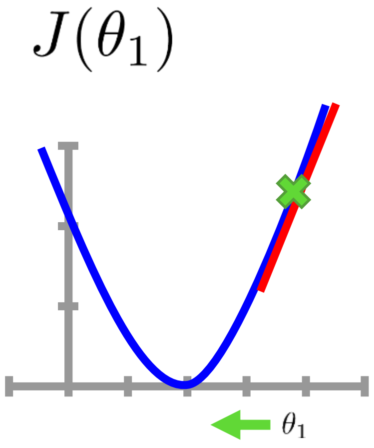
Correct: Simultaneous update

$\text{temp0} := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$
$\text{temp1} := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$
$\theta_0 := \text{temp0}$
$\theta_1 := \text{temp1}$

Incorrect:

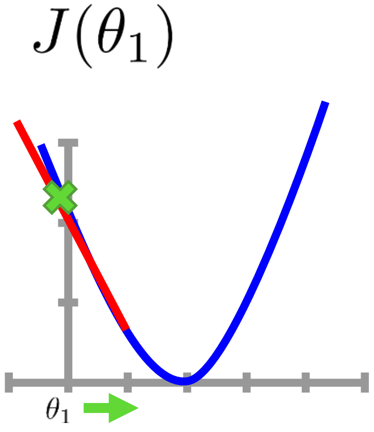
$\text{temp0} := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$
$\theta_0 := \text{temp0}$
$\text{temp1} := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$
$\theta_1 := \text{temp1}$

Gradient Descent Intuition



$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)$$

$$\frac{\partial}{\partial \theta_1} J(\theta_1) > 0$$

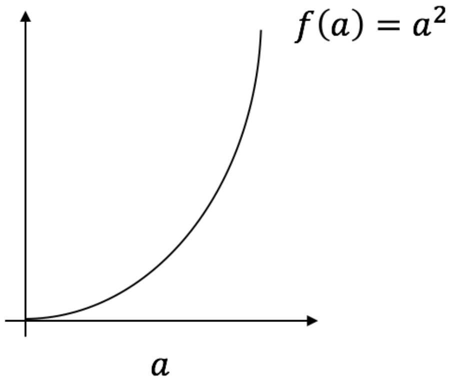


$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)$$

$$\frac{\partial}{\partial \theta_1} J(\theta_1) < 0$$

Note: The gradient of a function represent how small changes on a parameter $\theta_1$will affect the cost function $J(\theta_1)$

For example:

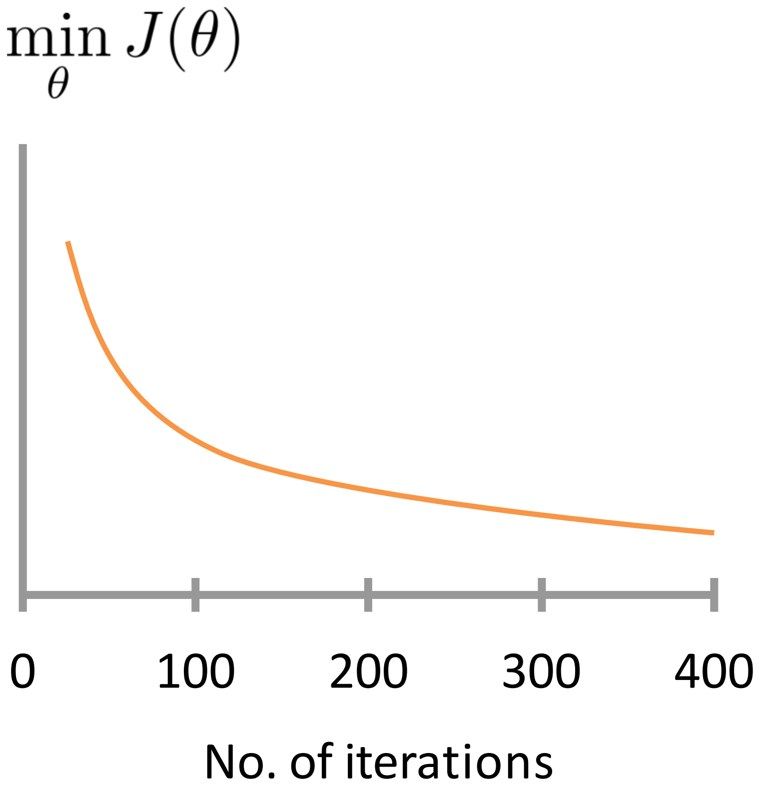


$$\frac{d\,f(a)}{da} = 2a$$

Slope (derivative) of $f(a)$ at $a = 5$ is 10

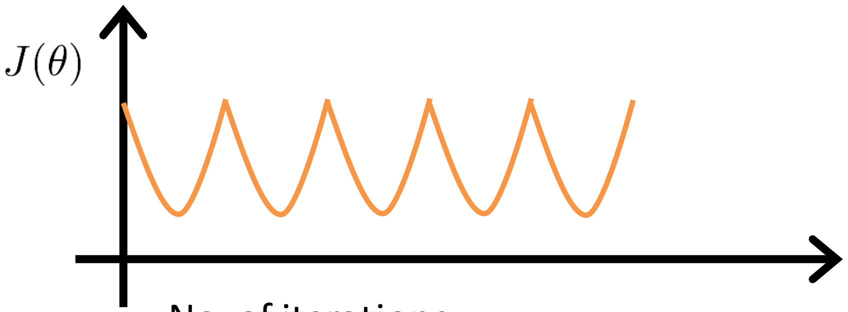
$a = 5$	$f(5) = 25$
$a = 5.001$	$f(5.001) \cong 25.010$

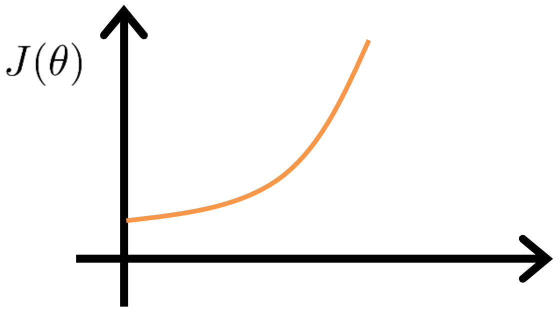
Gradient Descent working correclty?



Gradient Descent working correclty?

- For sufficiently small α, $J(\theta)$, should decrease on every iterations
- But if α is too small, gradient descent can be slow to converge
- If α is too large: $J(\theta)$ may not decrease on every iterations; may not converge
- Here some examples:





To choose α try: $\ldots, 0.001, 0.003, 0.01, 0.03, 0.1, 0.3, 1, \ldots$

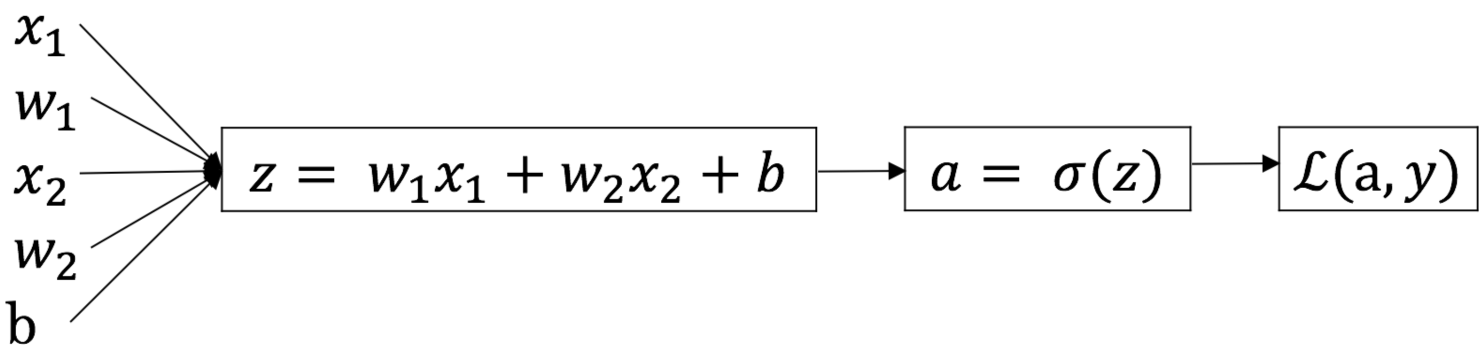
Logistic Regression - Gradient Descent

Logistic Regression Gradient Descent

Logistic Equations for one example:

$z = w^T x + b$
$a = \sigma(z)$
$Cost(\mathrm{a},\mathrm{y}) = \mathcal{L}(a, y) = -(yln(a) + (1 - y)\ln(1 - a))$

Using computational graph, we can represent as following:



Remember:

- we want to change parameters to reduce the loss.

- We can use the Gradient Descent Algorithm. Therefore, we need to compute derivatives…

Derivatives

$f(x)$	$f'(x)$	$f(x)$	$f'(x)$
x^n	nx^{n-1}	e^x	e^x
$\ln(x)$	$1/x$	$\sin(x)$	$\cos(x)$
$\cos(x)$	$-\sin(x)$	$\tan(x)$	$\sec^2(x)$
$\cot(x)$	$-\text{cosec}^2(x)$	$\sec(x)$	$\sec(x)\tan(x)$
$\text{cosec}(x)$	$-\text{cosec}(x)\cot(x)$	$\tan^{-1}(x)$	$1/(1+x^2)$
$\sin^{-1}(x)$	$1/\sqrt{1-x^2}$ for $\lvert x\rvert < 1$	$\cos^{-1}(x)$	$-1/\sqrt{1-x^2}$ for $\lvert x\rvert < 1$
$\sinh(x)$	$\cosh(x)$	$\cosh(x)$	$\sinh(x)$
$\tanh(x)$	$\text{sech}^2(x)$	$\coth(x)$	$-\text{cosech}^2(x)$
$\text{sech}(x)$	$-\text{sech}(x)\tanh(x)$	$\text{cosech}(x)$	$-\text{cosech}(x)\coth(x)$
$\sinh^{-1}(x)$	$1/\sqrt{x^2+1}$	$\cosh^{-1}(x)$	$1/\sqrt{x^2-1}$ for $x > 1$
$\tanh^{-1}(x)$	$1/(1-x^2)$ for $\lvert x\rvert < 1$	$\coth^{-1}(x)$	$-1/(x^2-1)$ for $\lvert x\rvert > 1$

Derivative of Sigmoid Function

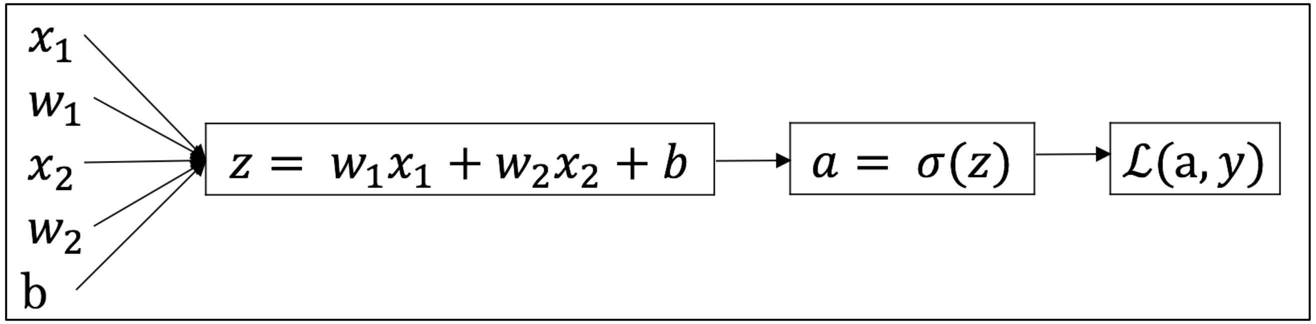
Let's denote the sigmoid function as $\sigma(x) = \frac{1}{1+e^{-x}}$

Its derivative is:

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

Logistic Regression derivatives

$z = w^T x + b$
$a = \sigma(z)$
$Cost(\mathrm{a}, \mathrm{y}) = \mathcal{L}(a, y) = -(y ln(a) + (1-y)\ln(1-a))$



Derivatives:

$$\frac{d\mathcal{L}}{da} = -\frac{y}{a} + \frac{1-y}{1-a}$$

$$\frac{d\mathcal{L}}{dz} = \frac{d\mathcal{L}}{da}\frac{da}{dz} = \left(-\frac{y}{a} + \frac{1-y}{1-a}\right)(a(1-a)) = a - y$$

$$\frac{d\mathcal{L}}{dw_1} = \frac{d\mathcal{L}}{da}\frac{da}{dz}\frac{dz}{dw_1} = (a-y)x_1$$

$$\frac{d\mathcal{L}}{dw_2} = \frac{d\mathcal{L}}{da}\frac{da}{dz}\frac{dz}{dw_2} = (a-y)x_2$$

$$\frac{d\mathcal{L}}{db} = \frac{d\mathcal{L}}{da}\frac{da}{dz}\frac{dz}{db} = (a-y)$$

Gradient Descent Algorithm for Logistic Regression:

$$w_1 \coloneqq w_1 - \alpha\frac{d\ J(w,b)}{dw_1}$$
$$w_2 \coloneqq w_2 - \alpha\frac{d\ J(w,b)}{dw_2}$$
$$b \coloneqq b - \alpha\frac{d\ J(w,b)}{db}$$

Logistic Regression on *m* examples

$$J(w,b) = \frac{1}{m}\sum_{i=1}^{m}\mathcal{L}(a^{(i)}, y^{(i)})$$

where:

$\mathcal{L}\left(a^{(i)}, y^{(i)}\right) = -\left(y^{(i)} ln\left(a^{(i)}\right) + \left(1 - y^{(i)}\right)\ln\left(1 - a^{(i)}\right)\right)$

$\mathrm{a}^{(i)} = \sigma\left(z^{(i)}\right)$

$z^{(i)} = w^T x^{(i)} + b$

$$\frac{d\ J(w,b)}{dw_1} = \frac{1}{m}\sum_{i=1}^{m}\frac{d\mathcal{L}(a^{(i)}, y^{(i)})}{dw_1} = \frac{1}{m}\sum_{i=1}^{m}\left(a^{(i)} - y^{(i)}\right)x_1^{(i)}$$

$$\frac{d\ J(w,b)}{dw_2} = \frac{1}{m}\sum_{i=1}^{m}\frac{d\mathcal{L}(a^{(i)}, y^{(i)})}{dw_1} = \frac{1}{m}\sum_{i=1}^{m}\left(a^{(i)} - y^{(i)}\right)x_2^{(i)}$$

$$\frac{d\ J(w,b)}{db} = \frac{1}{m}\sum_{i=1}^{m}\frac{d\mathcal{L}(a^{(i)}, y^{(i)})}{dw_1} = \frac{1}{m}\sum_{i=1}^{m}\left(a^{(i)} - y^{(i)}\right)$$

Gradient Descent Algorithm for Logistic Regression:

$$w_1 \coloneqq w_1 - \alpha\frac{d\ J(w,b)}{dw_1}$$

$$w_2 \coloneqq w_2 - \alpha\frac{d\ J(w,b)}{dw_2}$$

$$b \coloneqq b - \alpha\frac{d\ J(w,b)}{db}$$

Remember: the sum rule for derivatives states that the derivative of a sum is equal to the sum of the derivatives.