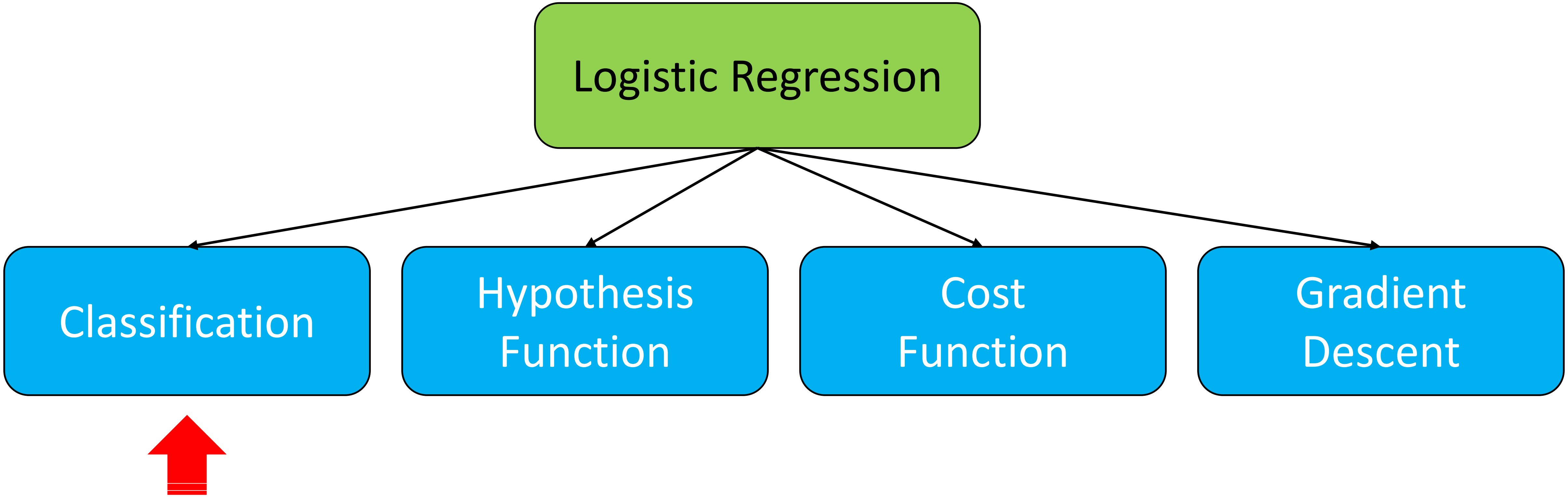
Perceptron, Logistic Regression, and Neural Networks

Outline


Logistic Regression
Classification
Hypothesis Function
Cost Function
Gradient Descent

Example 1: Malignant or Benign

- Consider the example of recognizing whether a tumor in an input image is malignant or benign

Input:

Can be represented as a matrix of pixels

Binary Classifier

Output:

Benign **0**

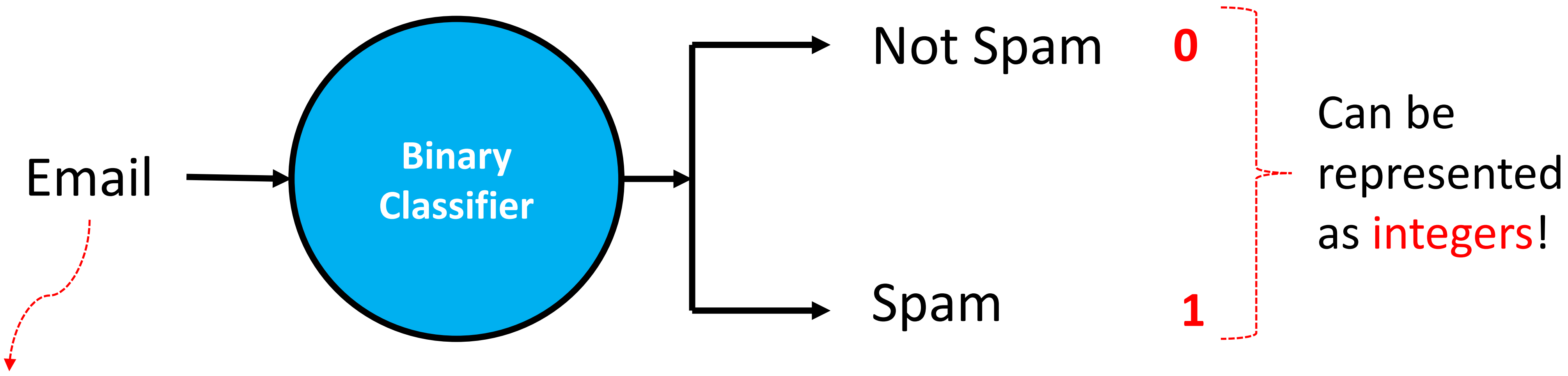
Malignant **1**

Can be represented as integers!

Example 2: Spam or Not Spam

- As another example, consider the problem of detecting whether an email is a spam or not a spam

Input: **Output:**



Can be represented as a vector $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_d]$, with each component $\mathbf{x}_i$ corresponding to the presence ($\mathbf{x}_i = 1$) or absence ($\mathbf{x}_i = 0$) of a particular <u>*word*</u> (or *feature*) in the email

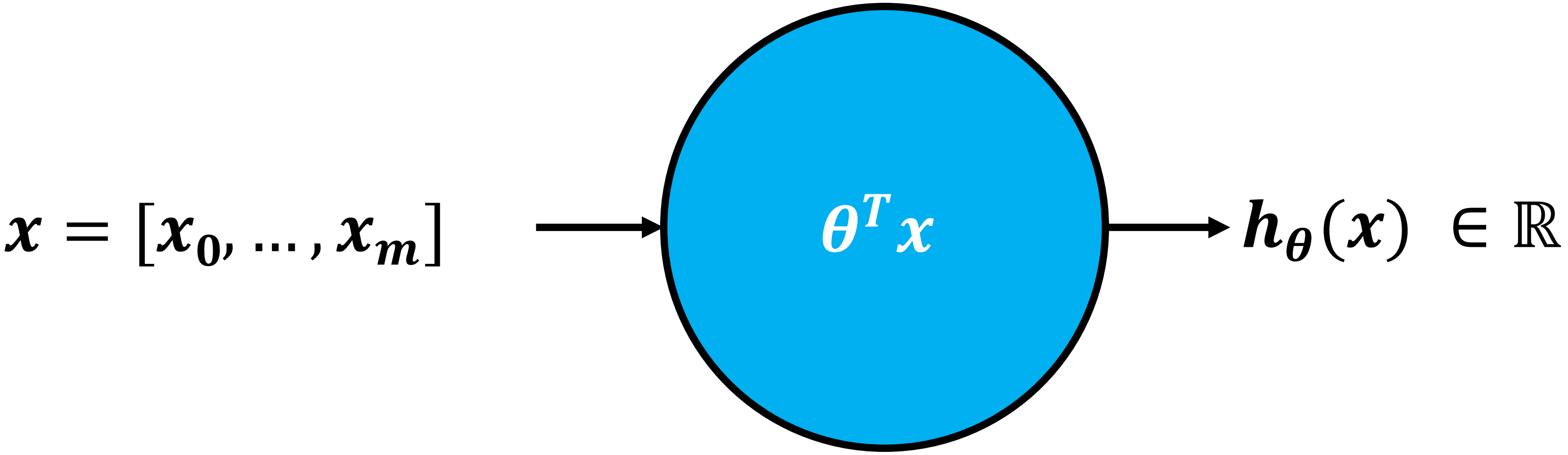
Regression vs. Classification

- What are the possible output values of the ***linear regression model*** $\boxed{\boldsymbol{h_\theta(x) = \theta^T x}}$?

$\boldsymbol{\theta}$ is the ***parameter vector***, that is, $\boldsymbol{\theta}$ = [$\boldsymbol{\theta_0}$, $\boldsymbol{\theta_1}$, ..., $\boldsymbol{\theta_m}$] (assuming $m + 1$ parameters) and $\boldsymbol{x}$ is the ***feature vector***, that is, $\boldsymbol{x}$ = [$\boldsymbol{x_0}$, $\boldsymbol{x_1}$, ..., $\boldsymbol{x_m}$] (assuming $m + 1$ features and $\boldsymbol{x_0} = 1$ to account for the intercept term, namely, $\boldsymbol{\theta_0}$)

Regression vs. Classification

- What are the possible output values of the ***linear regression model*** $\boldsymbol{h_\theta(x) = \theta^T x}$?
  - Real-valued outputs



Regression vs. Classification

- What are the possible output values of the ***linear regression model*** $\boldsymbol{h_\theta(x) = \theta^T x}$?
  - Real-valued outputs

$$\boldsymbol{x} = \underbrace{[\mathbf{1, 0, 0, 1, 0}]}_{\text{Representing an email}} \longrightarrow$$

Spam Filter

$\boldsymbol{\theta^T x}$

$$\longrightarrow \boldsymbol{h_\theta(x)} = \begin{bmatrix} \mathbf{3.2} \\ \mathbf{4.1} \\ \mathbf{2.9} \\ \mathbf{6.7} \\ \mathbf{1.1} \end{bmatrix} [\mathbf{1, 0, 0, 1, 0}]$$

Assuming $\boldsymbol{\theta} = [\mathbf{3.2, 4.1, 2.9, 6.7, 1.1}]$

Regression vs. Classification

- What are the possible output values of the ***linear regression model*** $\boldsymbol{h_\theta(x) = \theta^T x}$?
    - Real-valued outputs

$\in \mathbb{R}$, which makes it a

regression problem

$$\boldsymbol{x = [1, 0, 0, 1, 0]} \longrightarrow \boxed{\textbf{\textit{Spam Filter}}\quad \boldsymbol{\theta^T x}} \longrightarrow \boldsymbol{h_\theta(x) = 3.2 + 6.7 = 9.9}$$

A spam or not a spam?

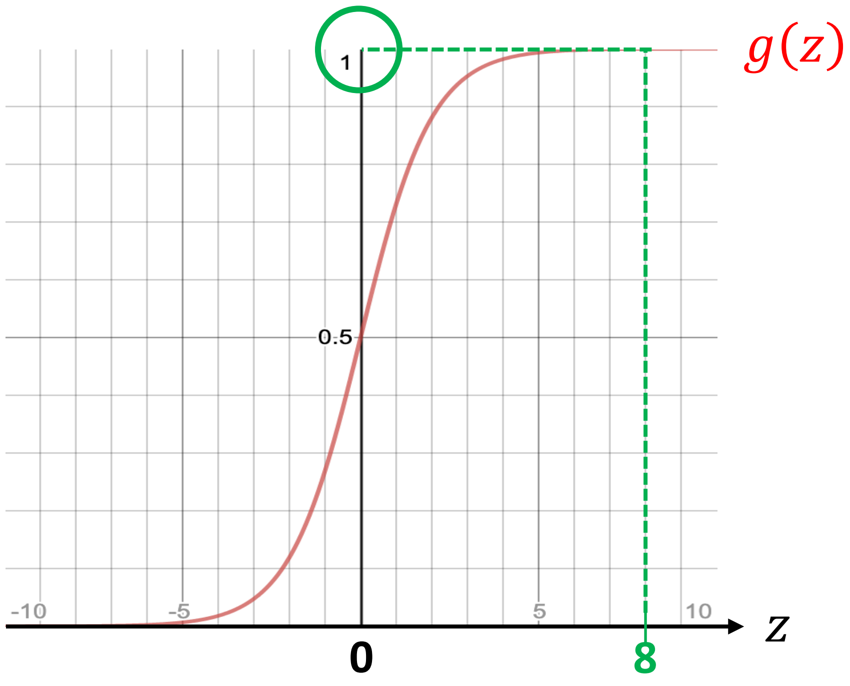
We need a *discrete-valued* output (e.g., 1 or 0)

Regression vs. Classification

- How can we make the possible outputs of $\boldsymbol{h_\theta(x)} = \boldsymbol{\theta^T x}$ discrete-valued (as opposed to real-valued)?
  - By using an ***activation function*** (e.g., ***sigmoid or logistic function***)

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

$z \in \mathbb{R}, but$
$g(z) \in [0,1]$



Assume a labeled example ($\boldsymbol{x}$, y):

If y = **1**, we want $g(z) \approx 1$ (i.e., we want a correct prediction)

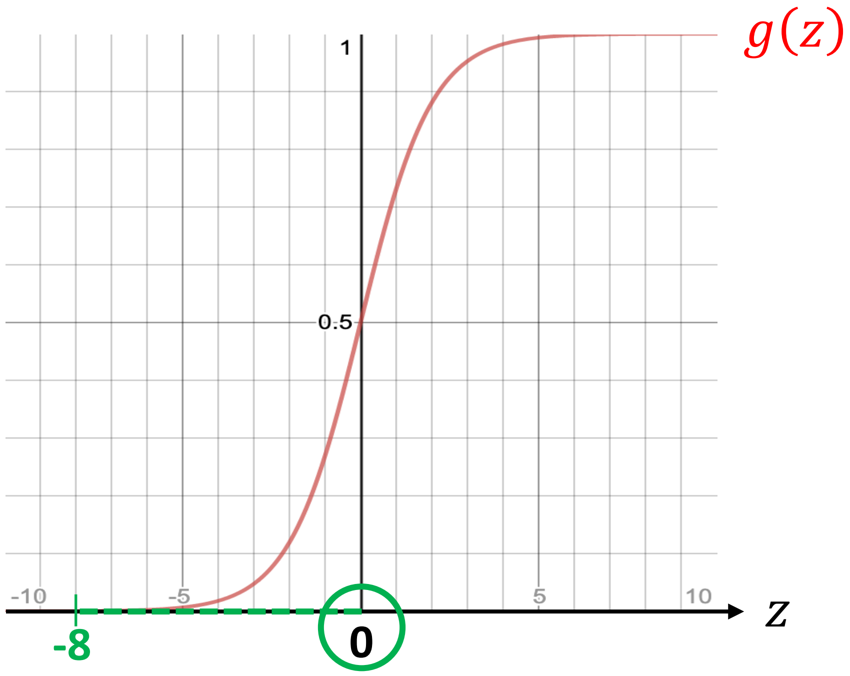
For this to happen, $\boldsymbol{z} \gg \boldsymbol{0}$

Regression vs. Classification

- How can we make the possible outputs of $\boldsymbol{h_\theta(x)} = \boldsymbol{\theta^T x}$ discrete-valued (as opposed to real-valued)?
  - By using an ***activation function*** (e.g., ***sigmoid or logistic function***)

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

$z \in \mathbb{R}, but$
$g(z) \in [0,1]$



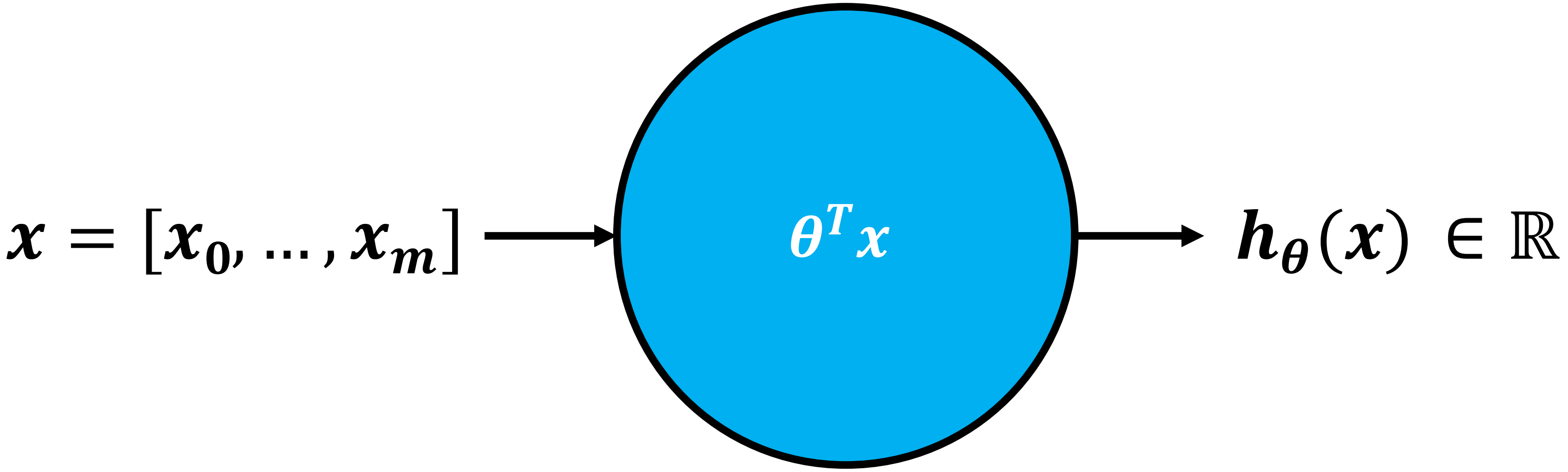
Assume a labeled example ($\boldsymbol{x}$, y):

If y = **0**, we want $g(z) \approx 0$ (i.e., we want a correct prediction)

For this to happen, $\boldsymbol{z \ll 0}$

Regression vs. Classification

- How can we make the possible outputs of $\boldsymbol{h_\theta(x) = \theta^T x}$ discrete-valued (as opposed to real-valued)?
    - By using an ***activation function*** (e.g., ***sigmoid or logistic function***)



Regression vs. Classification

- How can we make the possible outputs of $\boldsymbol{h_\theta(x)} = \boldsymbol{\theta^T x}$ discrete-valued (as opposed to real-valued)?
  - By using an ***activation function*** (e.g., ***sigmoid or logistic function***)

$$\boldsymbol{x} = [\boldsymbol{x_0}, \dots, \boldsymbol{x_m}] \longrightarrow g(\underbrace{\theta^T x}_{z}) \longrightarrow \boldsymbol{h_\theta(x)} \in [0,1]$$

We can now apply *thresholding*--
for example, if $\boldsymbol{h_\theta(x)} < 0.5$, predict 0;
otherwise, predict 1

Regression vs. Classification

- How can we make the possible outputs of $\boldsymbol{h_\theta(x) = \theta^T x}$ discrete-valued (as opposed to real-valued)?
  - By using an ***activation function*** (e.g., ***sigmoid or logistic function***)

$$\boldsymbol{x = [x_0, \ldots, x_m]} \longrightarrow \boldsymbol{g(\underbrace{\theta^T x}_{z})} \longrightarrow \boldsymbol{h_\theta(x)} \in [0,1]$$

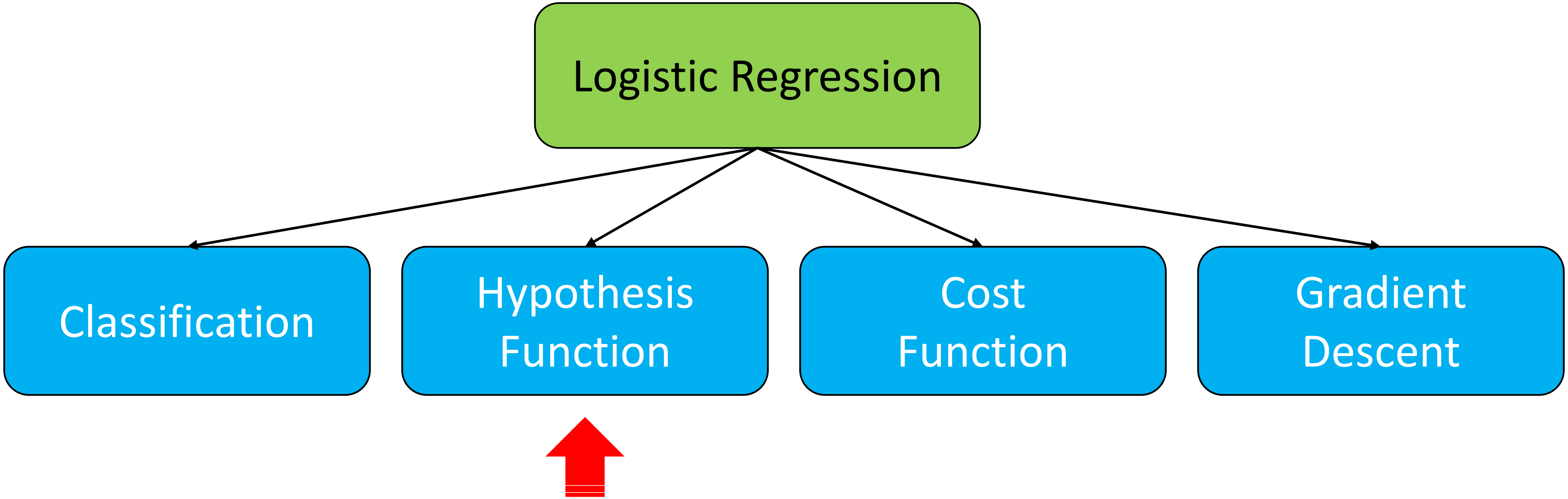
This makes it a *classification problem* (i.e., NOT a regression problem anymore) since the output renders a discrete one (*especially after applying thresholding*)

So, What is Logistic Regression?

- Logistic regression is a machine learning algorithm that can be used to *classify* input data into discrete output (e.g., *input emails into spam or non-spam and tumour images into benign or malignant*)
  - **Note**: The word "regression" in the name does not mean that the algorithm is a regression algorithm (rather it is a classification algorithm)

- Major questions about logistic regression:
  - What is the ***hypothesis function*** (or ***model***)?
  - What is the ***cost function***?
  - How can we learn the parameters of the model?

Outline


Logistic Regression
Classification
Hypothesis Function
Cost Function
Gradient Descent

The Logistic Regression Model

- What will be the output of the model $\boldsymbol{h_\theta(x) = \theta^T x}$, where $\boldsymbol{\theta}$ = $[\boldsymbol{\theta_0}, \dots, \boldsymbol{\theta_m}]$ and $\boldsymbol{x} = [\boldsymbol{x_0}, \dots, \boldsymbol{x_m}]$?
  - Real-valued

- How can we make the output of $\boldsymbol{h_\theta(x)}$ discrete?
  - By using the logistic function as follows:

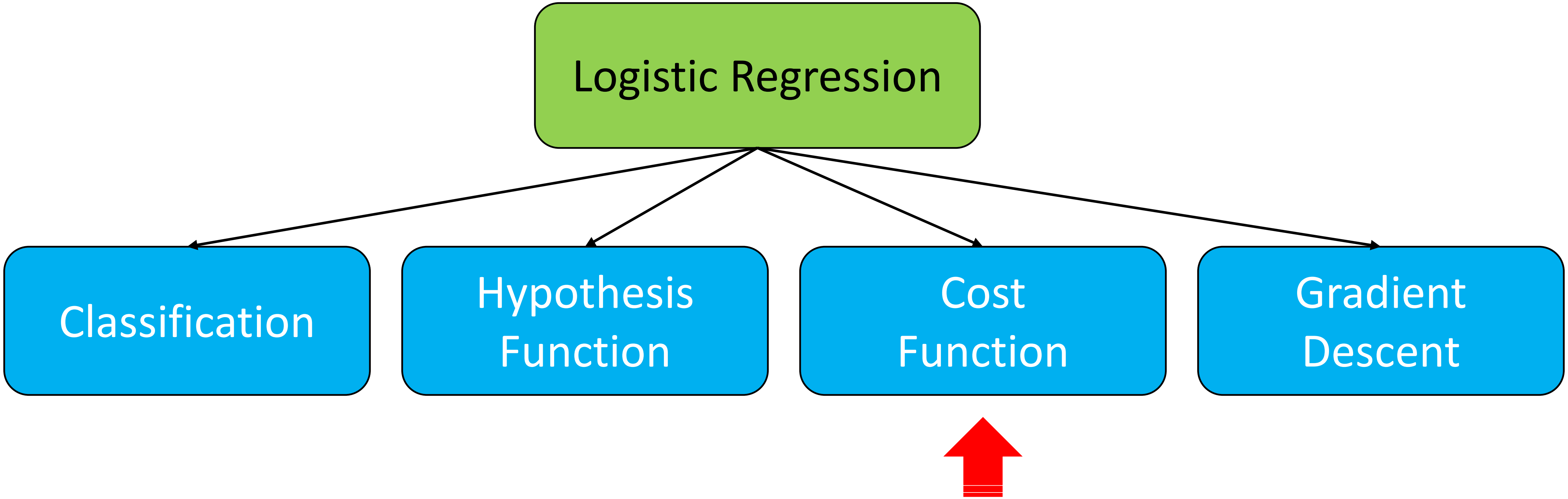
$$\boldsymbol{h_\theta(x)} = \color{red}{\boldsymbol{g(\theta^T x)}} = \boxed{\frac{1}{1+e^{-\boldsymbol{\theta}^T x}}}$$

This is the logistic regression model or hypothesis function

  - And then applying thresholding *after* learning the model to predict the output as follows:

$$\begin{cases} \boldsymbol{if\ h_\theta(x) < 0.5\ \text{predict}\ 0} \\ \boldsymbol{if\ h_\theta(x) \geq 0.5\ \text{predict}\ 1} \end{cases}$$

Outline



Logistic Regression
Classification
Hypothesis Function
Cost Function
Gradient Descent

Towards Identifying the Logistic Regression Cost Function

- How to learn a *logistic regression model* $\boldsymbol{h_\theta(x)} = \boldsymbol{g(\theta^T x)}$, where $\boldsymbol{\theta}$ = $[\boldsymbol{\theta_0}, \ldots, \boldsymbol{\theta_m}]$ and $\boldsymbol{x} = [\boldsymbol{x_0}, \ldots, \boldsymbol{x_m}]$?
  - Perhaps, by minimizing ***Mean Squared Error (MSE)***. That is:

$$\text{minimize} \frac{1}{2n} \sum_{i=1}^{n} \left(y'^{(i)} - y^{(i)}\right)^2$$

$$\equiv$$

$$\underset{\theta}{\text{minimize}} \boxed{\frac{1}{2n} \sum_{i=1}^{n} \left((g(\theta^T x))^{(i)} - y^{(i)}\right)^2}$$

Cost function $\boldsymbol{J(\theta)}$

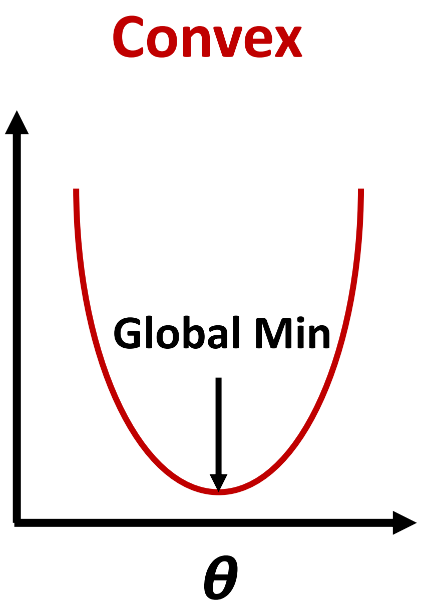
$$\equiv$$

$$\underset{\theta}{\text{minimize}}\, \boldsymbol{J(\theta)}$$

Towards Identifying the Logistic Regression Cost Function

- How to learn a *logistic regression model* $\boldsymbol{h_\theta(x)} = \boldsymbol{g(\theta^T x)}$, where $\boldsymbol{\theta}$ = $[\boldsymbol{\theta_0}, \dots, \boldsymbol{\theta_m}]$ and $\boldsymbol{x} = [\boldsymbol{x_0}, \dots, \boldsymbol{x_m}]$?
  - Perhaps, by minimizing ***Mean Squared Error (MSE)***. That is:

Easy for Gradient Descent to locate!



$$\text{minimize} \frac{1}{2n} \sum_{i=1}^{n} \left(y'^{(i)} - y^{(i)} \right)^2$$

$$\equiv$$

$$\underset{\theta}{\text{minimize}} \boxed{\frac{1}{2n} \sum_{i=1}^{n} \left((g(\theta^T x))^{(i)} - y^{(i)} \right)^2}$$

$$\equiv$$

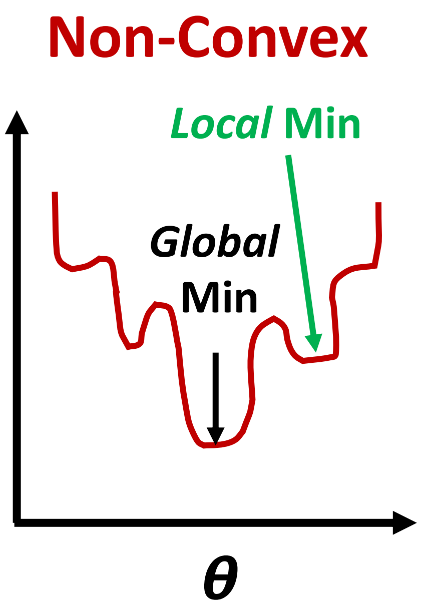
$$\underset{\theta}{\text{minimize}}\, \boldsymbol{J(\theta)}$$

Unfortunately, if we plot this cost function, it will turn out to be "**non-convex**"

Towards Identifying the Logistic Regression Cost Function

- How to learn a *logistic regression model* $\boldsymbol{h_\theta(x)} = \boldsymbol{g(\theta^T x)}$, where $\boldsymbol{\theta}$ = $[\boldsymbol{\theta_0}, \dots, \boldsymbol{\theta_m}]$ and $\boldsymbol{x} = [\boldsymbol{x_0}, \dots, \boldsymbol{x_m}]$?
  - Perhaps, by minimizing ***Mean Squared Error (MSE)***. That is:

Gradient Descent might get stuck at a *local* min and fail to locate the *global* min!



$$\text{minimize} \frac{1}{2n} \sum_{i=1}^{n} \left(y'^{(i)} - y^{(i)}\right)^2$$

$$\equiv$$

$$\underset{\theta}{\text{minimize}} \boxed{\frac{1}{2n} \sum_{i=1}^{n} \left((g(\theta^T x))^{(i)} - y^{(i)}\right)^2}$$

$$\equiv$$

$$\underset{\theta}{\text{minimize}}\, \boldsymbol{J(\theta)}$$

Unfortunately, if we plot this cost function, it will turn out to be "**non-convex**"

Towards Identifying the Logistic Regression Cost Function

- How to learn a *logistic regression model* $\boldsymbol{h_\theta(x)} = \boldsymbol{g(\theta^T x)}$, where $\boldsymbol{\theta}$ = $[\boldsymbol{\theta_0}, \ldots, \boldsymbol{\theta_m}]$ and $\boldsymbol{x} = [\boldsymbol{x_0}, \ldots, \boldsymbol{x_m}]$?
  - Perhaps, by minimizing ***Mean Squared Error (MSE)***. That is:

We need a cost function that is *convex*, hence, being more suitable for Gradient Descent

$$\text{minimize} \frac{1}{2n} \sum_{i=1}^{n} \left(y'^{(i)} - y^{(i)}\right)^2$$

$$\equiv$$

$$\underset{\theta}{\text{minimize}} \boxed{\frac{1}{2n} \sum_{i=1}^{n} \left((g(\theta^T x))^{(i)} - y^{(i)}\right)^2}$$

$$\equiv$$

$$\underset{\theta}{\text{minimize}} \, \boldsymbol{J(\theta)}$$

Unfortunately, if we plot this cost function, it will turn out to be "**non-convex**"

The Logistic Regression Cost Function

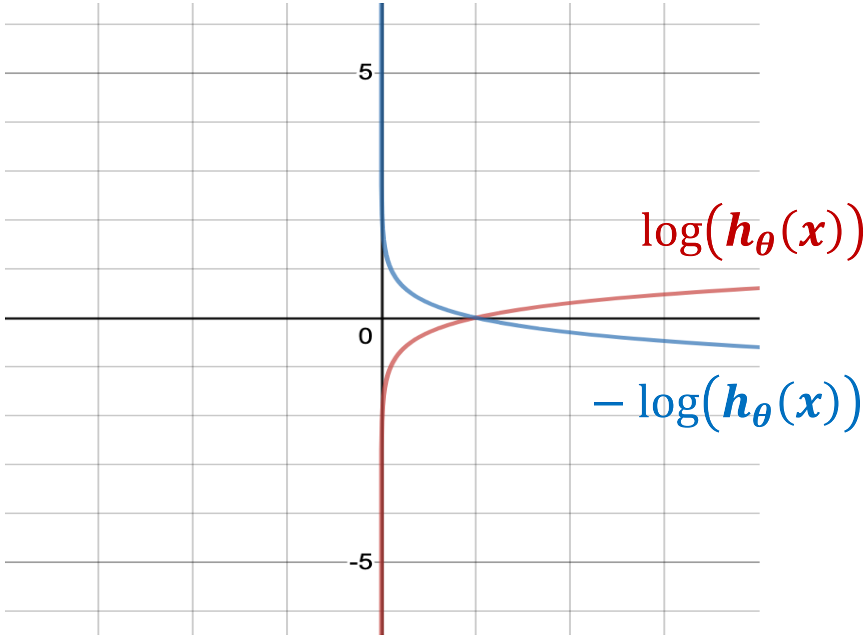
- How to learn a *logistic regression model* $\boldsymbol{h_\theta(x)} = \boldsymbol{g(\theta^T x)}$, where $\boldsymbol{\theta}$ = $[\boldsymbol{\theta_0}, \ldots, \boldsymbol{\theta_m}]$ and $\boldsymbol{x} = [\boldsymbol{x_0}, \ldots, \boldsymbol{x_m}]$?
  - Let us try a different cost function. That is:

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

The Logistic Regression Cost Function

- How to learn a *logistic regression model* $\boldsymbol{h_\theta(x)} = \boldsymbol{g(\theta^T x)}$, where $\boldsymbol{\theta}$ = $[\boldsymbol{\theta_0}, \dots, \boldsymbol{\theta_m}]$ and $\boldsymbol{x} = [\boldsymbol{x_0}, \dots, \boldsymbol{x_m}]$?
  - Let us try a different cost function. That is:

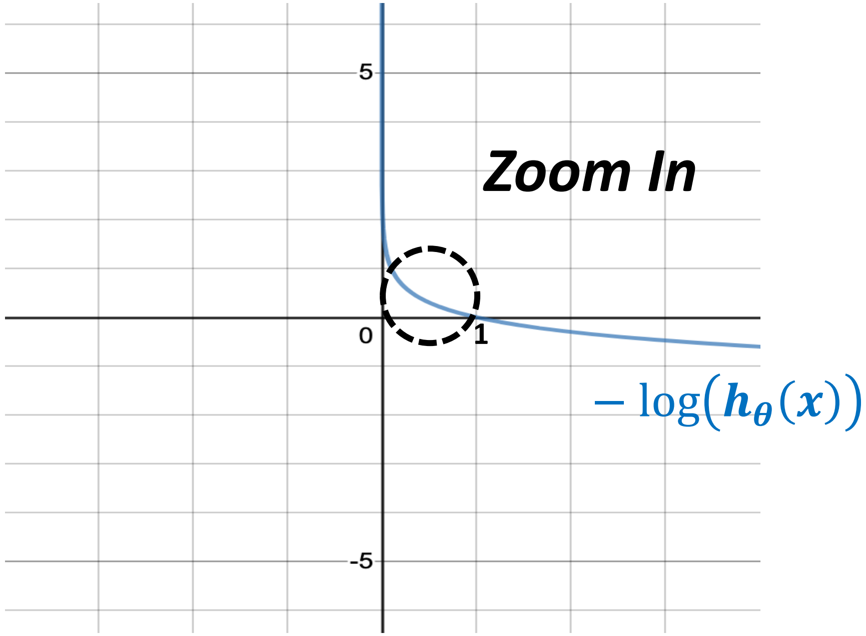
$$\text{Cost}(\boldsymbol{h_\theta(x)}, y) = \begin{cases} -\log(\boldsymbol{h_\theta(x)})\ \boldsymbol{if\ y = 1} \\ -\log(\boldsymbol{1 -\ h_\theta(x)})\ \boldsymbol{if\ y = 0} \end{cases}$$



The Logistic Regression Cost Function

- How to learn a *logistic regression model* $\boldsymbol{h_\theta(x)} = \boldsymbol{g(\theta^T x)}$, where $\boldsymbol{\theta} = [\boldsymbol{\theta_0}, \dots, \boldsymbol{\theta_m}]$ and $\boldsymbol{x} = [\boldsymbol{x_0}, \dots, \boldsymbol{x_m}]$?
  - Let us try a different cost function. That is:

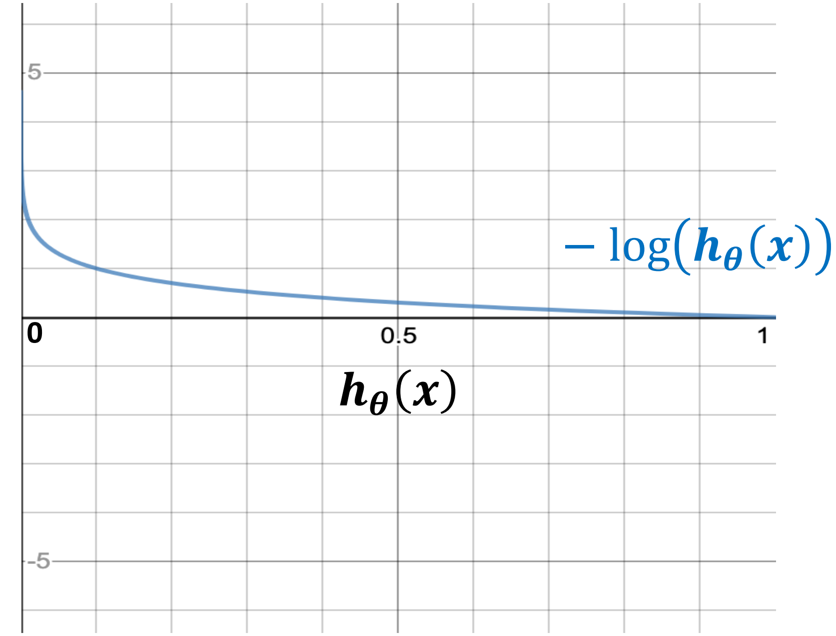
$$\text{Cost}(\boldsymbol{h_\theta(x)}, \text{y}) = \begin{cases} -\log(\boldsymbol{h_\theta(x)})\ \boldsymbol{if\ y = 1} \\ -\log(\boldsymbol{1} - \boldsymbol{h_\theta(x)})\ \boldsymbol{if\ y = 0} \end{cases}$$



The Logistic Regression Cost Function

- How to learn a *logistic regression model* $\boldsymbol{h_\theta(x)} = \boldsymbol{g(\theta^T x)}$, where $\boldsymbol{\theta}$ = $[\boldsymbol{\theta_0}, \dots, \boldsymbol{\theta_m}]$ and $\boldsymbol{x} = [\boldsymbol{x_0}, \dots, \boldsymbol{x_m}]$?
  - Let us try a different cost function. That is:

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

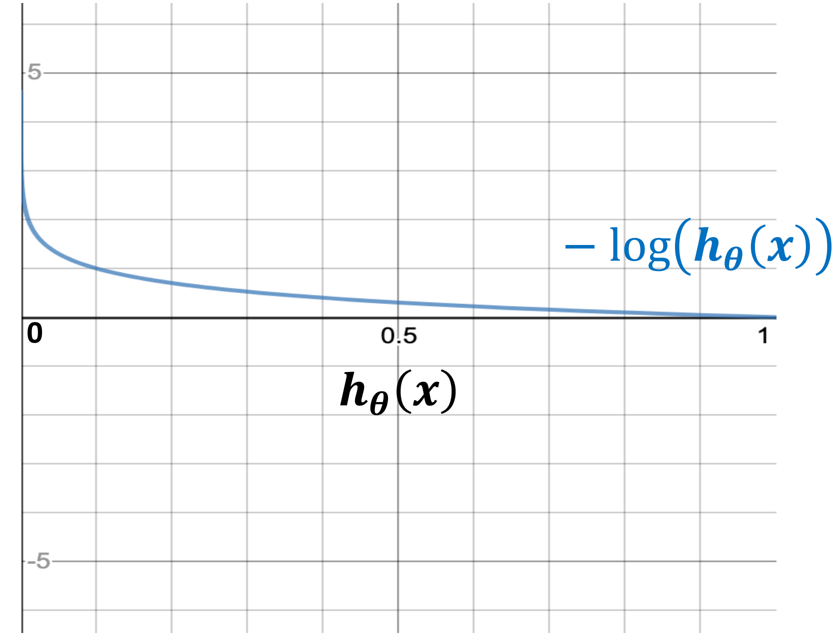


$$\boldsymbol{If\ y = 1} \begin{cases} \text{As } \boldsymbol{h_\theta(x)} \rightarrow 0,\ -\log(\boldsymbol{h_\theta(x)}) \rightarrow \infty \\ \text{As } \boldsymbol{h_\theta(x)} \rightarrow 1,\ -\log(\boldsymbol{h_\theta(x)}) \rightarrow 0 \text{ (i.e., cost} \rightarrow 0) \end{cases}$$

The Logistic Regression Cost Function

- How to learn a *logistic regression model* $\boldsymbol{h_\theta(x)} = \boldsymbol{g(\theta^T x)}$, where $\boldsymbol{\theta}$ = $[\boldsymbol{\theta_0}, \dots, \boldsymbol{\theta_m}]$ and $\boldsymbol{x} = [\boldsymbol{x_0}, \dots, \boldsymbol{x_m}]$?
    - Let us try a different cost function. That is:

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

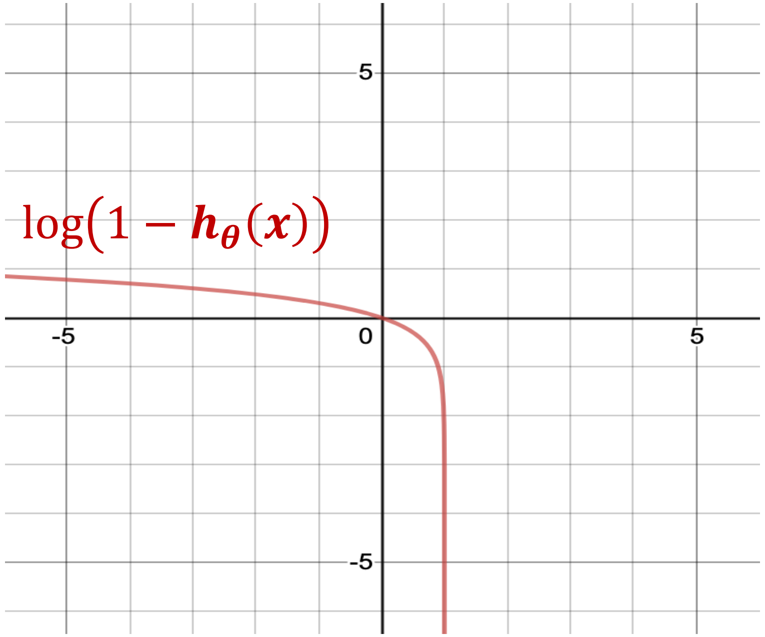


This captures the intuition that if $\boldsymbol{h_\theta(x) = 0}$ (i.e., what we will predict is $\mathbf{0}$), but $\boldsymbol{y = 1}$ (hence, we are mispredicting!), we shall penalize the learning algorithm by a very large cost!

The Logistic Regression Cost Function

- How to learn a *logistic regression model* $\boldsymbol{h_\theta(x)} = \boldsymbol{g(\theta^T x)}$, where $\boldsymbol{\theta} = [\boldsymbol{\theta_0}, \dots, \boldsymbol{\theta_m}]$ and $\boldsymbol{x} = [\boldsymbol{x_0}, \dots, \boldsymbol{x_m}]$?
  - Let us try a different cost function. That is:

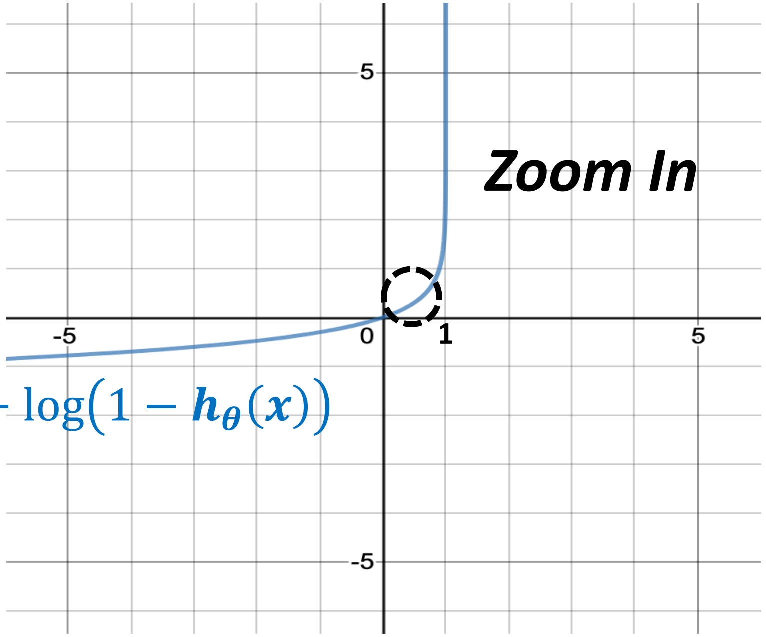
$$\text{Cost}(\boldsymbol{h_\theta(x)}, \mathrm{y}) = \begin{cases} -\log(\boldsymbol{h_\theta(x)})\ \boldsymbol{if\ y = 1} \\ -\log(\boldsymbol{1 - h_\theta(x)})\ \boldsymbol{if\ y = 0} \end{cases}$$



The Logistic Regression Cost Function

- How to learn a *logistic regression model* $\boldsymbol{h_\theta(x)} = \boldsymbol{g(\theta^T x)}$, where $\boldsymbol{\theta}$ = $[\boldsymbol{\theta_0}, \dots, \boldsymbol{\theta_m}]$ and $\boldsymbol{x} = [\boldsymbol{x_0}, \dots, \boldsymbol{x_m}]$?
  - Let us try a different cost function. That is:

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

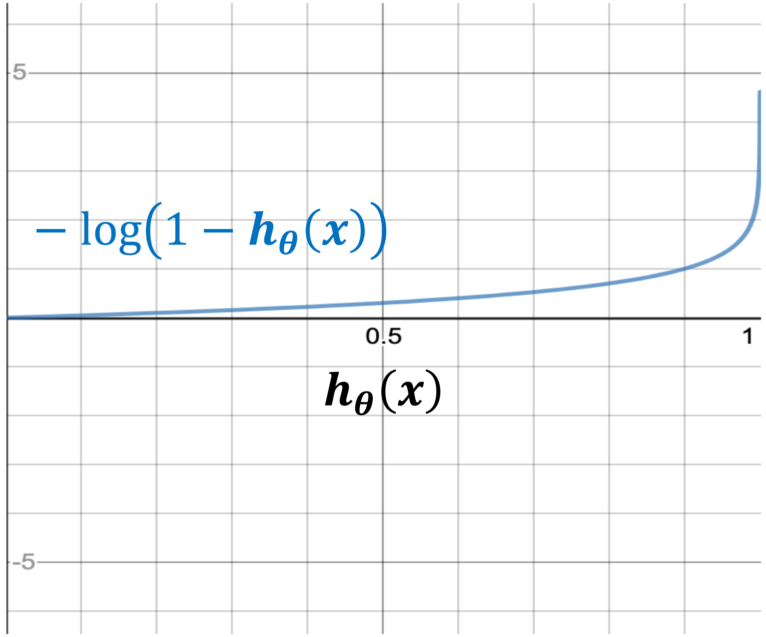


$-\log(1 - \boldsymbol{h_\theta(x)})$

The Logistic Regression Cost Function

- How to learn a *logistic regression model* $\boldsymbol{h_\theta(x)} = \boldsymbol{g(\theta^T x)}$, where $\boldsymbol{\theta}$ = $[\boldsymbol{\theta_0}, \ldots, \boldsymbol{\theta_m}]$ and $\boldsymbol{x} = [\boldsymbol{x_0}, \ldots, \boldsymbol{x_m}]$?
  - Let us try a different cost function. That is:

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

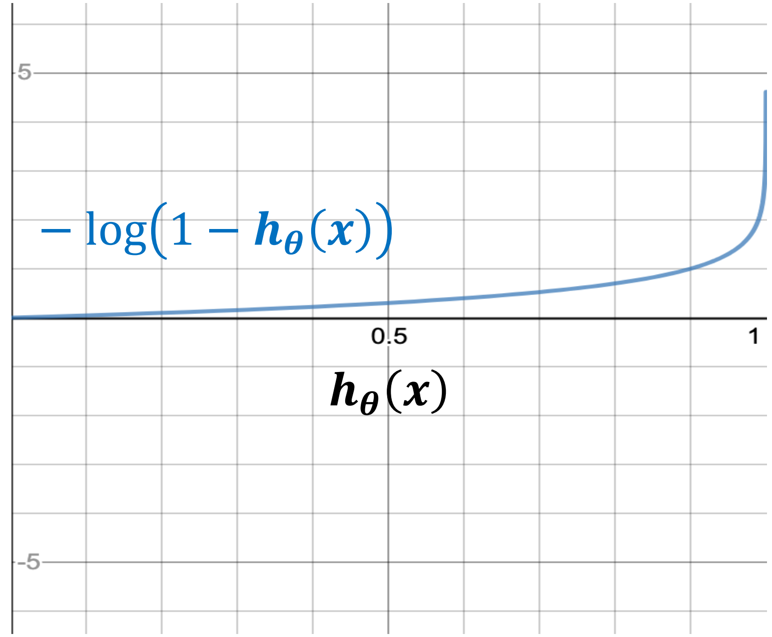


$$\boldsymbol{If\ y = 0} \begin{cases} \text{As } \boldsymbol{h_\theta(x)} \rightarrow 0,\ -\log(\boldsymbol{h_\theta(x)}) \rightarrow 0 \\ \text{As } \boldsymbol{h_\theta(x)} \rightarrow 1,\ -\log(\boldsymbol{h_\theta(x)}) \rightarrow \infty \text{ (i.e., cost } \rightarrow \infty) \end{cases}$$

The Logistic Regression Cost Function

- How to learn a *logistic regression model* $\boldsymbol{h_\theta(x)} = \boldsymbol{g(\theta^T x)}$, where $\boldsymbol{\theta}$ = $[\boldsymbol{\theta_0}, \ldots, \boldsymbol{\theta_m}]$ and $\boldsymbol{x} = [\boldsymbol{x_0}, \ldots, \boldsymbol{x_m}]$?
  - Let us try a different cost function. That is:

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



This captures the intuition that if $\boldsymbol{h_\theta(x) = 1}$ (i.e., what we will predict is $\mathbf{1}$), but $\boldsymbol{y = 0}$ (i.e., we are mispredicting!), we shall penalize the learning algorithm by a very large cost!

The Logistic Regression Cost Function

- How to learn a *logistic regression model* $\boldsymbol{h_\theta(x)} = \boldsymbol{g(\theta^T x)}$, where $\boldsymbol{\theta}$ = $[\boldsymbol{\theta_0}, \dots, \boldsymbol{\theta_m}]$ and $\boldsymbol{x} = [\boldsymbol{x_0}, \dots, \boldsymbol{x_m}]$?
  - Let us try a different cost function. That is:

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

Equivalent To

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

The Logistic Regression Cost Function

- How to learn a *logistic regression model* $\boldsymbol{h_\theta(x)} = \boldsymbol{g(\theta^T x)}$, where $\boldsymbol{\theta}$ = $[\boldsymbol{\theta_0}, \dots, \boldsymbol{\theta_m}]$ and $\boldsymbol{x} = [\boldsymbol{x_0}, \dots, \boldsymbol{x_m}]$?
  - Let us try a different cost function. That is:

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

This function still assumes real-valued outputs for $\boldsymbol{h_\theta(x)}$ (i.e., still entails *a regression problem*), while logistic regression should predict discrete values (i.e., logistic regression is a *classification problem*)

The Logistic Regression Cost Function

- How to learn a *logistic regression model* $\boldsymbol{h_\theta(x)} = \boldsymbol{g(\theta^T x)}$, where $\boldsymbol{\theta}$ = $[\boldsymbol{\theta_0}, \dots, \boldsymbol{\theta_m}]$ and $\boldsymbol{x} = [\boldsymbol{x_0}, \dots, \boldsymbol{x_m}]$?
  - Let us try a different cost function. That is:

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

We still need to apply to it the logistic function:

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

The Logistic Regression Cost Function

- How to learn a *logistic regression model* $\boldsymbol{h_\theta(x)} = \boldsymbol{g(\theta^T x)}$, where $\boldsymbol{\theta}$ = $[\boldsymbol{\theta_0}, \dots, \boldsymbol{\theta_m}]$ and $\boldsymbol{x} = [\boldsymbol{x_0}, \dots, \boldsymbol{x_m}]$?
  - By minimizing the following cost function:

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

$$= -y\log(\boldsymbol{g(\theta^T x)}) - (1-y)\log(1 - \boldsymbol{g(\theta^T x)})$$

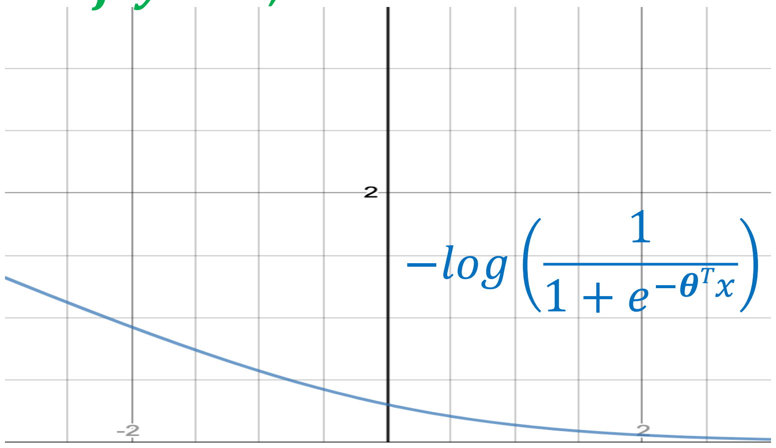
$$= -y\log\left(\frac{1}{1+e^{-\boldsymbol{\theta}^T x}}\right) - (1-y)\log\left(1 - \frac{1}{1+e^{-\boldsymbol{\theta}^T x}}\right)$$

The Logistic Regression Cost Function

- How to learn a *logistic regression model* $\boldsymbol{h_\theta(x)} = \boldsymbol{g(\theta^T x)}$, where $\boldsymbol{\theta}$ = $[\boldsymbol{\theta_0}, \dots, \boldsymbol{\theta_m}]$ and $\boldsymbol{x} = [\boldsymbol{x_0}, \dots, \boldsymbol{x_m}]$?
  - By minimizing the following cost function:

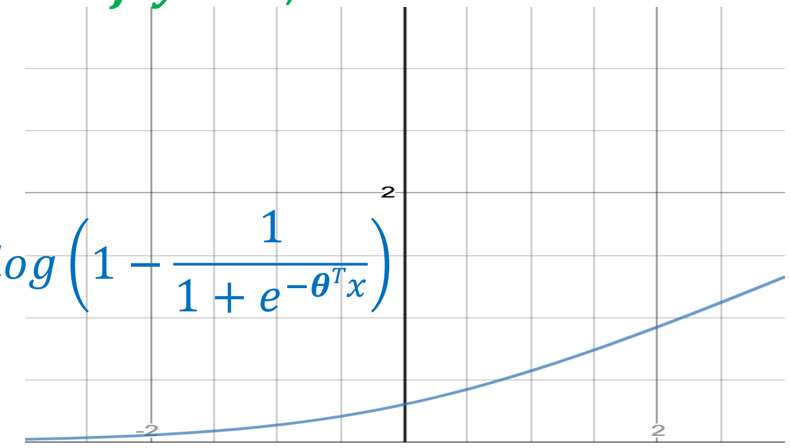
$$\text{Cost}(\boldsymbol{h_\theta(x)}, \text{y}) = \quad -y \log\left(\frac{1}{1+e^{-\boldsymbol{\theta}^T x}}\right) - (1-y)\log\left(1 - \frac{1}{1+e^{-\boldsymbol{\theta}^T x}}\right)$$

If $\boldsymbol{y = 1}$, we want $\boldsymbol{\theta^T x} >> \mathbf{0}$



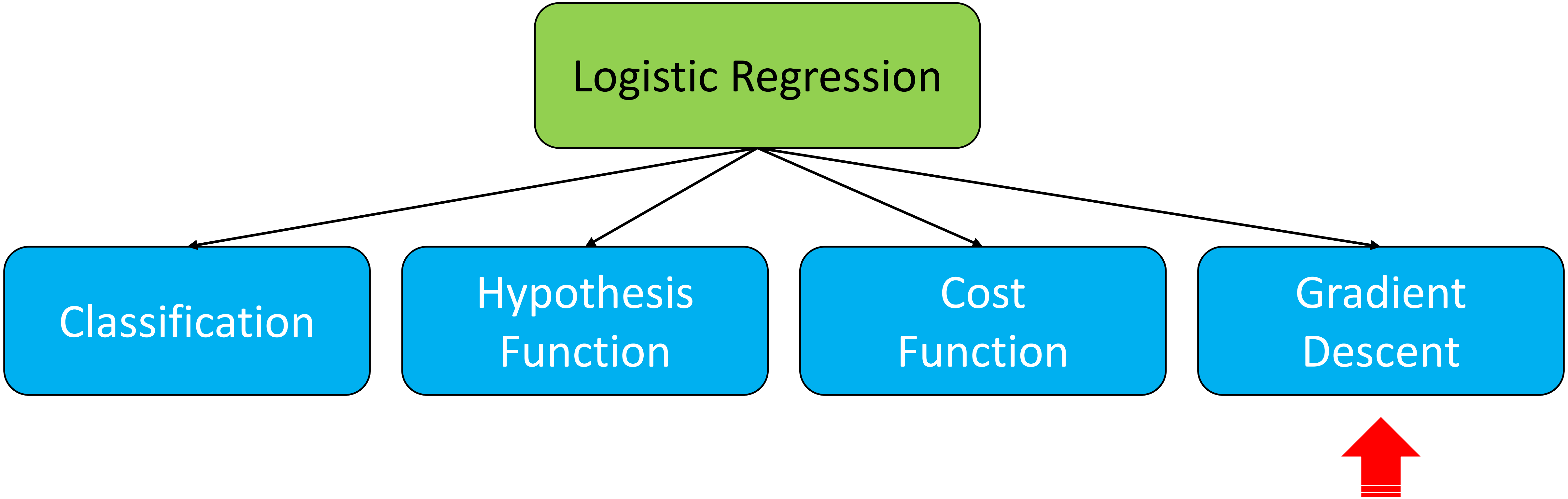
$\boldsymbol{\theta^T x}$

If $\boldsymbol{y = 0}$, we want $\boldsymbol{\theta^T x} << \mathbf{0}$



$\boldsymbol{\theta^T x}$

Outline


Logistic Regression
Classification
Hypothesis Function
Cost Function
Gradient Descent

Learning a Logistic Regression Model

- How to learn a *logistic regression model* $\boldsymbol{h_\theta}(\boldsymbol{x}) = \boldsymbol{g}(\boldsymbol{\theta}^T\boldsymbol{x})$, where $\boldsymbol{\theta}$ = $[\boldsymbol{\theta_0}, \ldots, \boldsymbol{\theta_m}]$ and $\boldsymbol{x} = [\boldsymbol{x_0}, \ldots, \boldsymbol{x_m}]$?
  - By minimizing the following cost function:

$$\text{Cost}(\boldsymbol{h_\theta}(\boldsymbol{x}), \text{y}) = \quad -y \log\left(\frac{1}{1+e^{-\boldsymbol{\theta}^T x}}\right) - (1-y)\log\left(1 - \frac{1}{1+e^{-\boldsymbol{\theta}^T x}}\right)$$

  - That is:

$$\underset{\theta}{\text{minimize}} \frac{1}{n}\sum_{i=1}^{n} \text{Cost}(\boldsymbol{h_\theta}(\boldsymbol{x})^{(i)}, y^{(i)})$$

$$\equiv$$

$$\underset{\theta}{\text{minimize}} \boxed{\frac{1}{n}\sum_{i=1}^{n} -y^{(i)} \log\left(\frac{1}{1+e^{-\boldsymbol{\theta}^T x^{(i)}}}\right) - (1-y)\log\left(1 - \frac{1}{1+e^{-\boldsymbol{\theta}^T x^{(i)}}}\right)}$$

Cost function $\boldsymbol{J(\theta)}$

Gradient Descent For Logistic Regression

- **Outline**:
  - Have cost function $\boldsymbol{J(\theta)}$, where $\boldsymbol{\theta}$ = $[\boldsymbol{\theta_0}, \dots, \boldsymbol{\theta_m}]$
  - Start off with some guesses for $\theta_0, \dots, \theta_m$
    - It does not really matter what values you start off with, but a common choice is to set them all initially to zero
  - Repeat until convergence{

Partial derivative

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

Note: Update all $\boldsymbol{\theta_j}$ simulatenously

}

***Learing rate**, which controls how big a step we take when we update θ_j*

Gradient Descent For Logistic Regression

- **Outline**:
  - Have cost function $\boldsymbol{J(\theta)}$, where $\boldsymbol{\theta}$ = $[\boldsymbol{\theta_0}, \ldots, \boldsymbol{\theta_m}]$
  - Start off with some guesses for $\theta_0, \ldots, \theta_m$
    - It does not really matter what values you start off with, but a common choice is to set them all initially to zero
  - Repeat until convergence{

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

}

The final formula after applying partial derivatives

Inference After Learning

- After learning the parameters $\boldsymbol{\theta}$ = $[\theta_0, \dots, \theta_m]$, we can predict the output of any new unseen $\boldsymbol{x} = [\boldsymbol{x_0}, \dots, \boldsymbol{x_m}]$ as follows:

$$\begin{cases} \boldsymbol{if\ h_\theta(x) = \dfrac{1}{1+e^{-\theta^T x}} < 0.5\ \text{predict}\ 0} \\ \\ \boldsymbol{Else\ if\ h_\theta(x) = \dfrac{1}{1+e^{-\theta^T x}} \geq 0.5\ \text{predict}\ 1} \end{cases}$$

A Concrete Example: The Training Phase

- Let us apply logistic regression on the spam email recognition problem, assuming $\boldsymbol{\alpha}$ = 0.5 and starting with $\boldsymbol{\theta}$ = [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

A Training Dataset

A Concrete Example: The Training Phase

- Let us apply logistic regression on the spam email recognition problem, assuming α = 0.5 and starting with θ = [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

1 entails that a word (i.e., "and") is *present* in an email (i.e., "Email **a**")

A Concrete Example: The Training Phase

- Let us apply logistic regression on the spam email recognition problem, assuming α = 0.5 and starting with θ = [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

0 entails that a word (i.e., "and") is *abscent* in an email (i.e., "Email **b**")

A Concrete Example: The Training Phase

- Let us apply logistic regression on the spam email recognition problem, assuming $\boldsymbol{\alpha}$ = 0.5 and starting with $\boldsymbol{\theta}$ = [0, 0, 0, 0, 0, 0]

We define 6 parameters (the first one, i.e., θ_0, is the intercept)

5 words (or ***features***) = $[\boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3}, \boldsymbol{x_4}, \boldsymbol{x_5}]$

	x_1 = and	x_2 = vaccine	x_3 = the	x_4 = of	x_5 = 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

A Concrete Example: The Training Phase

- Let us apply logistic regression on the spam email recognition problem, assuming α = 0.5 and starting with θ = [0, 0, 0, 0, 0, 0]

The parameter vector:
$\theta = [\theta_0, \theta_1, \theta_2, \theta_3, \theta_4, \theta_5]$

x = $[x_0, x_1, x_2, x_3, x_4, x_5]$ *The feature vector*

	$x_0 = 1$	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

To account for the intercept

Recap: Gradient Descent For Logistic Regression

- **Outline**:
  - Have cost function $\boldsymbol{J(\theta)}$, where $\boldsymbol{\theta}$ = $[\boldsymbol{\theta_0}, \dots, \boldsymbol{\theta_m}]$
  - Start off with some guesses for $\theta_0, \dots, \theta_m$
    - It does not really matter what values you start off with, but a common choice is to set them all initially to zero
  - Repeat until convergence{

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

}

First, let us calculate this factor for every example in our training dataset

A Concrete Example: The Training Phase

- Let us apply logistic regression on the spam email recognition problem, assuming $\boldsymbol{\alpha}$ = 0.5 and starting with $\boldsymbol{\theta}$ = [0, 0, 0, 0, 0, 0]

$\boldsymbol{x}$	$\boldsymbol{y}$	$\boldsymbol{\theta}^T\boldsymbol{x}$
[1,1,1,0,1,1]	**1**	[0,0,0,0,0,0]×[1,1,1,0,1,1]=0
[1,0,0,1,1,0]	**0**	[0,0,0,0,0,0]×[1,0,0,1,1,0]=0
[1,0,1,1,0,0]	**1**	[0,0,0,0,0,0]×[1,0,1,1,0,0]=0
[1,1,0,0,1,0]	**0**	[0,0,0,0,0,0]×[1,1,0,0,1,0]=0
[1,1,0,1,0,1]	**1**	[0,0,0,0,0,0]×[1,1,0,1,0,1]=0
[1,1,0,1,1,0]	**0**	[0,0,0,0,0,0]×[1,1,0,1,1,0]=0

Recap: Gradient Descent For Logistic Regression

- **Outline**:
  - Have cost function $\boldsymbol{J(\theta)}$, where $\boldsymbol{\theta}$ = $[\boldsymbol{\theta_0}, \dots, \boldsymbol{\theta_m}]$
  - Start off with some guesses for $\theta_0, \dots, \theta_m$
    - It does not really matter what values you start off with, but a common choice is to set them all initially to zero
  - Repeat until convergence{

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

}

Second, let us calculate this equation for every example in our training dataset and for every $\boldsymbol{\theta_j}$, where $\boldsymbol{j}$ is between $\boldsymbol{0}$ and $\boldsymbol{m}$

A Concrete Example: The Training Phase

- Let us apply logistic regression on the spam email recognition problem, assuming $\boldsymbol{\alpha}$ = 0.5 and starting with $\boldsymbol{\theta}$ = [0, 0, 0, 0, 0, 0]

$\boldsymbol{x}$	$\boldsymbol{y}$	$\boldsymbol{\theta}^T\boldsymbol{x}$	$(\frac{1}{1+e^{-\boldsymbol{\theta}^T\boldsymbol{x}}} - \boldsymbol{y})\boxed{\boldsymbol{x}_0}$
[1,1,1,0,1,1]	**1**	[0,0,0,0,0,0]×[1,1,1,0,1,1]=0	
[1,0,0,1,1,0]	**0**	[0,0,0,0,0,0]×[1,0,0,1,1,0]=0	
[1,0,1,1,0,0]	**1**	[0,0,0,0,0,0]×[1,0,1,1,0,0]=0	
[1,1,0,0,1,0]	**0**	[0,0,0,0,0,0]×[1,1,0,0,1,0]=0	
[1,1,0,1,0,1]	**1**	[0,0,0,0,0,0]×[1,1,0,1,0,1]=0	
[1,1,0,1,1,0]	**0**	[0,0,0,0,0,0]×[1,1,0,1,1,0]=0	

A Concrete Example: The Training Phase

- Let us apply logistic regression on the spam email recognition problem, assuming $\boldsymbol{\alpha}$ = 0.5 and starting with $\boldsymbol{\theta}$ = [0, 0, 0, 0, 0, 0]

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

Recap: Gradient Descent For Logistic Regression

- **Outline**:
  - Have cost function $\boldsymbol{J(\theta)}$, where $\boldsymbol{\theta}$ = $[\boldsymbol{\theta_0}, \dots, \boldsymbol{\theta_m}]$
  - Start off with some guesses for $\theta_0, \dots, \theta_m$
    - It does not really matter what values you start off with, but a common choice is to set them all initially to zero
  - Repeat until convergence{

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

Third, let us compute every $\boldsymbol{\theta_j}$

}

A Concrete Example: The Training Phase

- Let us apply logistic regression on the spam email recognition problem, assuming $\boldsymbol{\alpha}$ = 0.5 and starting with $\boldsymbol{\theta}$ = [0, 0, 0, 0, 0, 0]

$\boldsymbol{x}$	$\boldsymbol{y}$	$\boldsymbol{\theta^T x}$	$(\frac{1}{1+e^{-\boldsymbol{\theta}^T\boldsymbol{x}}} - \boldsymbol{y})\boldsymbol{x_0}$
[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	-0.5
[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

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

Then,

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

New $\boldsymbol{\theta_0}$

A Concrete Example: The Training Phase

- Let us apply logistic regression on the spam email recognition problem, assuming $\boldsymbol{\alpha}$ = 0.5 and starting with $\boldsymbol{\theta}$ = [0, 0, 0, 0, 0, 0]

$\boldsymbol{x}$	$\boldsymbol{y}$	$\boldsymbol{\theta^T x}$	$(\frac{1}{1+e^{-\boldsymbol{\theta}^T \boldsymbol{x}}} - \boldsymbol{y})\boldsymbol{x_0}$
[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	-0.5
[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

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

Then,

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

Old $\boldsymbol{\theta_0}$

A Concrete Example: The Training Phase

- Let us apply logistic regression on the spam email recognition problem, assuming $\boldsymbol{\alpha}$ = 0.5 and starting with $\boldsymbol{\theta}$ = [0, 0, 0, 0, 0, 0]

$\boldsymbol{x}$	$\boldsymbol{y}$	$\boldsymbol{\theta^T x}$	$(\frac{1}{1+e^{-\boldsymbol{\theta}^T\boldsymbol{x}}} - \boldsymbol{y})\boldsymbol{x_0}$
[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	-0.5
[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

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

Then,

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

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

New Paramter Vector:

$$\boldsymbol{\theta} = [0, \boldsymbol{\theta_1}, \boldsymbol{\theta_2}, \boldsymbol{\theta_3}, \boldsymbol{\theta_4}, \boldsymbol{\theta_5}]$$

A Concrete Example: The Training Phase

- Let us apply logistic regression on the spam email recognition problem, assuming $\boldsymbol{\alpha}$ = 0.5 and starting with $\boldsymbol{\theta}$ = [0, 0, 0, 0, 0, 0]

$\boldsymbol{x}$	$\boldsymbol{y}$	$\boldsymbol{\theta}^T\boldsymbol{x}$	$\left(\frac{1}{1+e^{-\boldsymbol{\theta}^T\boldsymbol{x}}} - \boldsymbol{y}\right)\boxed{\boldsymbol{x}_1}$
[1,1,1,0,1,1]	**1**	[0,0,0,0,0,0]×[1,1,1,0,1,1]=0	
[1,0,0,1,1,0]	**0**	[0,0,0,0,0,0]×[1,0,0,1,1,0]=0	
[1,0,1,1,0,0]	**1**	[0,0,0,0,0,0]×[1,0,1,1,0,0]=0	
[1,1,0,0,1,0]	**0**	[0,0,0,0,0,0]×[1,1,0,0,1,0]=0	
[1,1,0,1,0,1]	**1**	[0,0,0,0,0,0]×[1,1,0,1,0,1]=0	
[1,1,0,1,1,0]	**0**	[0,0,0,0,0,0]×[1,1,0,1,1,0]=0	

A Concrete Example: The Training Phase

- Let us apply logistic regression on the spam email recognition problem, assuming $\boldsymbol{\alpha}$ = 0.5 and starting with $\boldsymbol{\theta}$ = [0, 0, 0, 0, 0, 0]

$\boldsymbol{x}$	$\boldsymbol{y}$	$\boldsymbol{\theta^T x}$	$(\frac{1}{1+e^{-\boldsymbol{\theta}^T \boldsymbol{x}}} - \boldsymbol{y})\boldsymbol{x_1}$
[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

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

Then,

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

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

New Paramter Vector:

$$\boldsymbol{\theta} = [\mathbf{0}, \mathbf{0}, \boldsymbol{\theta_2}, \boldsymbol{\theta_3}, \boldsymbol{\theta_4}, \boldsymbol{\theta_5}]$$

A Concrete Example: The Training Phase

- Let us apply logistic regression on the spam email recognition problem, assuming $\boldsymbol{\alpha}$ = 0.5 and starting with $\boldsymbol{\theta}$ = [0, 0, 0, 0, 0, 0]

$\boldsymbol{x}$	$\boldsymbol{y}$	$\boldsymbol{\theta}^T \boldsymbol{x}$	$(\frac{1}{1+e^{-\boldsymbol{\theta}^T \boldsymbol{x}}} - \boldsymbol{y})\boldsymbol{x}_2$
[1,1,1,0,1,1]	**1**	[0,0,0,0,0,0]×[1,1,1,0,1,1]=0	
[1,0,0,1,1,0]	**0**	[0,0,0,0,0,0]×[1,0,0,1,1,0]=0	
[1,0,1,1,0,0]	**1**	[0,0,0,0,0,0]×[1,0,1,1,0,0]=0	
[1,1,0,0,1,0]	**0**	[0,0,0,0,0,0]×[1,1,0,0,1,0]=0	
[1,1,0,1,0,1]	**1**	[0,0,0,0,0,0]×[1,1,0,1,0,1]=0	
[1,1,0,1,1,0]	**0**	[0,0,0,0,0,0]×[1,1,0,1,1,0]=0	

A Concrete Example: The Training Phase

- Let us apply logistic regression on the spam email recognition problem, assuming $\boldsymbol{\alpha}$ = 0.5 and starting with $\boldsymbol{\theta}$ = [0, 0, 0, 0, 0, 0]

$\boldsymbol{x}$	$\boldsymbol{y}$	$\boldsymbol{\theta}^T\boldsymbol{x}$	$(\frac{1}{1+e^{-\boldsymbol{\theta}^T\boldsymbol{x}}} - \boldsymbol{y})\boldsymbol{x}_2$
[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

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

Then,

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

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

New Paramter Vector:

$$\boldsymbol{\theta} = [\mathbf{0}, \mathbf{0}, \mathbf{0.5}, \boldsymbol{\theta_3}, \boldsymbol{\theta_4}, \boldsymbol{\theta_5}]$$

A Concrete Example: The Training Phase

- Let us apply logistic regression on the spam email recognition problem, assuming $\boldsymbol{\alpha}$ = 0.5 and starting with $\boldsymbol{\theta}$ = [0, 0, 0, 0, 0, 0]

$\boldsymbol{x}$	$\boldsymbol{y}$	$\boldsymbol{\theta}^T\boldsymbol{x}$	$\left(\frac{1}{1+e^{-\boldsymbol{\theta}^T\boldsymbol{x}}} - \boldsymbol{y}\right)\boxed{\boldsymbol{x}_3}$
[1,1,1,0,1,1]	**1**	[0,0,0,0,0,0]×[1,1,1,0,1,1]=0	
[1,0,0,1,1,0]	**0**	[0,0,0,0,0,0]×[1,0,0,1,1,0]=0	
[1,0,1,1,0,0]	**1**	[0,0,0,0,0,0]×[1,0,1,1,0,0]=0	
[1,1,0,0,1,0]	**0**	[0,0,0,0,0,0]×[1,1,0,0,1,0]=0	
[1,1,0,1,0,1]	**1**	[0,0,0,0,0,0]×[1,1,0,1,0,1]=0	
[1,1,0,1,1,0]	**0**	[0,0,0,0,0,0]×[1,1,0,1,1,0]=0	

A Concrete Example: The Training Phase

- Let us apply logistic regression on the spam email recognition problem, assuming $\boldsymbol{\alpha}$ = 0.5 and starting with $\boldsymbol{\theta}$ = [0, 0, 0, 0, 0, 0]

$\boldsymbol{x}$	$\boldsymbol{y}$	$\boldsymbol{\theta^T x}$	$(\frac{1}{1+e^{-\boldsymbol{\theta}^T \boldsymbol{x}}} - \boldsymbol{y})\boldsymbol{x_3}$
[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

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

Then,

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

$$= 0 - 0.5 \times 0 = 0$$

New Paramter Vector:

$$\boldsymbol{\theta} = [\mathbf{0}, \mathbf{0}, \mathbf{0.5}, \mathbf{0}, \boldsymbol{\theta_4}, \boldsymbol{\theta_5}]$$

A Concrete Example: The Training Phase

- Let us apply logistic regression on the spam email recognition problem, assuming $\boldsymbol{\alpha}$ = 0.5 and starting with $\boldsymbol{\theta}$ = [0, 0, 0, 0, 0, 0]

$\boldsymbol{x}$	$\boldsymbol{y}$	$\boldsymbol{\theta}^T\boldsymbol{x}$	$(\frac{1}{1+e^{-\boldsymbol{\theta}^T\boldsymbol{x}}} - \boldsymbol{y})\boxed{\boldsymbol{x}_4}$
[1,1,1,0,1,1]	**1**	[0,0,0,0,0,0]×[1,1,1,0,1,1]=0	
[1,0,0,1,1,0]	**0**	[0,0,0,0,0,0]×[1,0,0,1,1,0]=0	
[1,0,1,1,0,0]	**1**	[0,0,0,0,0,0]×[1,0,1,1,0,0]=0	
[1,1,0,0,1,0]	**0**	[0,0,0,0,0,0]×[1,1,0,0,1,0]=0	
[1,1,0,1,0,1]	**1**	[0,0,0,0,0,0]×[1,1,0,1,0,1]=0	
[1,1,0,1,1,0]	**0**	[0,0,0,0,0,0]×[1,1,0,1,1,0]=0	

A Concrete Example: The Training Phase

- Let us apply logistic regression on the spam email recognition problem, assuming $\boldsymbol{\alpha}$ = 0.5 and starting with $\boldsymbol{\theta}$ = [0, 0, 0, 0, 0, 0]

$\boldsymbol{x}$	$\boldsymbol{y}$	$\boldsymbol{\theta^T x}$	$(\frac{1}{1+e^{-\boldsymbol{\theta}^T\boldsymbol{x}}} - \boldsymbol{y})\boldsymbol{x_4}$
[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

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

Then,

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

$$= 0 - 0.5\times 1 = -0.5$$

New Paramter Vector:

$$\boldsymbol{\theta} = [\mathbf{0}, \mathbf{0}, \mathbf{0.5}, \mathbf{0}, \mathbf{-0.5}, \boldsymbol{\theta_5}]$$

A Concrete Example: The Training Phase

- Let us apply logistic regression on the spam email recognition problem, assuming $\boldsymbol{\alpha}$ = 0.5 and starting with $\boldsymbol{\theta}$ = [0, 0, 0, 0, 0, 0]

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

A Concrete Example: The Training Phase

- Let us apply logistic regression on the spam email recognition problem, assuming $\boldsymbol{\alpha}$ = 0.5 and starting with $\boldsymbol{\theta}$ = [0, 0, 0, 0, 0, 0]

$\boldsymbol{x}$	$\boldsymbol{y}$	$\boldsymbol{\theta}^T\boldsymbol{x}$	$(\frac{1}{1+e^{-\boldsymbol{\theta}^T\boldsymbol{x}}} - \boldsymbol{y})\boldsymbol{x}_5$
[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

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

Then,

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

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

New Paramter Vector:

$$\boldsymbol{\theta} = [\mathbf{0}, \mathbf{0}, \mathbf{0.5}, \mathbf{0}, -\mathbf{0.5}, \mathbf{0.5}]$$

A Concrete Example: Testing

- Let us now *test* logistic regression on the spam email recognition problem, using the just learnt $\boldsymbol{\theta} = [\mathbf{0}, \mathbf{0}, \mathbf{0.5}, \mathbf{0}, \mathbf{-0.5}, \mathbf{0.5}]$
  - **Note**: Testing is typically done over a portion of the dataset that is not used during training, but rather kept only for testing the accuracy of the algorithm's predictions thus far
  - In this example, we will test over all the examples that we *used* during training, *just* for illustrative purposes

A Concrete Example: Testing

- Let us *test* logistic regression on the spam email recognition problem, using the just learnt $\boldsymbol{\theta} = [\mathbf{0}, \mathbf{0}, \mathbf{0.5}, \mathbf{0}, \mathbf{-0.5}, \mathbf{0.5}]$

$\boldsymbol{x}$	$\boldsymbol{y}$	$\boldsymbol{\theta}^T\boldsymbol{x}$
[1,1,1,0,1,1]	**1**	[0,0,0.5,0,-0.5,0.5]×[1,1,1,0,1,1]=0.5
[1,0,0,1,1,0]	**0**	[0,0,0.5,0,-0.5,0.5]×[1,0,0,1,1,0]=-0.5
[1,0,1,1,0,0]	**1**	[0,0,0.5,0,-0.5,0.5]×[1,0,1,1,0,0]=0.5
[1,1,0,0,1,0]	**0**	[0,0,0.5,0,-0.5,0.5]×[1,1,0,0,1,0]=-0.5
[1,1,0,1,0,1]	**1**	[0,0,0.5,0,-0.5,0.5]×[1,1,0,1,0,1]=0.5
[1,1,0,1,1,0]	**0**	[0,0,0.5,0,-0.5,0.5]×[1,1,0,1,1,0]=-0.5

A Concrete Example: Testing

- Let us *test* logistic regression on the spam email recognition problem, using the just learnt $\boldsymbol{\theta} = [\mathbf{0}, \mathbf{0}, \mathbf{0.5}, \mathbf{0}, \mathbf{-0.5}, \mathbf{0.5}]$

$\boldsymbol{x}$	$\boldsymbol{y}$	$\boldsymbol{\theta^T x}$	$\boldsymbol{h_\theta}(\boldsymbol{x}) = (\frac{1}{1+e^{-\boldsymbol{\theta^T x}}})$
[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
[1,0,0,1,1,0]	**0**	[0,0,0.5,0,-0.5,0.5]×[1,0,0,1,1,0]=-0.5	0.377540669
[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
[1,1,0,0,1,0]	**0**	[0,0,0.5,0,-0.5,0.5]×[1,1,0,0,1,0]=-0.5	0.377540669
[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
[1,1,0,1,1,0]	**0**	[0,0,0.5,0,-0.5,0.5]×[1,1,0,1,1,0]=-0.5	0.377540669

A Concrete Example: Testing

- Let us *test* logistic regression on the spam email recognition problem, using the just learnt $\boldsymbol{\theta} = [\mathbf{0}, \mathbf{0}, \mathbf{0.5}, \mathbf{0}, \mathbf{-0.5}, \mathbf{0.5}]$

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

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

A Concrete Example: Testing

- Let us *test* logistic regression on the spam email recognition problem, using the just learnt $\boldsymbol{\theta} = [\mathbf{0}, \mathbf{0}, \mathbf{0.5}, \mathbf{0}, \mathbf{-0.5}, \mathbf{0.5}]$

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

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

A Concrete Example: Testing

- Let us *test* logistic regression on the spam email recognition problem, using the just learnt $\boldsymbol{\theta = [0, 0, 0.5, 0, -0.5, 0.5]}$

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

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

NO Mispredictions!

A Concrete Example: Inference

- Let us infer whether a given new email, say, ***k*** = [1, 0, 1, 0, 0, 1] is a spam or not, using logistic regression with the just learnt parameter vector $\boldsymbol{\theta} = [\mathbf{0}, \mathbf{0}, \mathbf{0.5}, \mathbf{0}, \mathbf{-0.5}, \mathbf{0.5}]$

	$\boldsymbol{x_0} = 1$	$\boldsymbol{x_1}$ = and	$\boldsymbol{x_2}$ = vaccine	$\boldsymbol{x_3}$ = the	$\boldsymbol{x_4}$ = of	$\boldsymbol{x_5}$ = 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

Our Training Dataset

A Concrete Example: Inference

- Let us infer whether a given new email, say, ***k*** = [1, 0, 1, 0, 0, 1] is a spam or not, using logistic regression with the just learnt parameter vector $\boldsymbol{\theta} = [\mathbf{0}, \mathbf{0}, \mathbf{0.5}, \mathbf{0}, \mathbf{-0.5}, \mathbf{0.5}]$

	$x_0 = 1$	$x_1 =$ 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
Email **k**	1	0	1	0	0	1	**?**

A Concrete Example: Inference

- Let us infer whether a given new email, say, ***k*** = [1, 0, 1, 0, 0, 1] is a spam or not, using logistic regression with the just learnt parameter vector $\boldsymbol{\theta} = [\mathbf{0}, \mathbf{0}, \mathbf{0.5}, \mathbf{0}, \mathbf{-0.5}, \mathbf{0.5}]$

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

$$\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0.5} \\ \mathbf{0} \\ \mathbf{-0.5} \\ \mathbf{0.5} \end{bmatrix} [\mathbf{1}, \mathbf{0}, \mathbf{1}, \mathbf{0}, \mathbf{0}, \mathbf{1}] = (\mathbf{0.5} \times \mathbf{1}) + (\mathbf{0.5} \times \mathbf{1}) = \mathbf{1}$$

$$= \frac{1}{1 + e^{-\mathbf{1}}}$$

$$= 0.731$$

≥ 0.5 ➔ Class 1 (i.e., Spam)

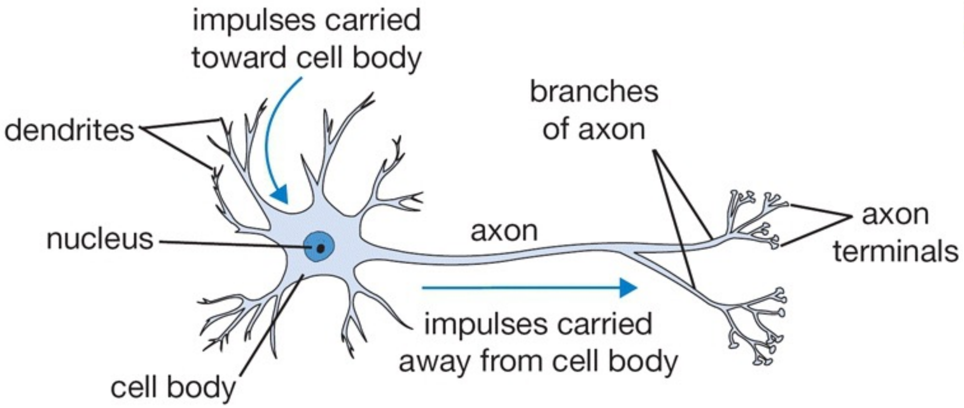
A Concrete Example: Inference

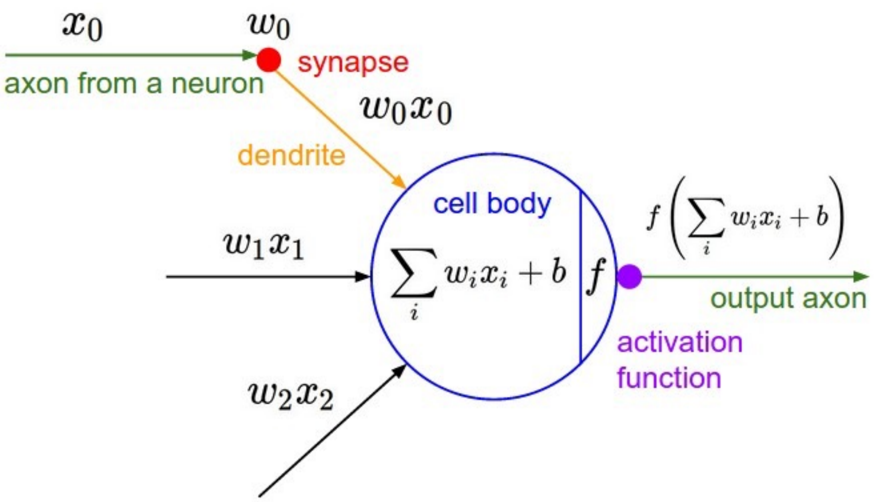
- Let us infer whether a given new email, say, ***k*** = [1, 0, 1, 0, 0, 1] is a spam or not, using logistic regression with the just learnt parameter vector $\boldsymbol{\theta} = [\mathbf{0}, \mathbf{0}, \mathbf{0.5}, \mathbf{0}, \mathbf{-0.5}, \mathbf{0.5}]$

	$\boldsymbol{x_0}$ = 1	$\boldsymbol{x_1}$ = and	$\boldsymbol{x_2}$ = vaccine	$\boldsymbol{x_3}$ = the	$\boldsymbol{x_4}$ = of	$\boldsymbol{x_5}$ = 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
Email **k**	1	0	1	0	0	1	**1**

Somehow interesting since it considered “vaccine” and “nigeria” indicative of spam!

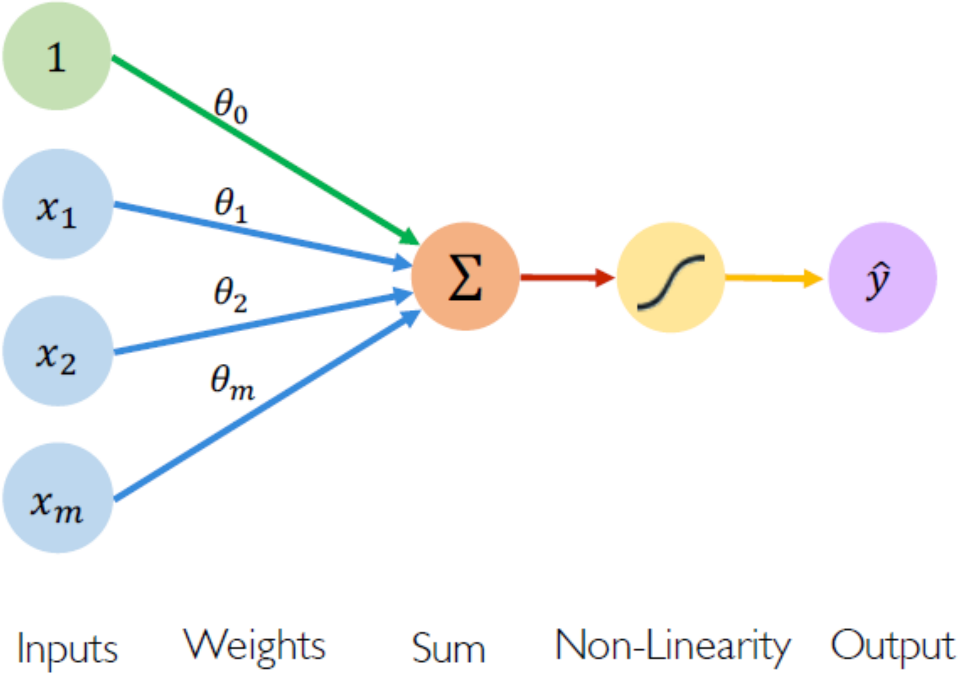
Inspiration behind Neural Networks





http://cs231n.github.io/neural-networks-1/

Perceptron



Output

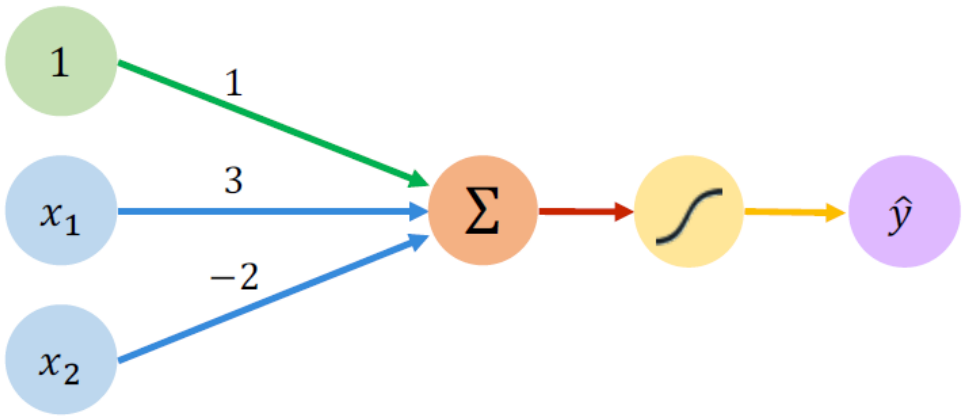
Linear combination of inputs

$$\hat{y} = g\left(\theta_0 + \sum_{i=1}^{m} x_i\,\theta_i\right)$$

Non-linear activation function

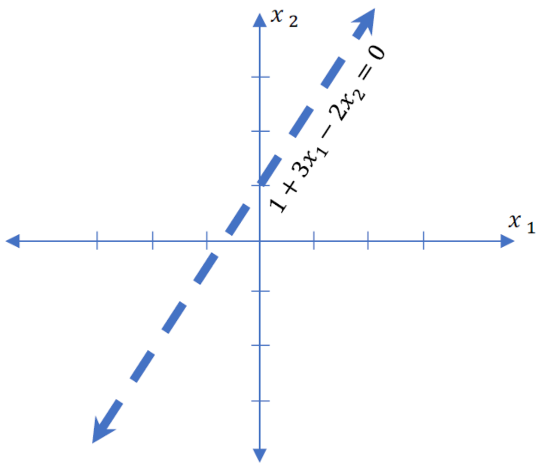
Bias

Why Non-linear Activation?

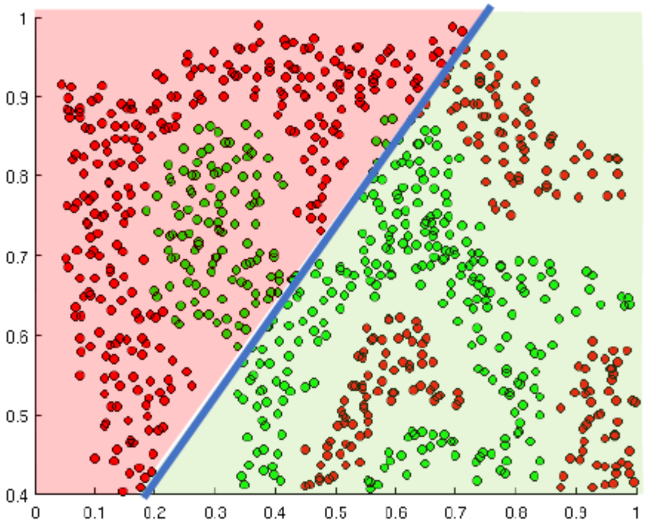


We have: $\theta_0 = 1$ and $\boldsymbol{\theta} = \begin{bmatrix} 3 \\ -2 \end{bmatrix}$

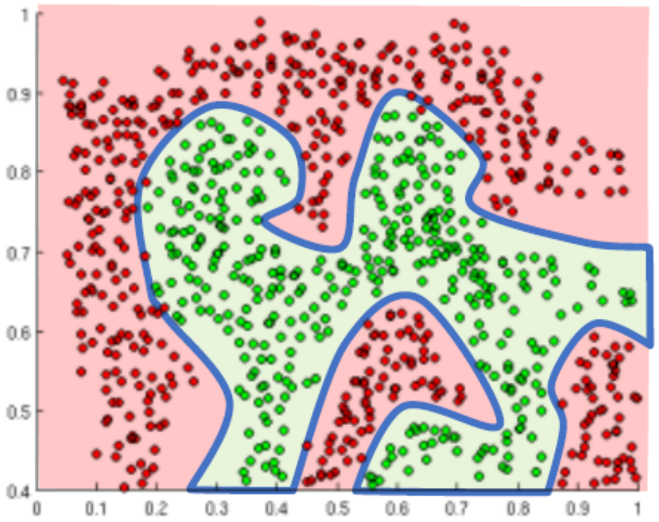
$$\hat{y} = g(\theta_0 + \boldsymbol{X}^T\boldsymbol{\theta})$$
$$= g\left(1 + \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 3 \\ -2 \end{bmatrix}\right)$$
$$\hat{y} = g(\underbrace{1 + 3x_1 - 2x_2}_{\text{This is just a line in 2D!}})$$



Why Non-linear Activation?



Linear Activation functions produce linear decisions no matter the network size

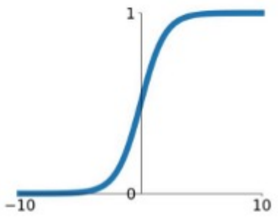


Non-linearities allow us to approximate arbitrarily complex functions

Activation Functions

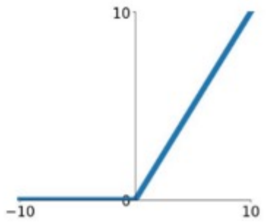
Sigmoid

$\sigma(x) = \frac{1}{1+e^{-x}}$



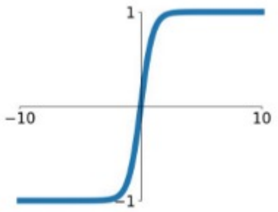
ReLU

$\max(0, x)$



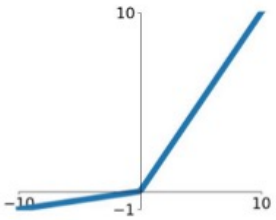
tanh

$\tanh(x)$



Leaky ReLU

$\max(0.1x, x)$



Activation Functions

- Main problem with Sigmoid and tanh activation

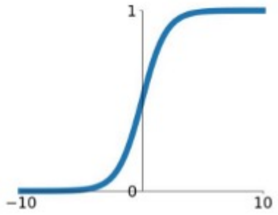
Vanishing gradient problem

Issue of exceeding small gradients when training neural networks

Worse with multiple layers in a NN

Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

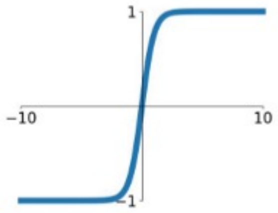


Additional problem with Sigmoid

Slower convergence than tanh

tanh

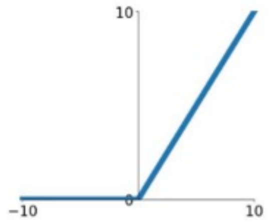
$$\tanh(x)$$



Activation Functions

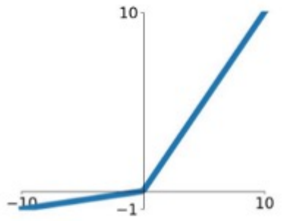
- Rectified Linear Units (ReLU)
  - Developed to overcome the vanishing gradient problem
  - However, some neuron may “die”

ReLU
$$\max(0, x)$$

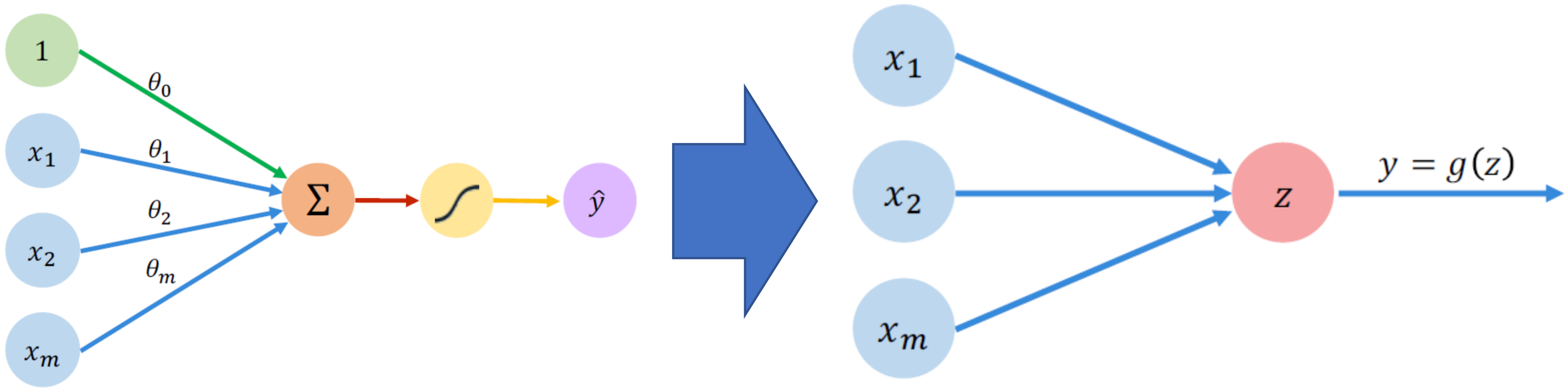


- Leaky ReLU
  - Prevents neurons from “dying” by using a small negative slope

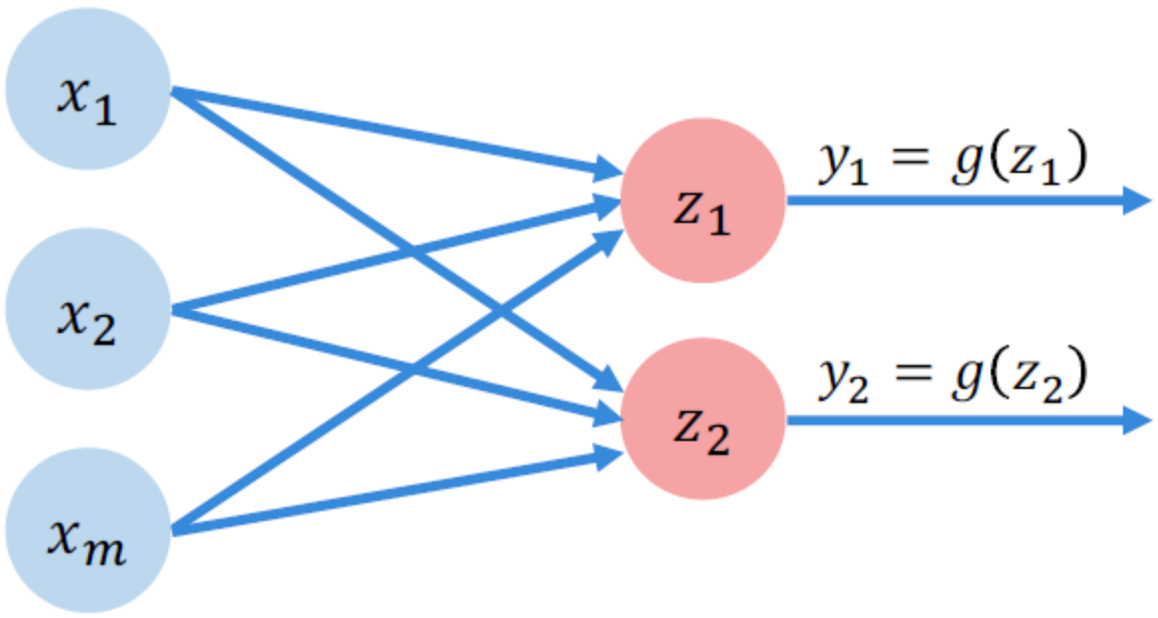
Leaky ReLU
$$\max(0.1x, x)$$



Simplifying the Perceptron

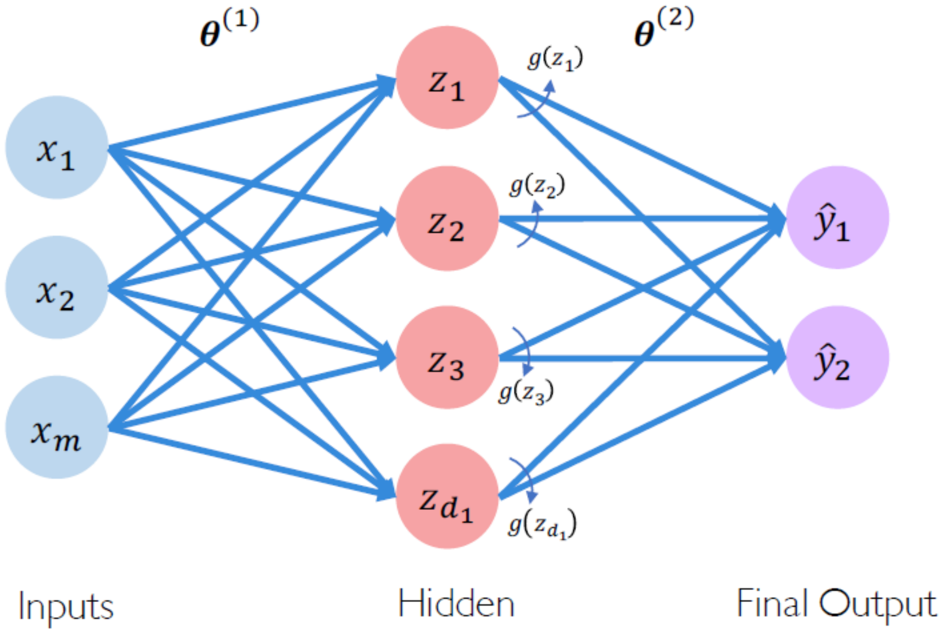


Multiple Perceptron



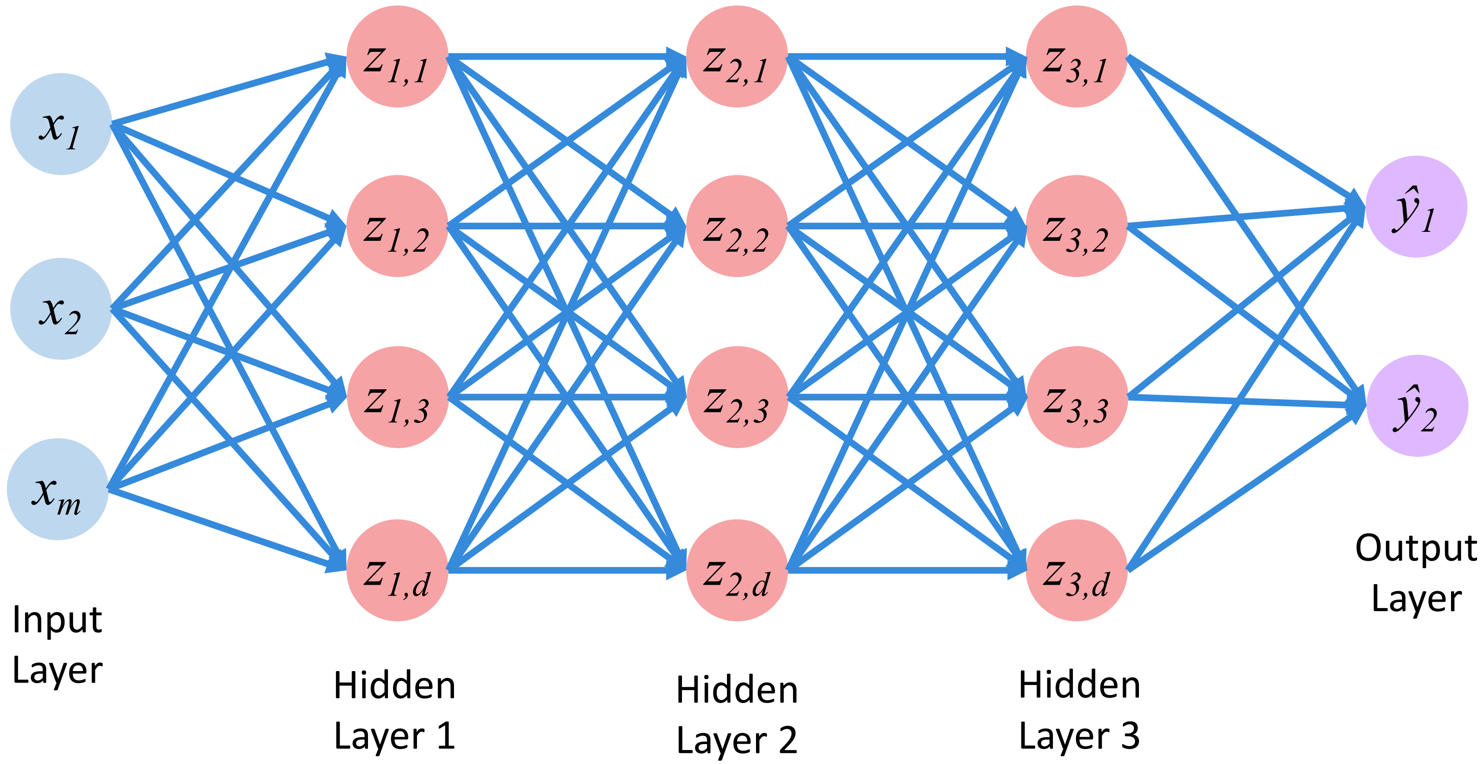
Neural Network

- Multiple perceptrons, aka a Multi-layer Perceptron (MLP)

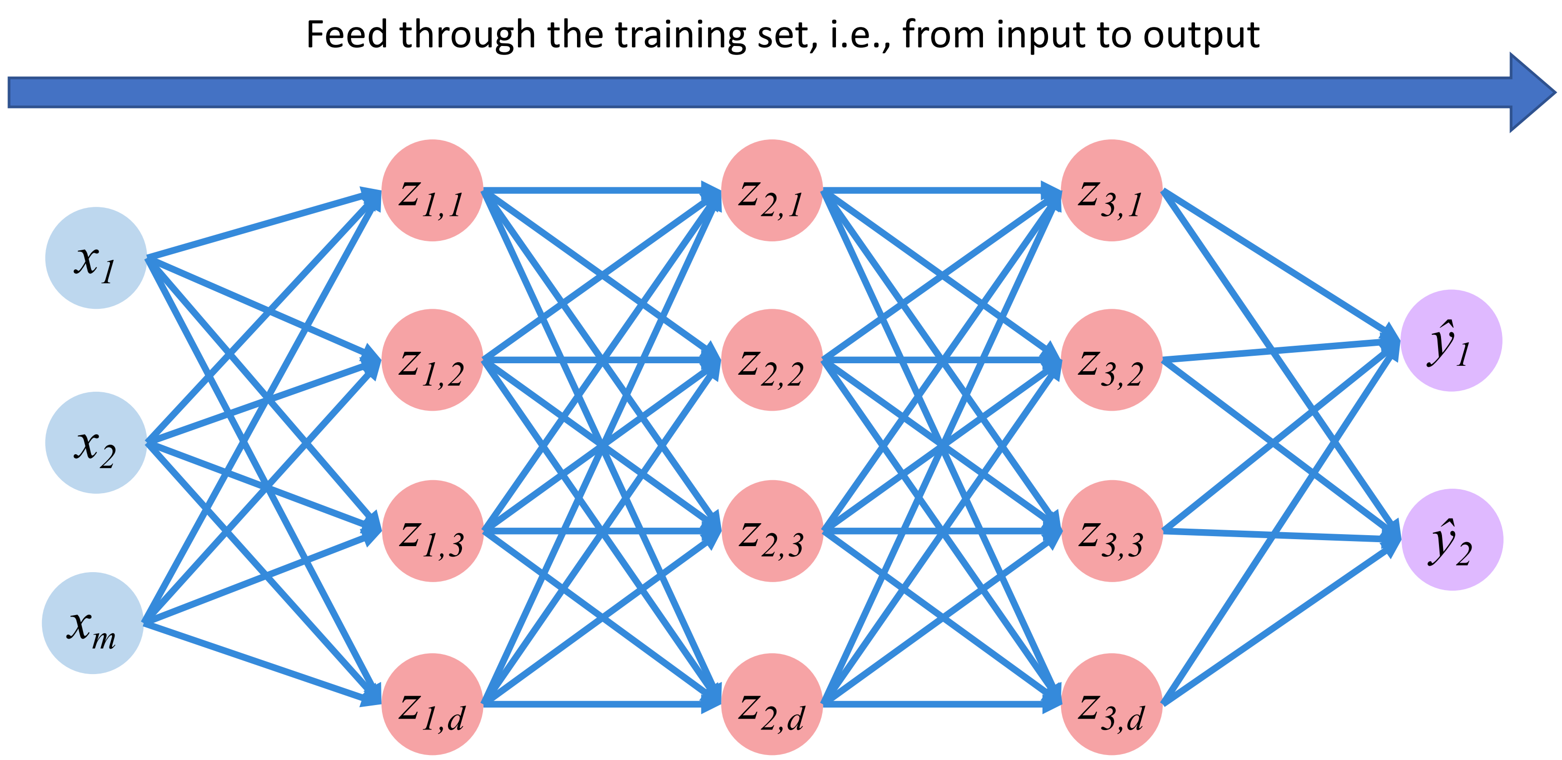


Deep Neural Network

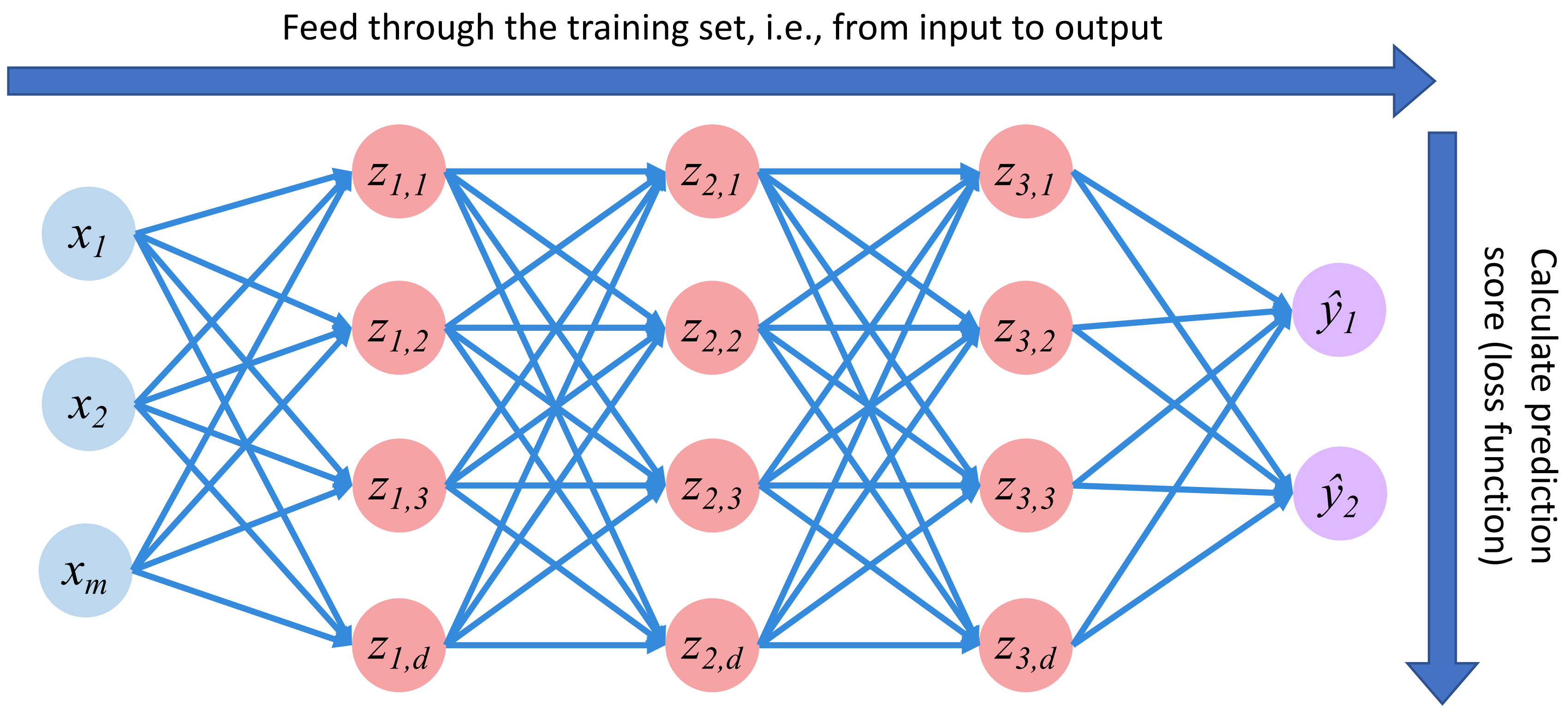
- Adding on multiple hidden layers



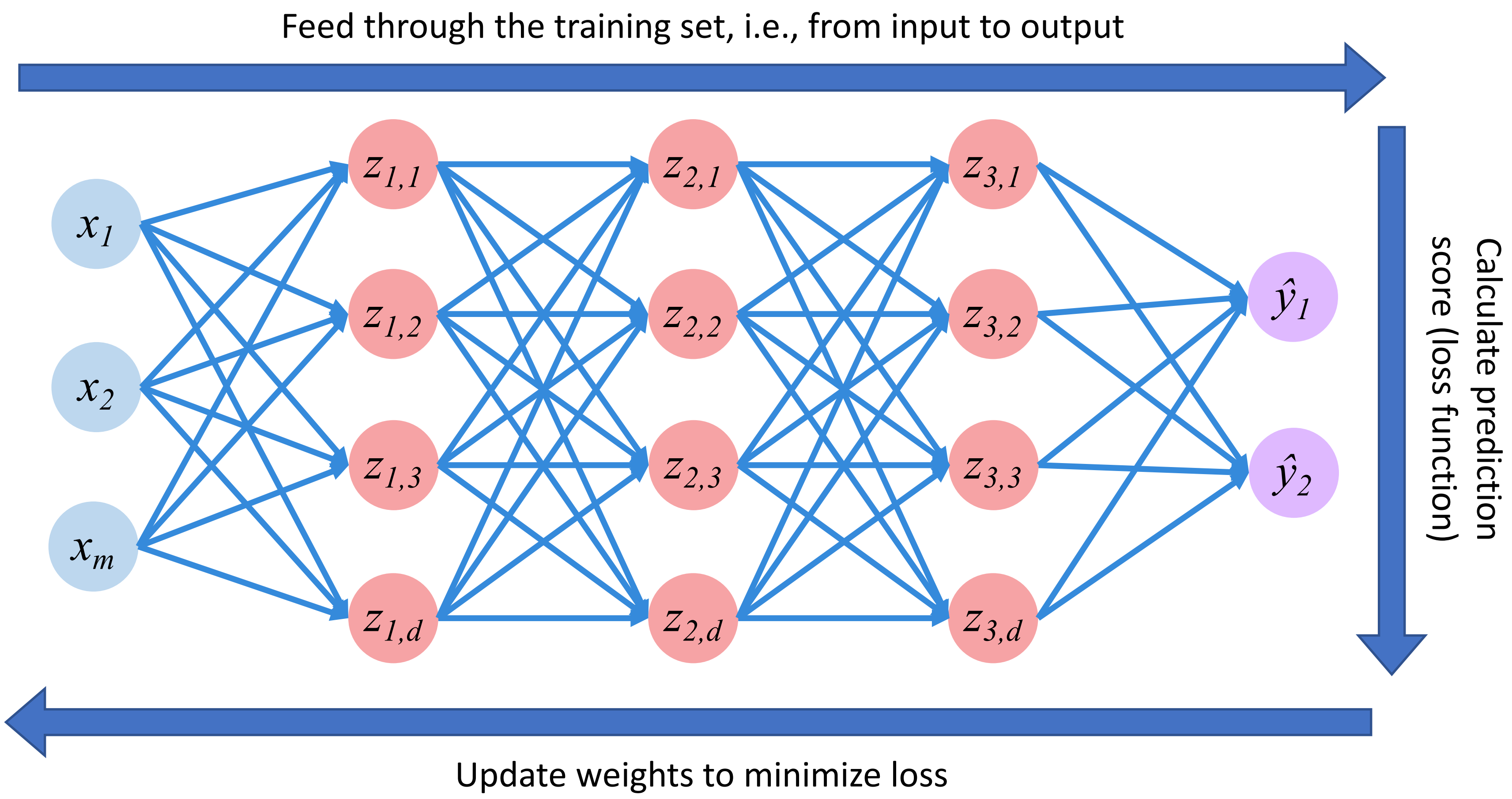
Training a Neural Network



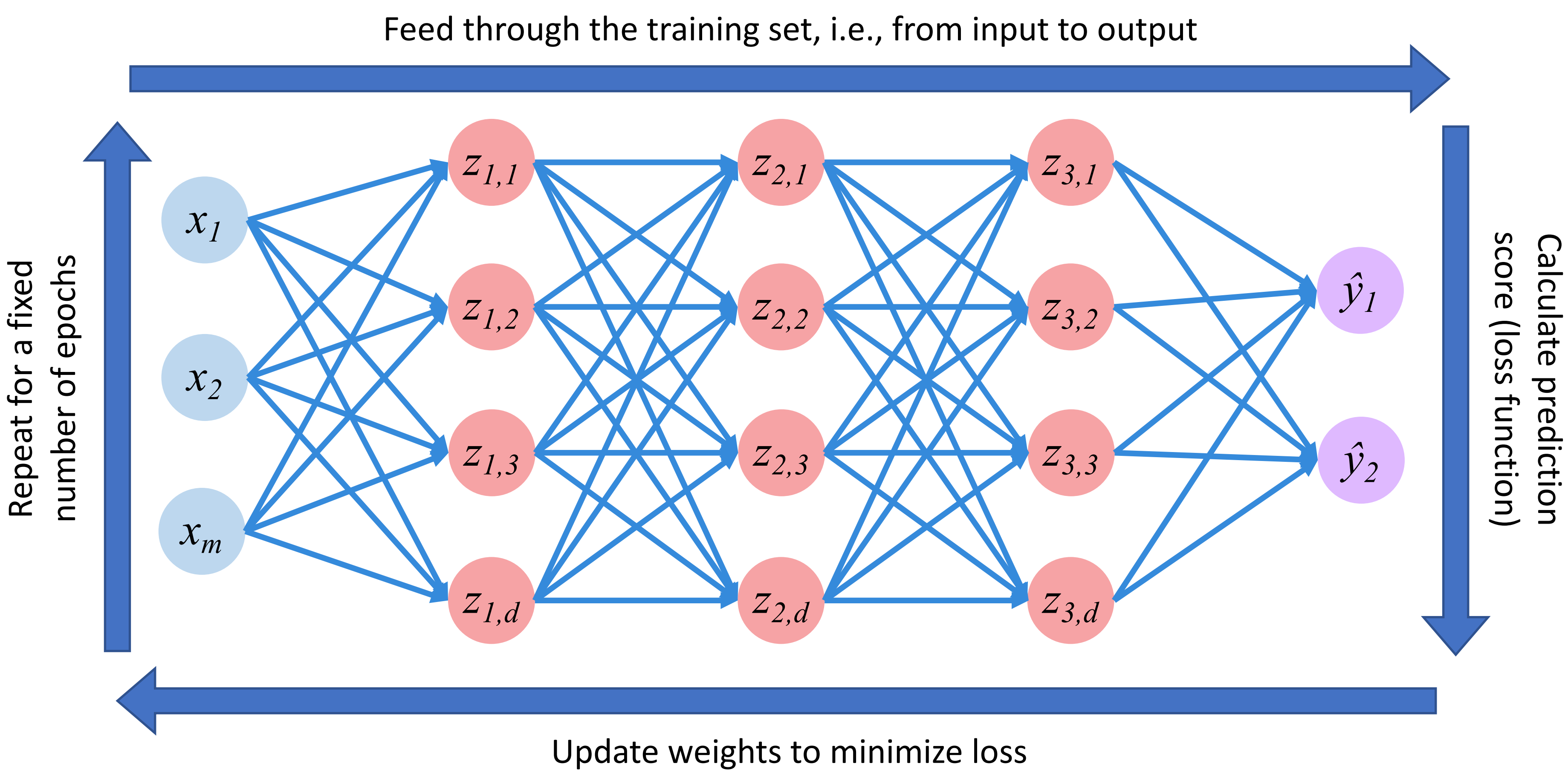
Training a Neural Network



Training a Neural Network



Training a Neural Network



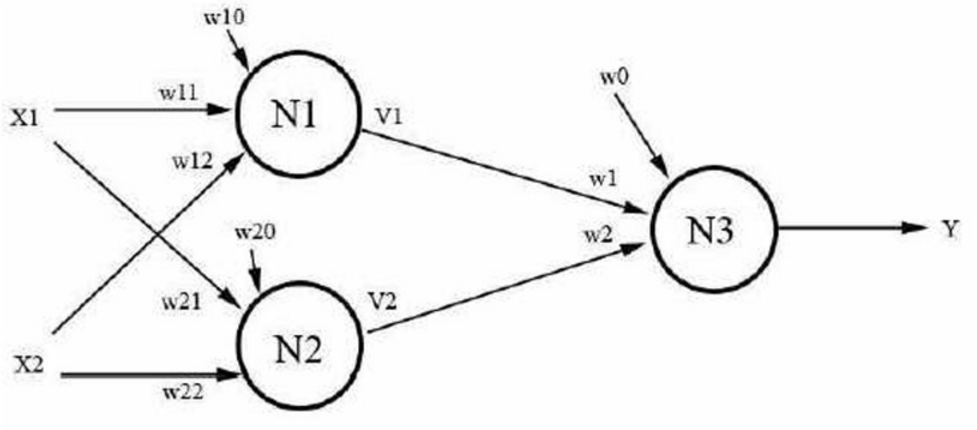
Training a Neural Network

- What type of loss function?
- How to learn and update weights?
- How much training data to use?

Forward propagation and Backpropagation

1. Given inputs and the labels first do forward propagation through the network. It is done by feeding the input to the network. Finally, we will get an output from the last layer. This process is called forward propagation.
2. Calculate the error/loss and gradients of the loss with respect to each of the parameters in the network. The error/loss is the difference between the network generated output and the actual output in the training dataset.
3. Backpropagate the gradients and update/tune the weights using gradient descent technique. This process is called backpropagation.

Quiz



Assume all units are linear. Can this 2-layer network represent decision boundaries that a standard regression model y = b0 + b1x1 + b2x2 + c cannot?

A. Yes
B. No

Assume the hidden units use logistic activation functions and the output unit uses a linear activation unit. Can this network represent non-linear decision boundaries?

A. Yes
B. No

Using logistic activation functions for both hidden and output units, it is possible to approximate any complicated decision surface by combining many piecewise linear decision boundaries. Explain what changes you would need to make to the above network so you could approximate any decision boundary.

Question

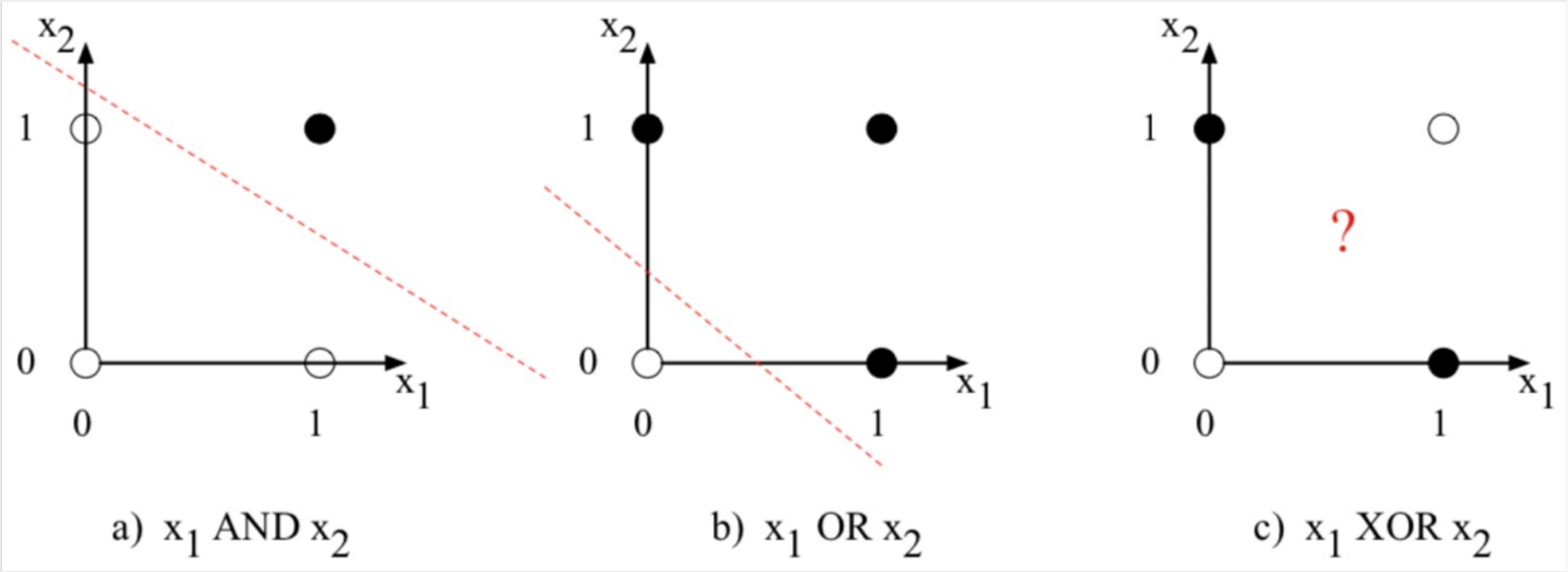
- XOR Function

AND			OR			XOR		
x1	x2	y	x1	x2	y	x1	x2	y
0	0	0	0	0	0	0	0	0
0	1	0	0	1	1	0	1	1
1	0	0	1	0	1	1	0	1
1	1	1	1	1	1	1	1	0

Can you design a neural network to solve XOR function?

XOR Function

- Perceptron is a linear classifier but XOR is not linearly separable
- for a 2D input x0 and x1, the perceptron equation: w1x1 + w2x2 + b = 0 is the equation of a line



XOR Function

- Network of simple linear (perceptron) units cannot solve XOR problem.
  - a network formed by many layers of purely linear units can always be reduced to a single layer of linear units.

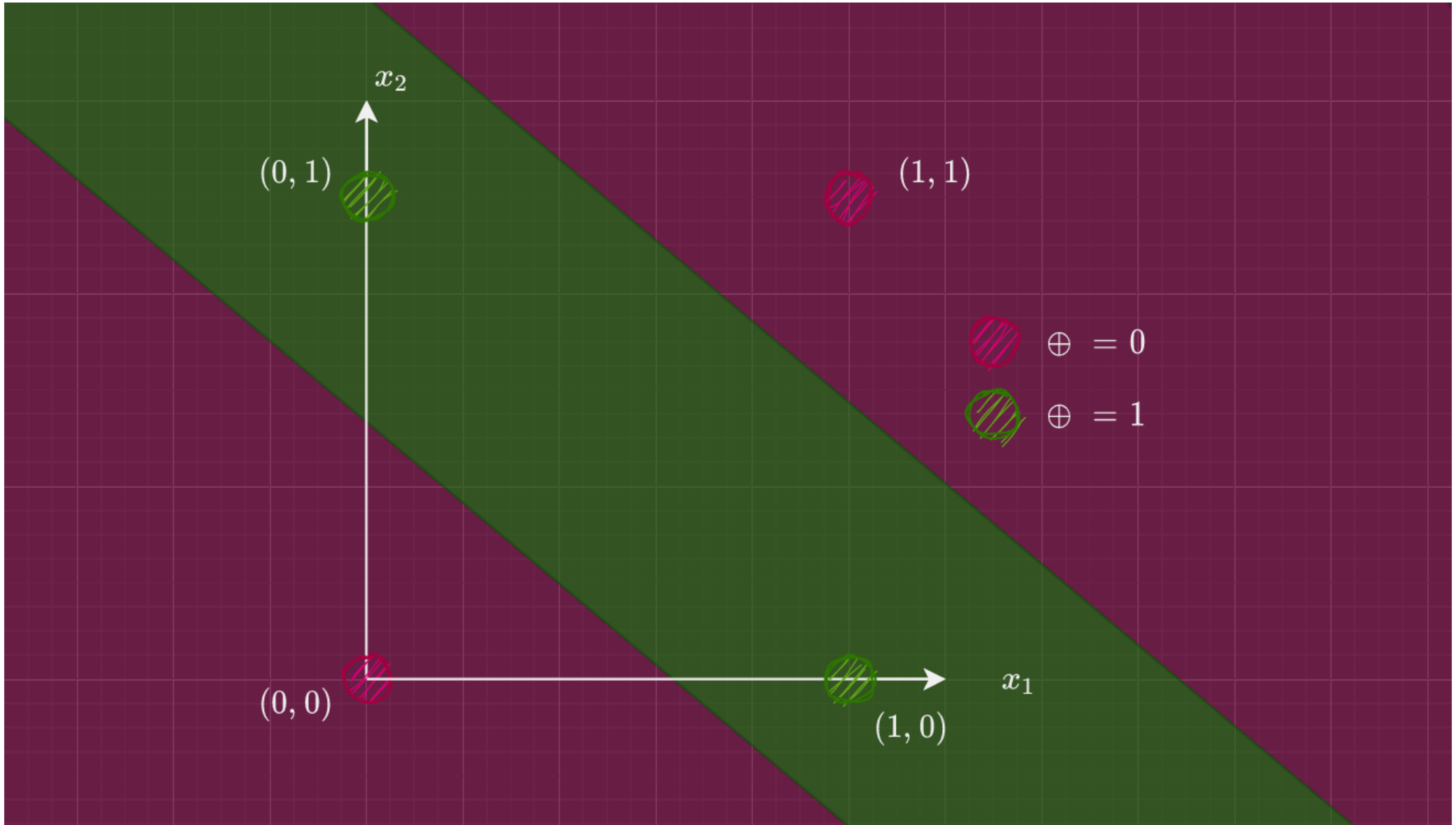
$$
\begin{aligned}
a^{[1]} &= z^{[1]} = W^{[1]} \cdot x + b^{[1]} \\
a^{[2]} &= z^{[2]} = W^{[2]} \cdot a^{[1]} + b^{[2]} \\
&= W^{[2]} \cdot (W^{[1]} \cdot x + b^{[1]}) + b^{[2]} \\
&= (W^{[2]} \cdot W^{[1]}) \cdot x + (W^{[2]} \cdot b^{[1]} + b^{[2]}) \\
&= W' \cdot x + b'
\end{aligned}
$$

no more expressive than logistic regression!

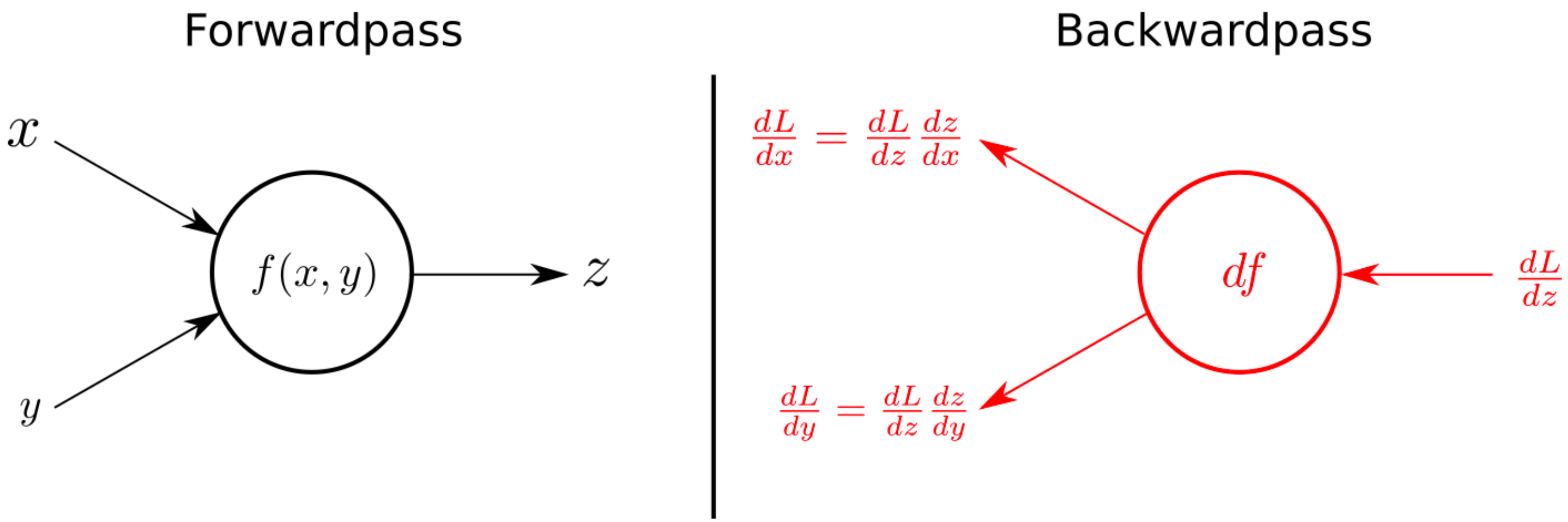
we’ve already shown that a single unit cannot solve the XOR problem

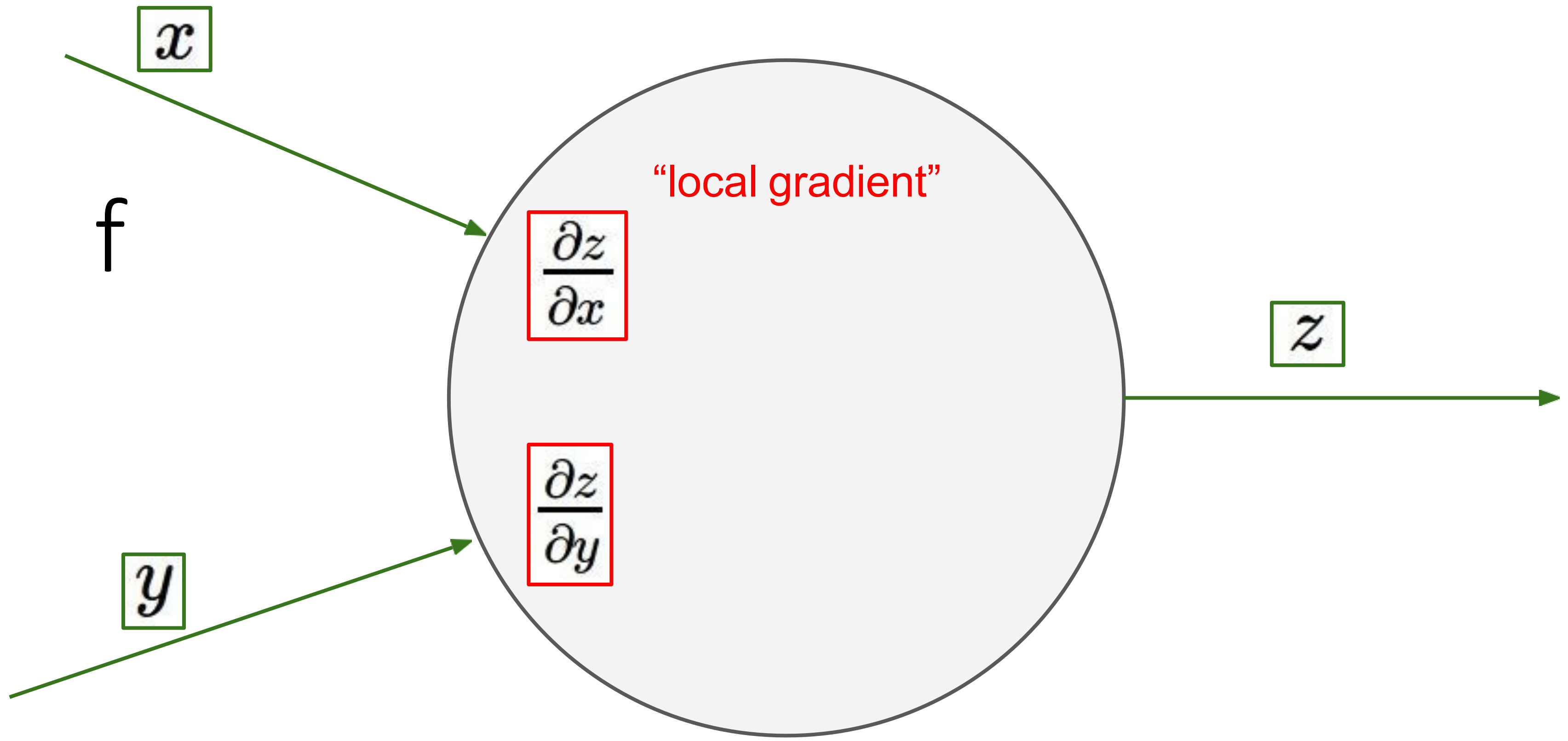
XOR Problem: Solution?

- Hidden layer forms a linearly separable representation for the input.

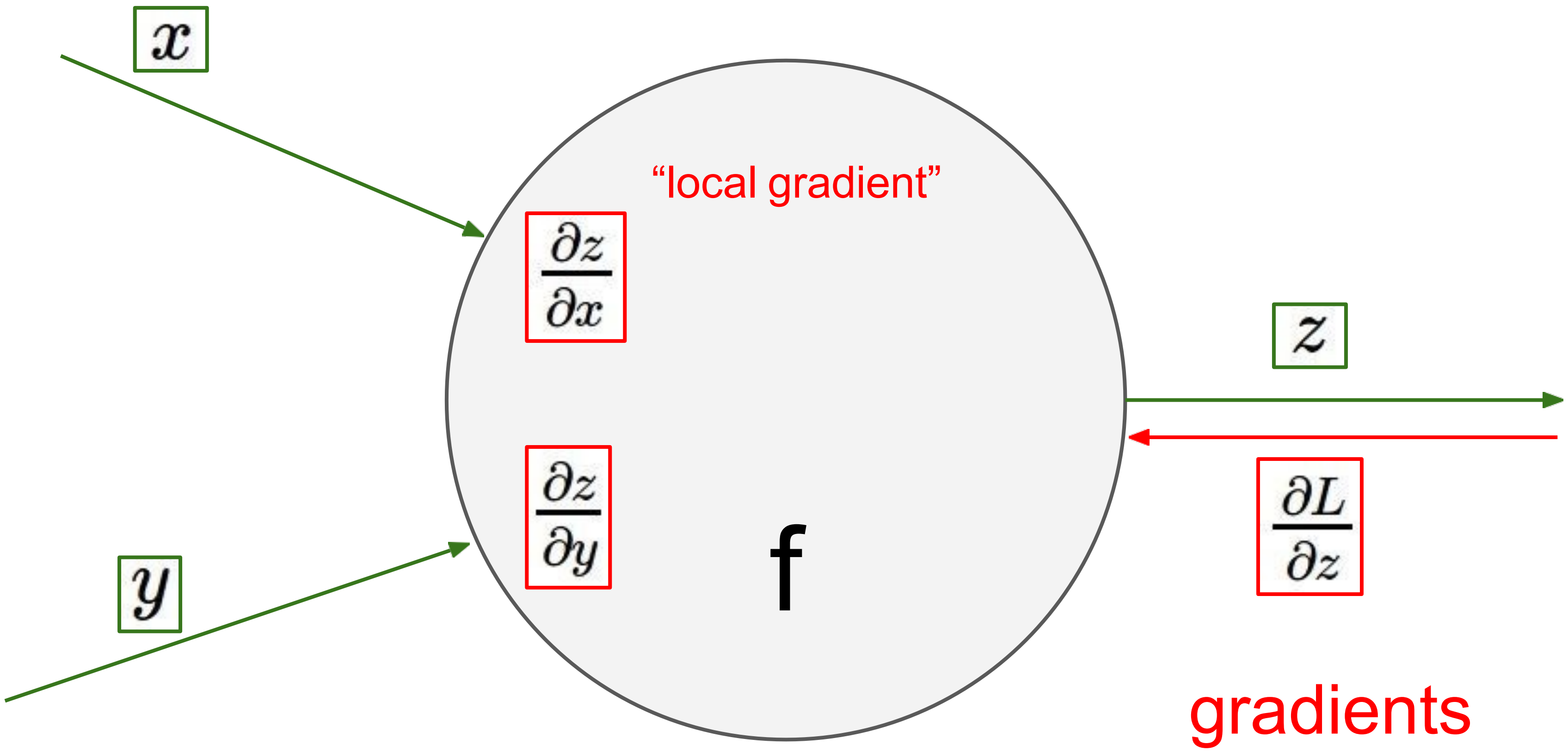


Forward propagation and Backpropagation

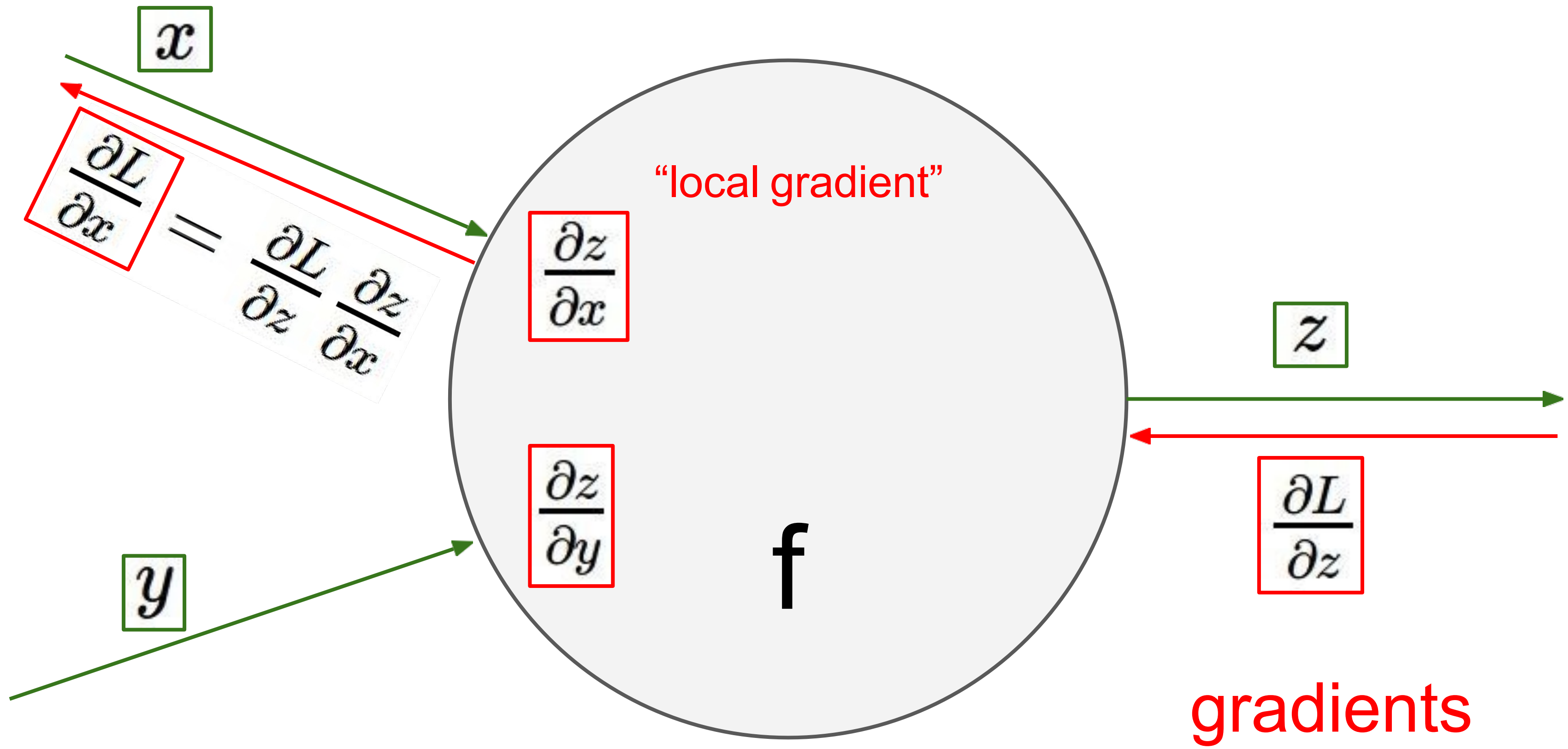




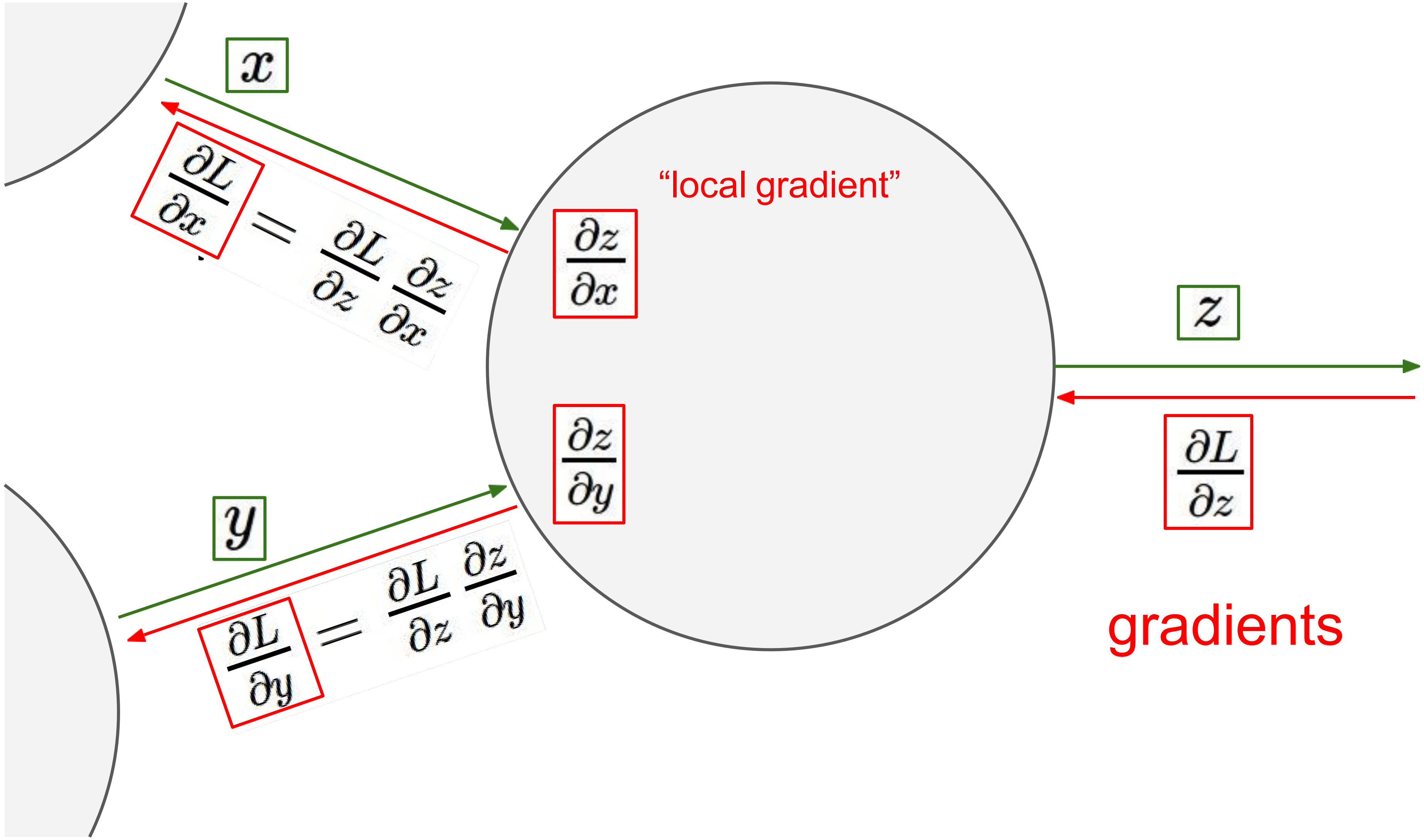
x
f
"local gradient"
$\frac{\partial z}{\partial x}$
$\frac{\partial z}{\partial y}$
y
z



x
y
"local gradient"
$\frac{\partial z}{\partial x}$
$\frac{\partial z}{\partial y}$
f
z
$\frac{\partial L}{\partial z}$
gradients



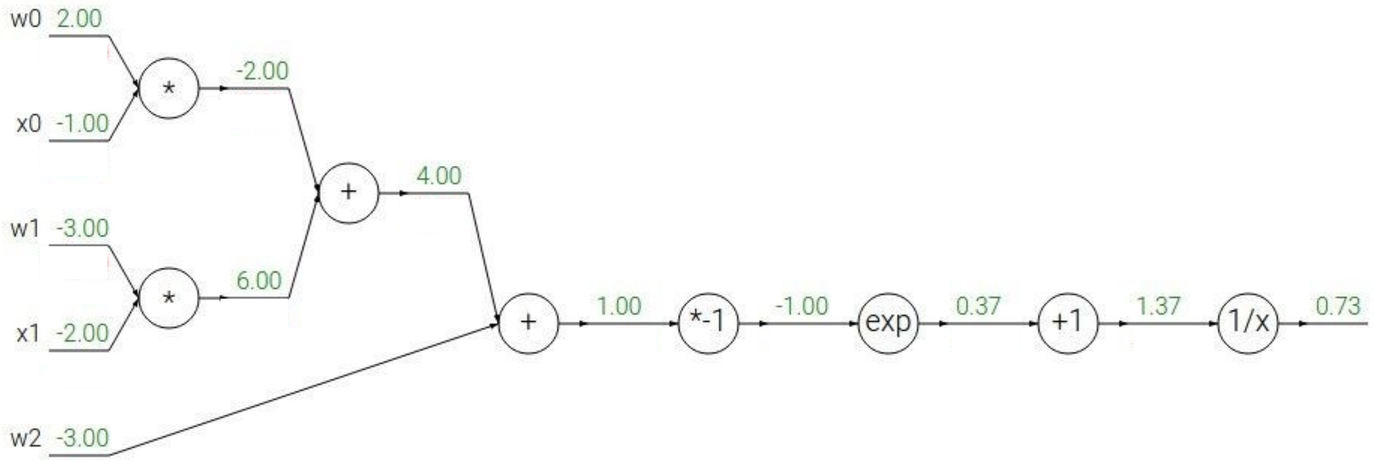
x
$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial z}\frac{\partial z}{\partial x}$
"local gradient"
$\frac{\partial z}{\partial x}$
$\frac{\partial z}{\partial y}$
y
f
z
$\frac{\partial L}{\partial z}$
gradients



x
$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial z}\frac{\partial z}{\partial x}$
"local gradient"
$\frac{\partial z}{\partial x}$
$\frac{\partial z}{\partial y}$
z
$\frac{\partial L}{\partial z}$
y
$\frac{\partial L}{\partial y} = \frac{\partial L}{\partial z}\frac{\partial z}{\partial y}$
gradients

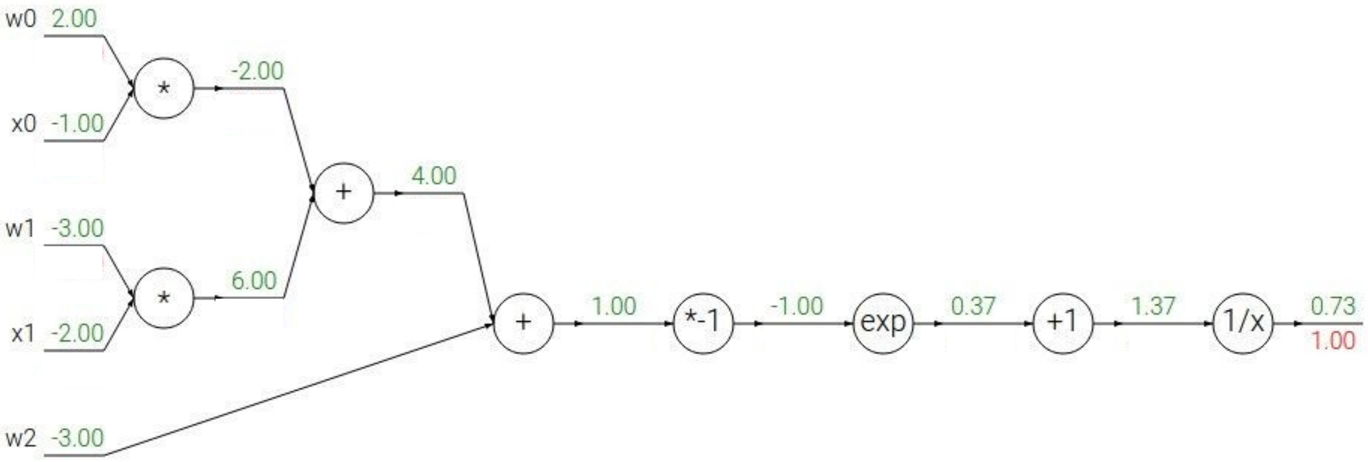
Another example:

$$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



Calculate the gradient of the LOSS with respect to w0, x0, w1, x1 and w2.

Another example: $$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

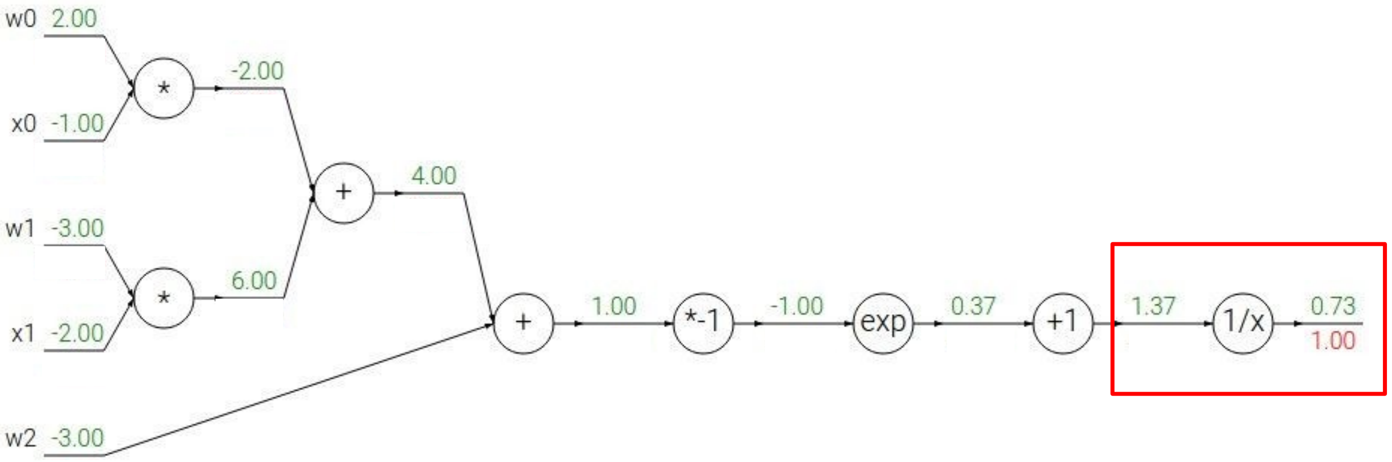
$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

Another example:

$$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

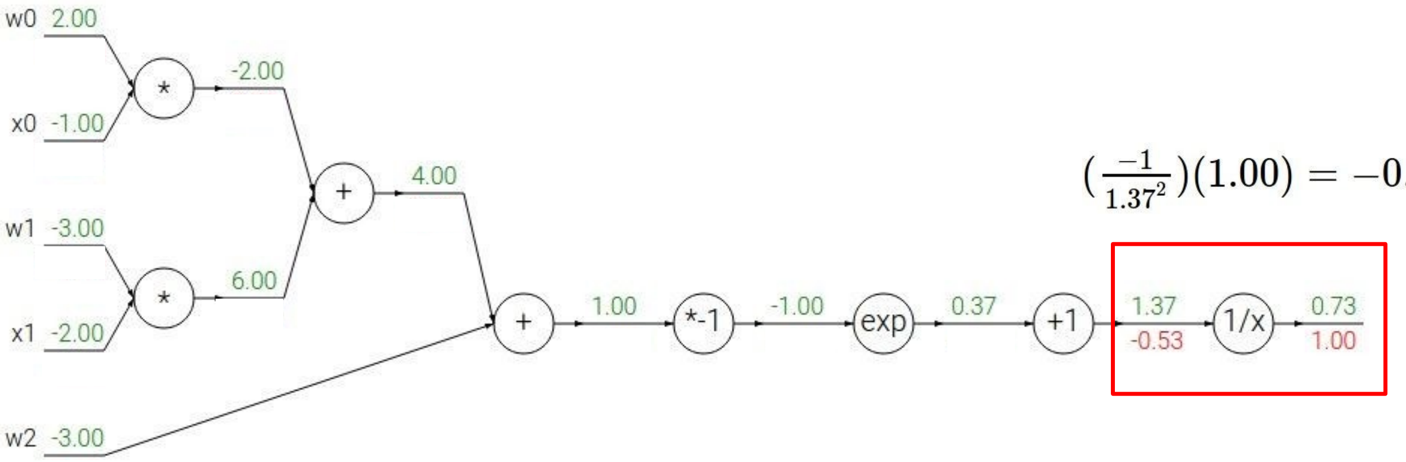
$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$\boxed{f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2}$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

Another example: $$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$

$$\left(\frac{-1}{1.37^2}\right)(1.00) = -0.53$$



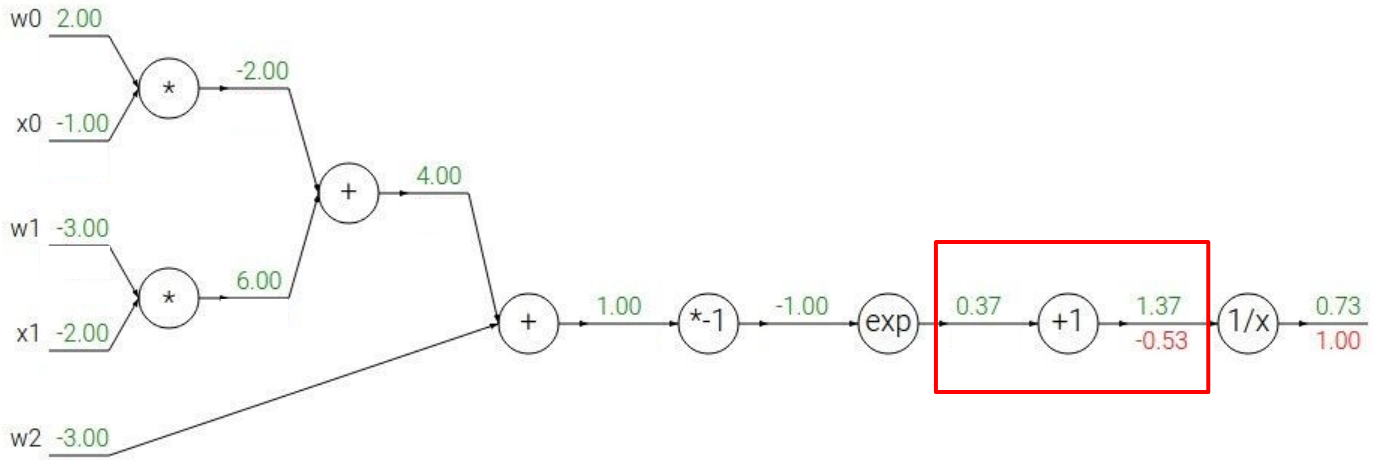
$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$\boxed{f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2}$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

Another example: $$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

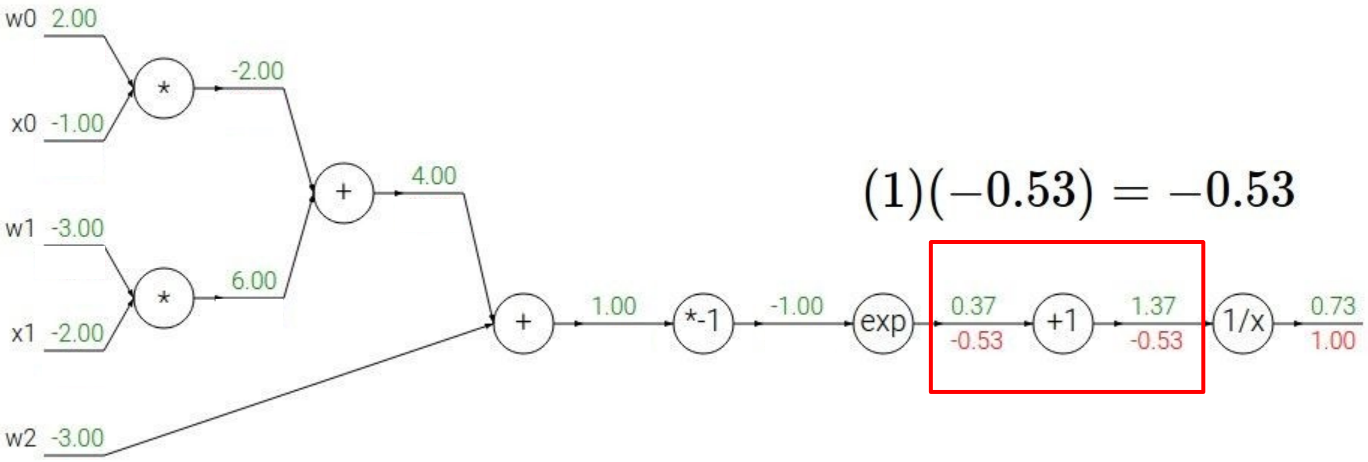
$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

Another example: $$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$

$$(1)(-0.53) = -0.53$$



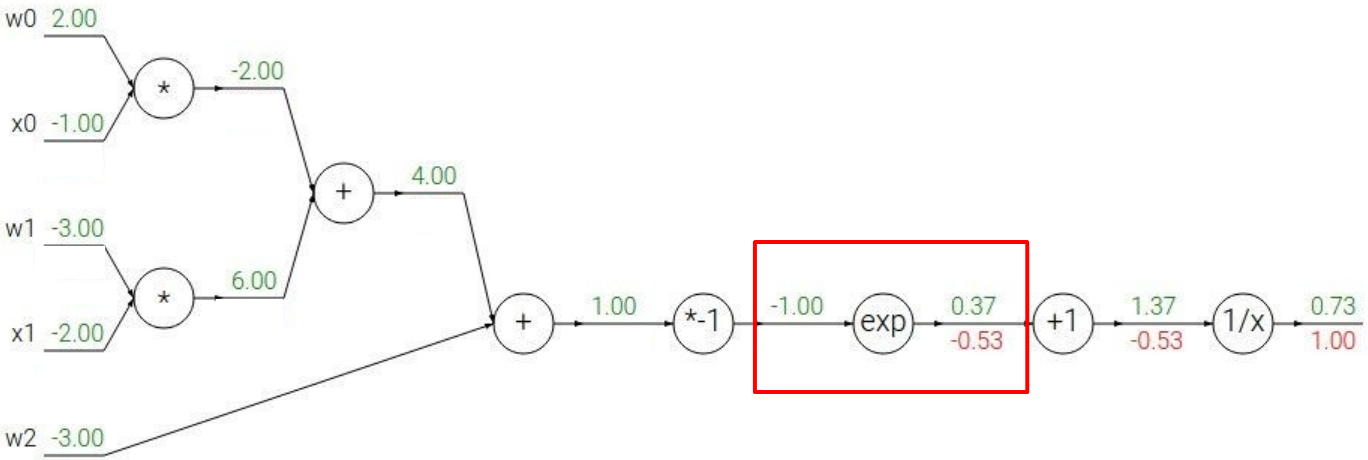
$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

Another example: $$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



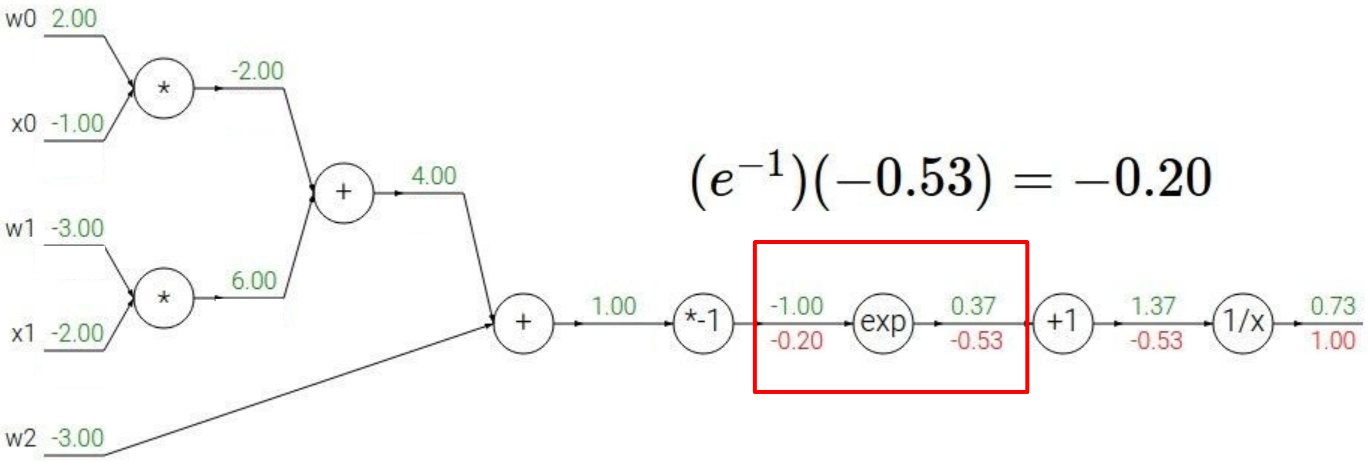
$$\boxed{f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x}$$

$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

Another example: $$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



$$(e^{-1})(-0.53) = -0.20$$

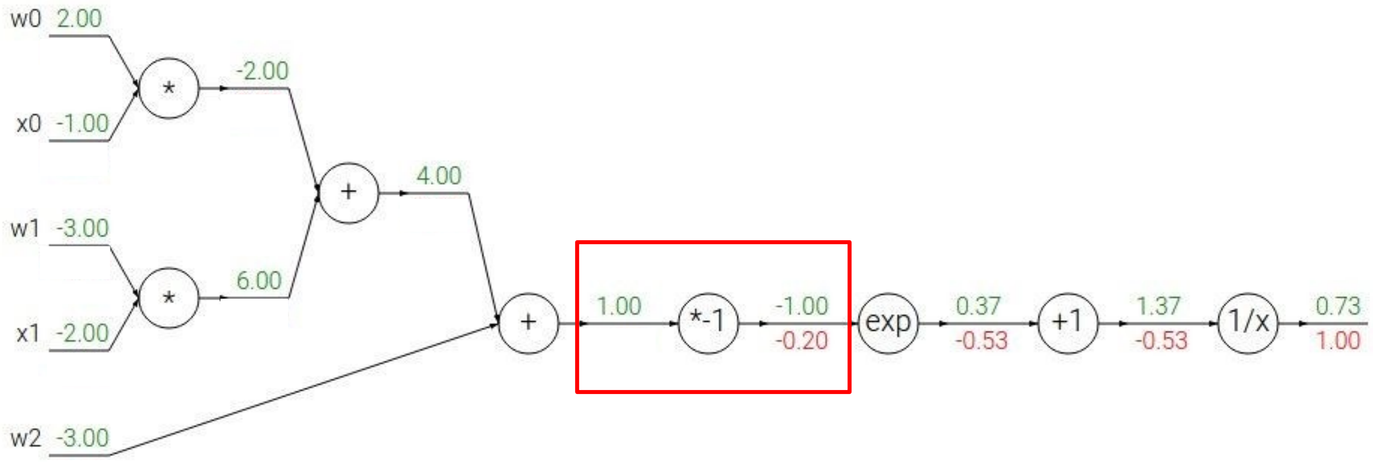
$$\boxed{f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x}$$

$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

Another example: $$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



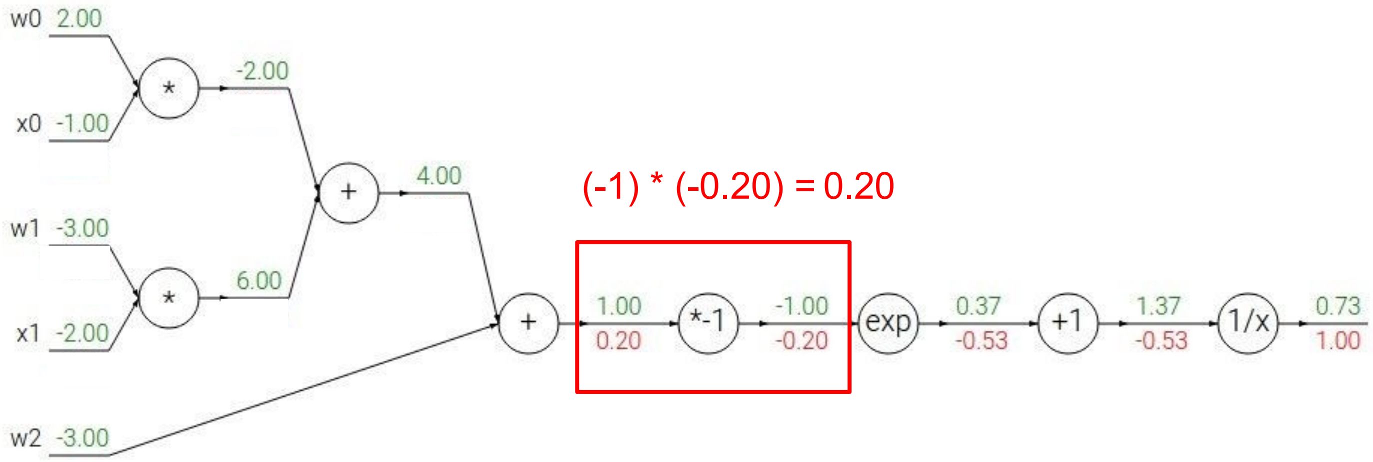
$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

$$\boxed{f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a}$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

Another example: $$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

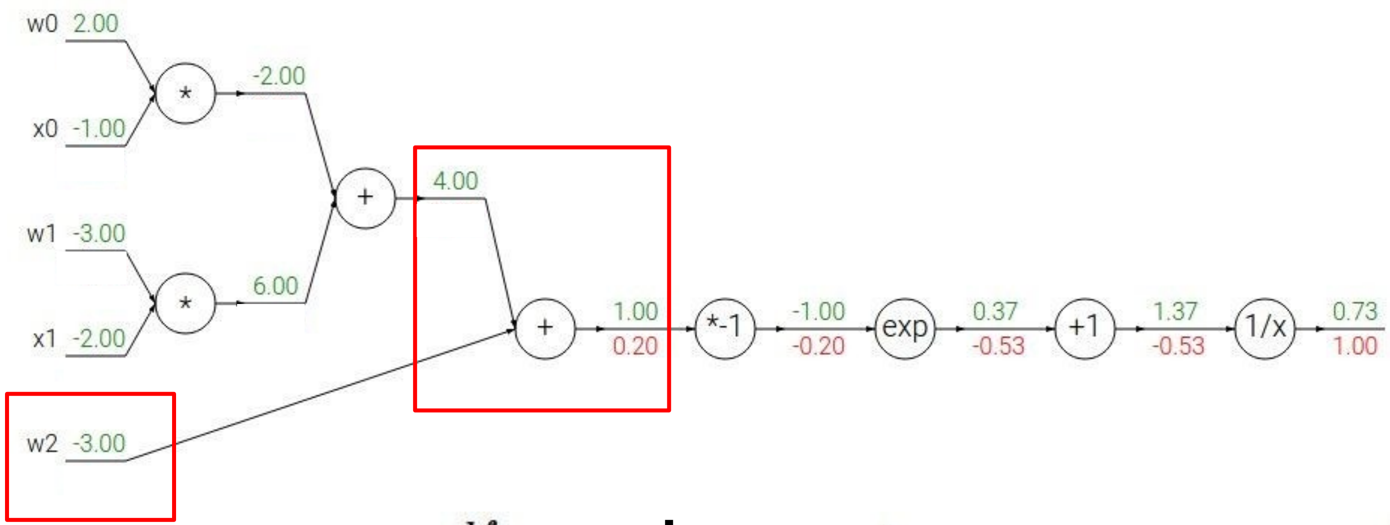
$$\boxed{f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a}$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

Another example:

$$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



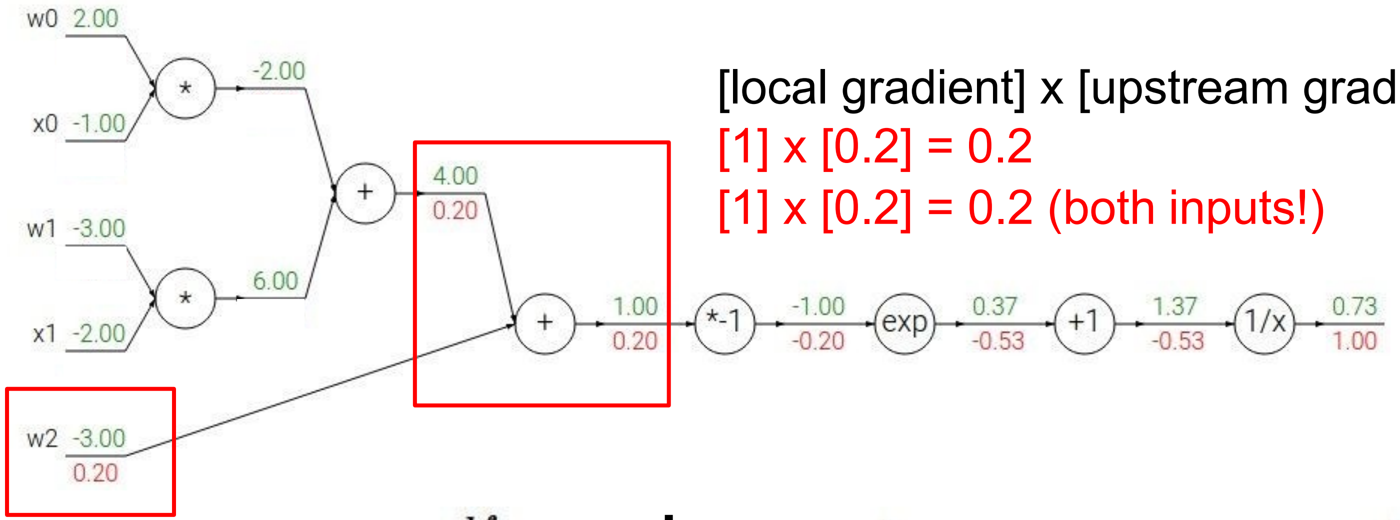
$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

Another example: $$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



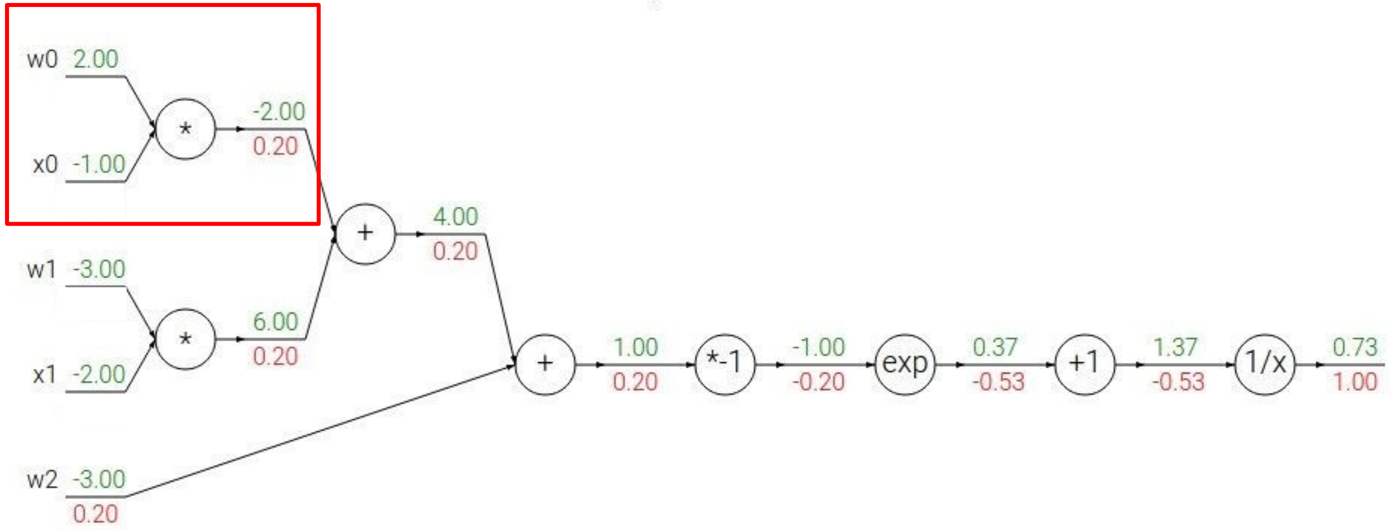
[local gradient] x [upstream gradient]

[1] x [0.2] = 0.2

[1] x [0.2] = 0.2 (both inputs!)

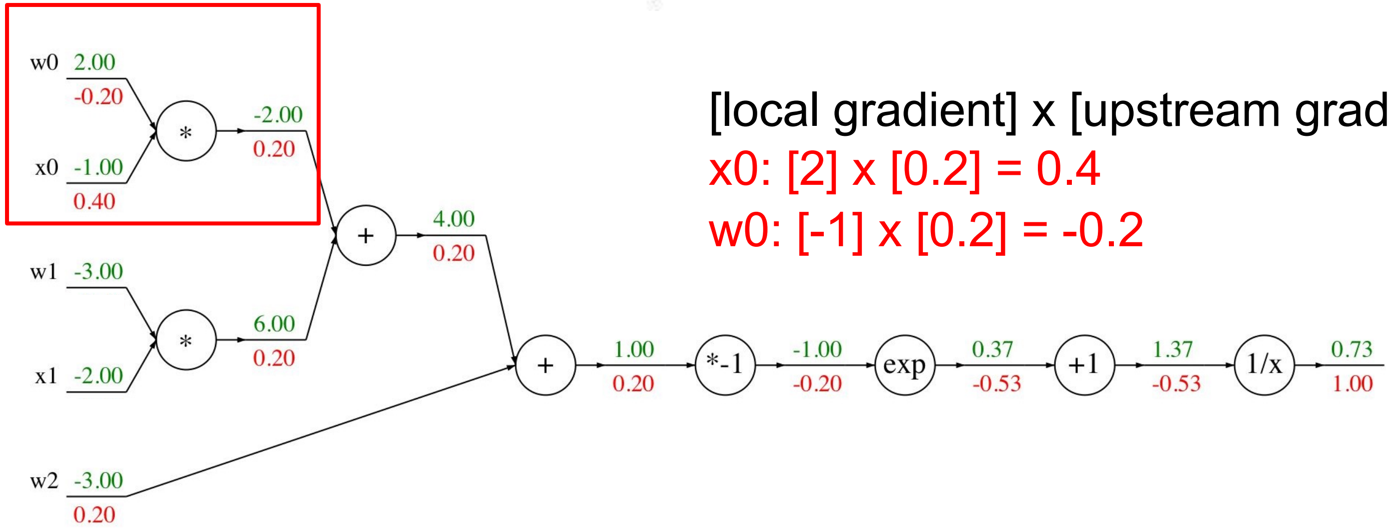
$f(x) = e^x$	$\rightarrow$	$\frac{df}{dx} = e^x$	$f(x) = \frac{1}{x}$	$\rightarrow$	$\frac{df}{dx} = -1/x^2$
$f_a(x) = ax$	$\rightarrow$	$\frac{df}{dx} = a$	$f_c(x) = c + x$	$\rightarrow$	$\frac{df}{dx} = 1$

Another example: $$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



$f(x) = e^x$	$\rightarrow$	$\frac{df}{dx} = e^x$	$f(x) = \frac{1}{x}$	$\rightarrow$	$\frac{df}{dx} = -1/x^2$
$f_a(x) = ax$	$\rightarrow$	$\frac{df}{dx} = a$	$f_c(x) = c + x$	$\rightarrow$	$\frac{df}{dx} = 1$

Another example: $f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$



[local gradient] x [upstream gradient]

x0: [2] x [0.2] = 0.4

w0: [-1] x [0.2] = -0.2

$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$

$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$

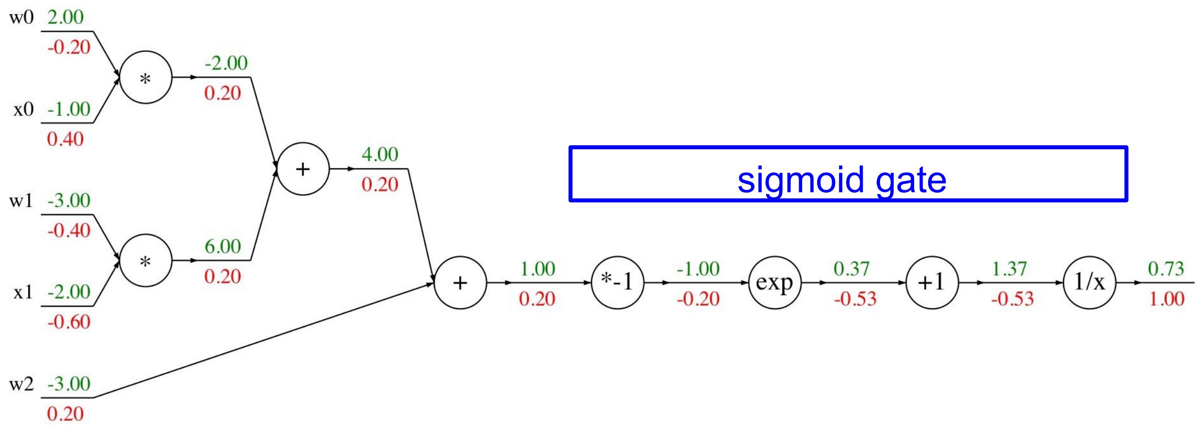
$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$

$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$

$$f(w,x) = \frac{1}{1 + e^{-(w_0 x_0 + w_1 x_1 + w_2)}}$$

$$\boxed{\sigma(x) = \frac{1}{1 + e^{-x}}}$$ sigmoid function

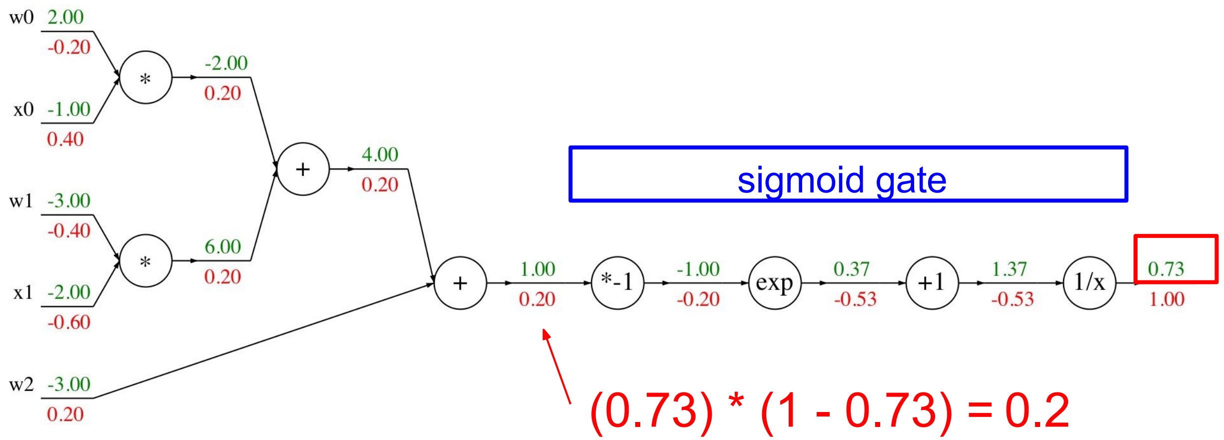
$$\frac{d\sigma(x)}{dx} = \frac{e^{-x}}{(1+e^{-x})^2} = \left(\frac{1 + e^{-x} - 1}{1 + e^{-x}}\right)\left(\frac{1}{1+e^{-x}}\right) = (1 - \sigma(x))\,\sigma(x)$$



$$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$

$$\boxed{\sigma(x) = \frac{1}{1+e^{-x}}}$$ sigmoid function

$$\frac{d\sigma(x)}{dx} = \frac{e^{-x}}{(1+e^{-x})^2} = \left(\frac{1+e^{-x}-1}{1+e^{-x}}\right)\left(\frac{1}{1+e^{-x}}\right) = (1-\sigma(x))\,\sigma(x)$$



(0.73) * (1 - 0.73) = 0.2

Logistic regression: binary classification

The loss function is -

$$: L(w) = -\frac{1}{m} \sum_{i=1}^{m} [y^{(i)} logh(x^{(i)}) + (1 - y^{(i)}) log(1 - h(x^{(i)}))]$$

$x^{(i)}$ is a vector for all x_j (j=0,1, … , n), and $y^{(i)}$ is the target value for this example.

$$h(x) = \frac{1}{1 + e^{-w^T x}}$$

Softmax

- Use softmax for multi-class classification
  - K is the number classes

$$P(y = j \mid z^{(i)}) = \phi_{softmax}(z^{(i)}) = \frac{e^{z^{(i)}}}{\sum_{j=0}^{k} e^{z_k^{(i)}}},$$

where we define the net input z as

$$z = w_0 x_0 + w_1 x_1 + \ldots + w_m x_m = \sum_{l=0}^{m} w_l x_l = \mathbf{w}^T \mathbf{x}.$$

The loss function is $H(y, p) = -\sum_i y_i log(p_i)$

Softmax vs Sigmoid function in Logistic classifier?

- In the two-class logistic regression, the predicted probabilities are as follows, using the sigmoid function:

$$\Pr(Y_i = 0) = \frac{e^{-\beta.\mathbf{X}_i}}{1 + e^{-\beta_0 \cdot \mathbf{X}_i}}$$

$$\Pr(Y_i = 1) = 1 - \Pr(Y_i = 0) = \frac{1}{1 + e^{-\beta.\mathbf{X}_i}}$$

In the multiclass logistic regression, with K classes, the predicted probabilities are as follows, using the softmax function:

$$\Pr(Y_i = k) = \frac{e^{\beta_k \cdot \mathbf{X}_i}}{\sum_{0 \le c \le K} e^{\beta_c \cdot \mathbf{X}_i}}$$

Softmax vs Sigmoid function in Logistic classifier?

- One can observe that the softmax function is an extension of the sigmoid function to the multiclass case, as explained below. Let's look at the multiclass logistic regression, with K=2 classes:

$$\Pr(Y_i = 0) = \frac{e^{\boldsymbol{\beta}_0 \cdot \mathbf{X}_i}}{\sum_{0 \leq c \leq K} e^{\boldsymbol{\beta}_c \cdot \mathbf{X}_i}} = \frac{e^{\boldsymbol{\beta}_0 \cdot \mathbf{X}_i}}{e^{\boldsymbol{\beta}_0 \cdot \mathbf{X}_i} + e^{\boldsymbol{\beta}_1 \cdot \mathbf{X}_i}} = \frac{e^{(\boldsymbol{\beta}_0 - \boldsymbol{\beta}_1) \cdot \mathbf{X}_i}}{e^{(\boldsymbol{\beta}_0 - \boldsymbol{\beta}_1) \cdot \mathbf{X}_i} + 1} = \frac{e^{-\boldsymbol{\beta} . \mathbf{X}_i}}{1 + e^{-\boldsymbol{\beta} \cdot \mathbf{X}_i}}$$

$$\Pr(Y_i = 1) = \frac{e^{\boldsymbol{\beta}_1 \cdot \mathbf{X}_i}}{\sum_{0 \leq c \leq K} e^{\boldsymbol{\beta}_c \cdot \mathbf{X}_i}} = \frac{e^{\boldsymbol{\beta}_1 \cdot \mathbf{X}_i}}{e^{\boldsymbol{\beta}_0 \cdot \mathbf{X}_i} + e^{\boldsymbol{\beta}_1 \cdot \mathbf{X}_i}} = \frac{1}{e^{(\boldsymbol{\beta}_0 - \boldsymbol{\beta}_1) \cdot \mathbf{X}_i} + 1} = \frac{1}{1 + e^{-\boldsymbol{\beta} . \mathbf{X}_i}}$$

where $\boldsymbol{\beta} = -(\boldsymbol{\beta}_0 - \boldsymbol{\beta}_1)$

Types of Loss Functions

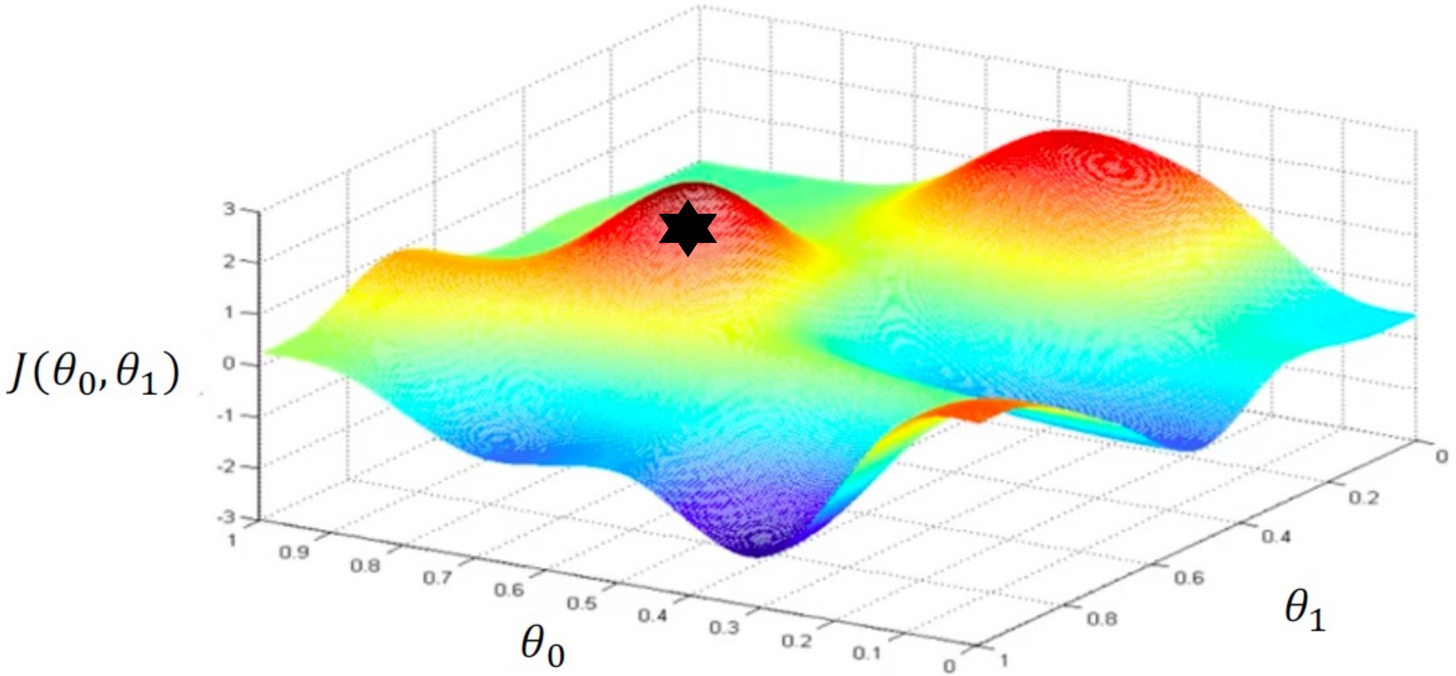
- For Regression tasks (continuous values)
  - Mean Absolute Error
  - Mean Squared Error

- For Classification tasks (discrete categories)
  - Categorical Cross Entropy
  - Binary Cross Entropy

- And various others in Keras

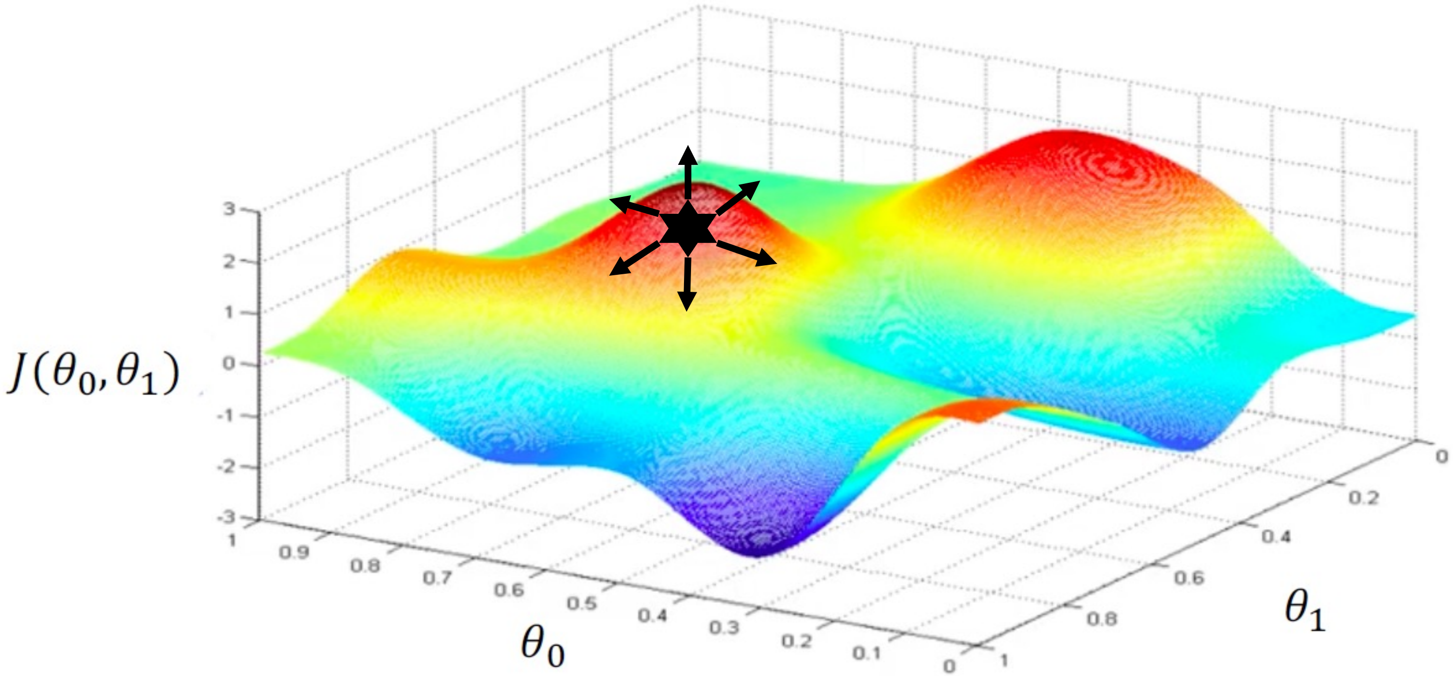
Gradient Descent

- Standard approach for learning weights



Gradient Descent

- Standard approach for learning weights



Gradient Descent

- Standard approach for learning weights
  - What about the learning rate?