Machine Learning

CS 539
Worcester Polytechnic Institute
Department of Computer Science
Instructor: Prof. Kyumin Lee

Project Teams

- Yu-Chi Liang, William Ryan, Riley Blair, and Stephen Fanning
- Chris Lee, Andrew Kerekon, Amulya Mohan, Alex Siracusa, and Sulaiman Moukheiber
- Vagmi Bhagavathula, Deepti Gosukonda, Adina Palayoor, Bishoy Soliman Hanna, and Jared Chan
- Sreeram Marimuthu, Oruganty Nitya Phani Santosh, Sarah Olson, and Thomas Pianka
- Shubham Dashrath Wagh, Atharva Pradip Kulkarni, Amit Virchandbhai Prajapati, Niveditha Narasimha Murthy
- Aria Yan, Alisha Peeriz, Nupur Kalkumbe, Pavan Antala, Rutuja Madhumilind Dongre
- Noushin Khosravi Largani, Jinqin Xiong, Kexin Li, Ronit Kapoor, and Yiqun Duan
- Phil Brush, Liam Hall, Jared Morgan, Alex
- Khang Luu, Austin Aguirre, Brock Dubey, Ivan Klevanski,
- Adhiraj, Karl, Shariq Madha, Yue Bao, Vasilli Gorbunov
- Edward Smith, Michael Alicea, Cutter Beck, Blake Bruell, Anushka Bangal
- Rahul Chhatbar, Sonu Tejwani, Deep Suchak, Shoan Bhatambare
- Daniel Fox, Bijesh Shrestha, Chad Hucey, Aayush Sangani, Ivan Lim
- Devesh Bhangale, Shipra Poojary, Jagruti Chitte, Parth Shroff, Saurabh Pande
- Alessandra Serpes, Khushita Joshi, Sanjeeeth Nagappa Chakrasali, Shaun Noronha, Sankalp Vyas
- Rohan Rana, Theo Coppola, Olivia Raisbeck

So far, 72 students formed teams.

Previous Class…

Linear Basis Function Models

Previous Class…

Linear Basis Function Models

Regularization

Perceptron

HW2

- https://canvas.wpi.edu/courses/57384/assignments/339358

Logistic Regression

(models probability of output in terms of input)

Intuition Behind the Objective (log loss)

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

- 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

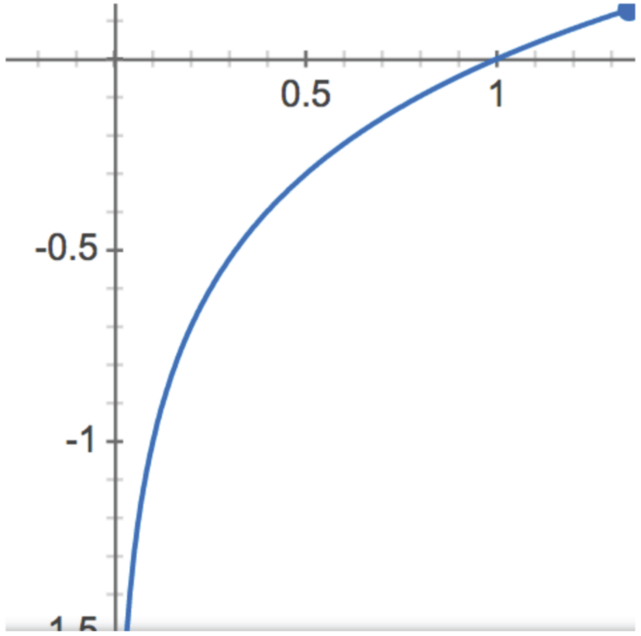
$$J(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n}\text{cost}\left(h_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}), y^{(i)}\right)$$

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

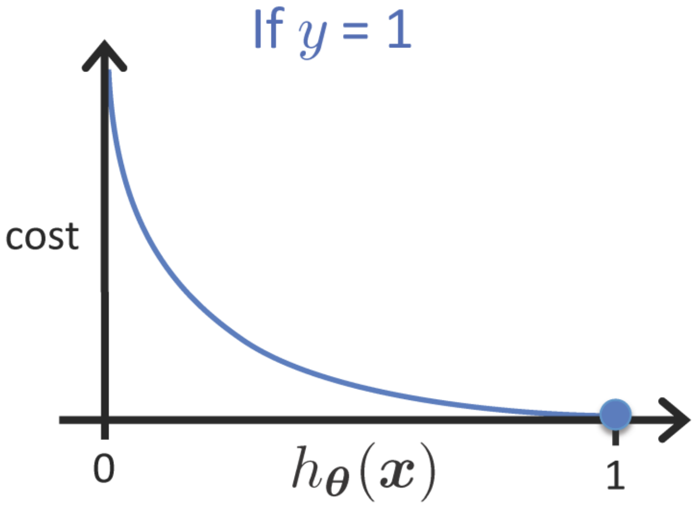
$$\text{cost}\left(h_{\boldsymbol{\theta}}(\boldsymbol{x}), y\right) = \begin{cases} -\log(h_{\boldsymbol{\theta}}(\boldsymbol{x})) & \text{if } y = 1 \\ -\log(1 - h_{\boldsymbol{\theta}}(\boldsymbol{x})) & \text{if } y = 0 \end{cases}$$

Aside: Recall the plot of log(z)



Intuition Behind the Objective

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

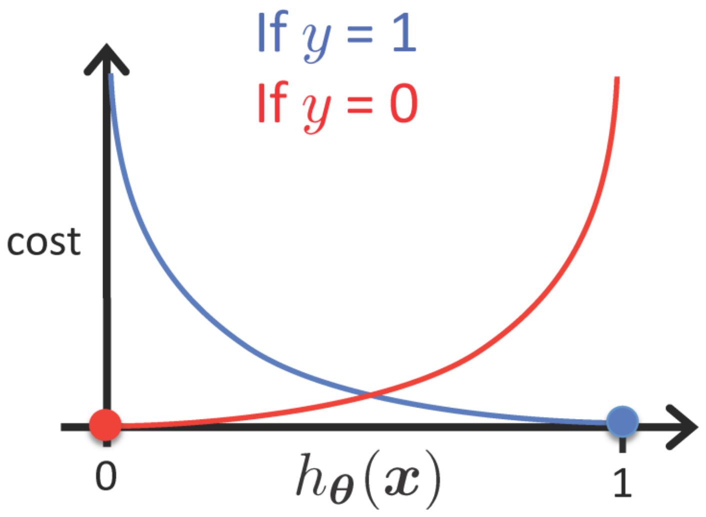


If y = 1

- 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

Intuition Behind the Objective

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



If y = 0

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

Regularized Logistic Regression

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

- We can regularize logistic regression exactly as before:

$$\begin{aligned} J_{\text{regularized}}(\boldsymbol{\theta}) &= J(\boldsymbol{\theta}) + \frac{\lambda}{2}\sum_{j=1}^{d}\theta_j^2 \\ &= J(\boldsymbol{\theta}) + \frac{\lambda}{2}\|\boldsymbol{\theta}_{[1:d]}\|_2^2 \end{aligned}$$

Gradient Descent for Logistic Regression

$$J_{\text{reg}}(\boldsymbol{\theta}) = -\frac{1}{n}\sum_{i=1}^{n}\left[y^{(i)} \log h_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}) + \left(1 - y^{(i)}\right) \log\left(1 - h_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)})\right)\right] + \frac{\lambda}{2}\|\boldsymbol{\theta}_{[1:d]}\|_2^2$$

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

- Initialize $\boldsymbol{\theta}$
- Repeat until convergence

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

simultaneous update for j = 0 ... d

Use the natural logarithm ($\ln = \log_e$) to cancel with the exp() in $h_{\boldsymbol{\theta}}(\boldsymbol{x})$

Gradient Descent for Logistic Regression

$$J_{\text{reg}}(\boldsymbol{\theta}) = -\frac{1}{n}\sum_{i=1}^{n}\left[y^{(i)}\log h_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}) + \left(1 - y^{(i)}\right)\log\left(1 - h_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)})\right)\right] + \frac{\lambda}{2}\|\boldsymbol{\theta}_{[1:d]}\|_2^2$$

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

- Initialize $\boldsymbol{\theta}$
- Repeat until convergence (simultaneous update for j = 0 ... d)

$$\theta_0 \leftarrow \theta_0 - \alpha\frac{1}{n}\sum_{i=1}^{n}\left(h_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right) - y^{(i)}\right)$$

$$\theta_j \leftarrow \theta_j - \alpha\frac{1}{n}\sum_{i=1}^{n}\left(h_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right) - y^{(i)}\right)x_j^{(i)} - \alpha\lambda\theta_j$$

https://stats.stackexchange.com/questions/278771/how-is-the-cost-function-from-logistic-regression-derivated

Gradient Descent for Logistic Regression

- Initialize $\boldsymbol{\theta}$
- Repeat until convergence (simultaneous update for j = 0 ... d)

$$\theta_0 \leftarrow \theta_0 - \alpha \frac{1}{n} \sum_{i=1}^{n} \left(h_{\boldsymbol{\theta}} \left(\boldsymbol{x}^{(i)} \right) - y^{(i)} \right)$$

$$\theta_j \leftarrow \theta_j - \alpha \frac{1}{n} \sum_{i=1}^{n} \left(h_{\boldsymbol{\theta}} \left(\boldsymbol{x}^{(i)} \right) - y^{(i)} \right) x_j^{(i)} - \alpha \lambda \theta_j$$

This looks IDENTICAL to linear regression!!!

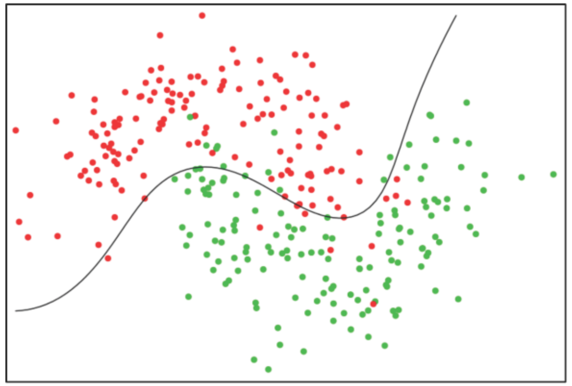
- However, the form of the model is very different:

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

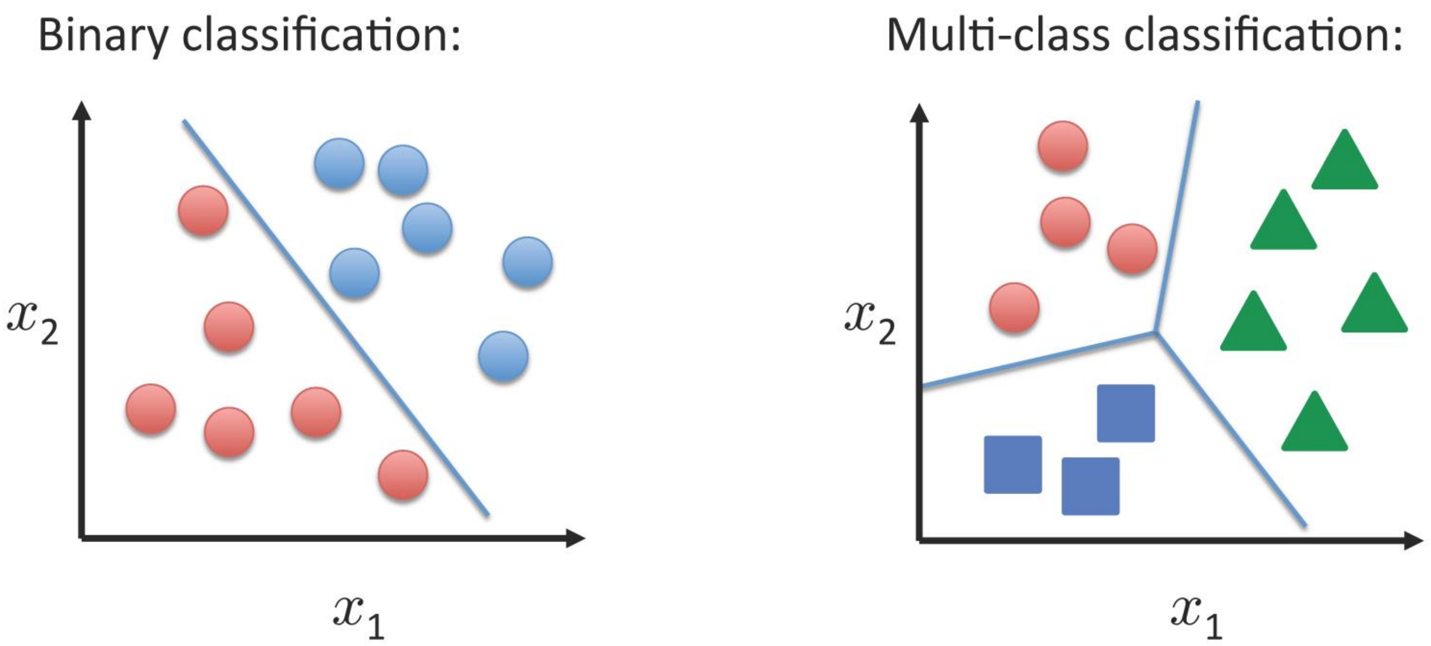
Non-Linear Decision Boundary

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

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



Multi-Class Classification

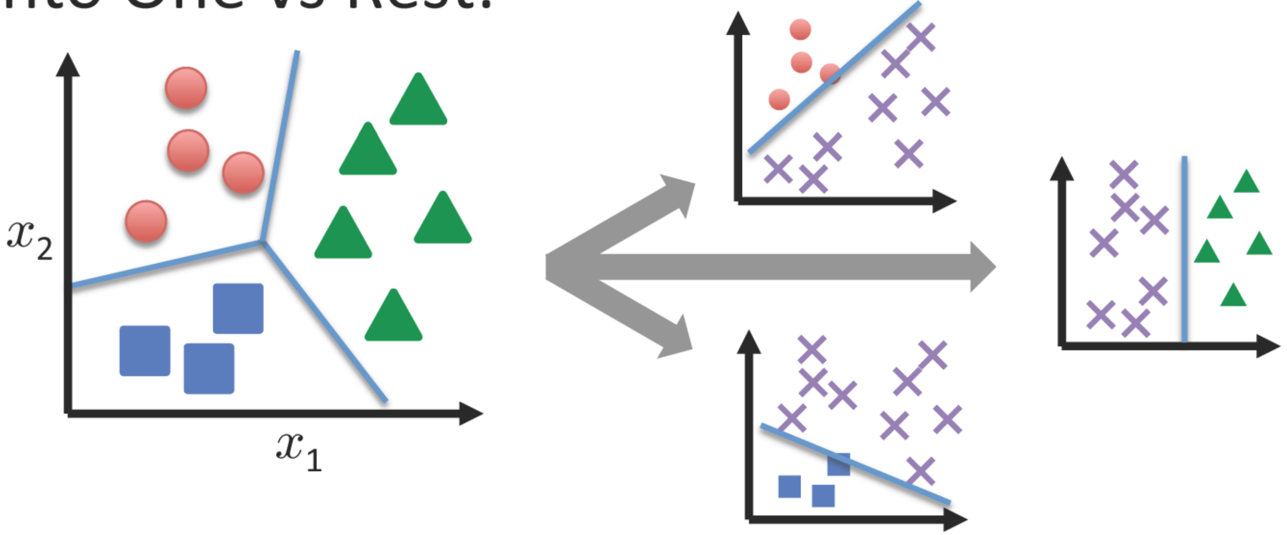


Disease diagnosis: healthy / cold / flu / pneumonia

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 $y = i$ with

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

Multi-Class Logistic Regression

- For 2 classes:

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \frac{1}{1 + \exp(-\boldsymbol{\theta}^\mathsf{T}\boldsymbol{x})} = \frac{\exp(\boldsymbol{\theta}^\mathsf{T}\boldsymbol{x})}{\underbrace{\boxed{1}}_{\text{weight assigned to } y = 0} + \underbrace{\boxed{\exp(\boldsymbol{\theta}^\mathsf{T}\boldsymbol{x})}}_{\text{weight assigned to } y = 1}}$$

- For C classes {1, ..., C}:

$$p(y = c \mid \boldsymbol{x}; \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_C) = \frac{\exp(\boldsymbol{\theta}_c^\mathsf{T}\boldsymbol{x})}{\sum_{c=1}^{C} \exp(\boldsymbol{\theta}_c^\mathsf{T}\boldsymbol{x})}$$

  – Called the **softmax** function (normalized exponential)

Implementing Multi-Class Logistic Regression

- Use $h_c(\boldsymbol{x}) = \dfrac{\exp(\boldsymbol{\theta}_c^\mathsf{T} \boldsymbol{x})}{\sum_{c=1}^{C} \exp(\boldsymbol{\theta}_c^\mathsf{T} \boldsymbol{x})}$ as the model for class c

- Gradient descent simultaneously updates all parameters for all models
  - Same derivative as before, just with the above $h_c(\boldsymbol{x})$

- Predict class label as the most probable label

$$\max_c h_c(\boldsymbol{x})$$

Gradient Descent vs.
Stochastic Gradient Descent

Gradient Descent

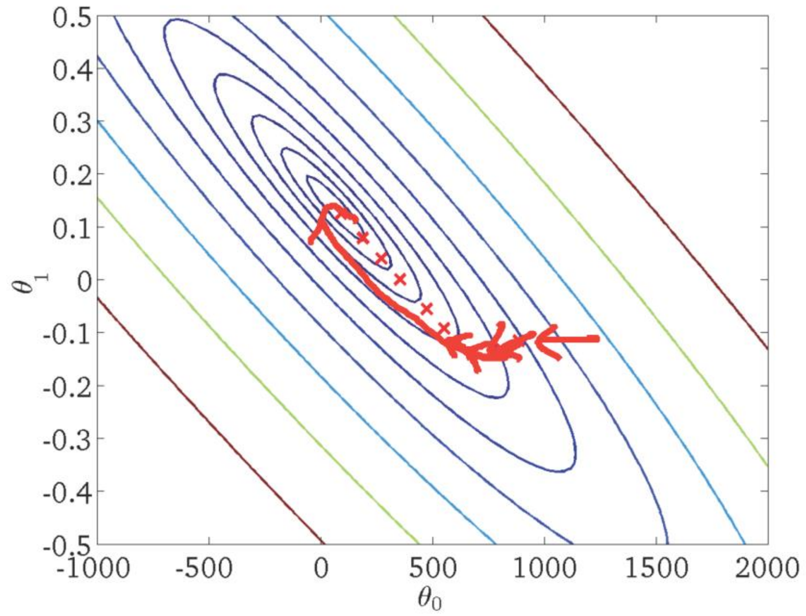
Batch Gradient Descent

Initialize θ

Repeat {

$$\theta_j \leftarrow \theta_j - \alpha \underbrace{\frac{1}{n}\sum_{i=1}^{n}\left(h_{\boldsymbol{\theta}}\left(\mathbf{x}_i\right) - y_i\right) x_{ij}}_{\frac{\partial}{\partial \theta_j} J(\boldsymbol{\theta})} \qquad \text{for } j = 0 \ldots d$$

}



Stochastic Gradient Descent

Initialize θ

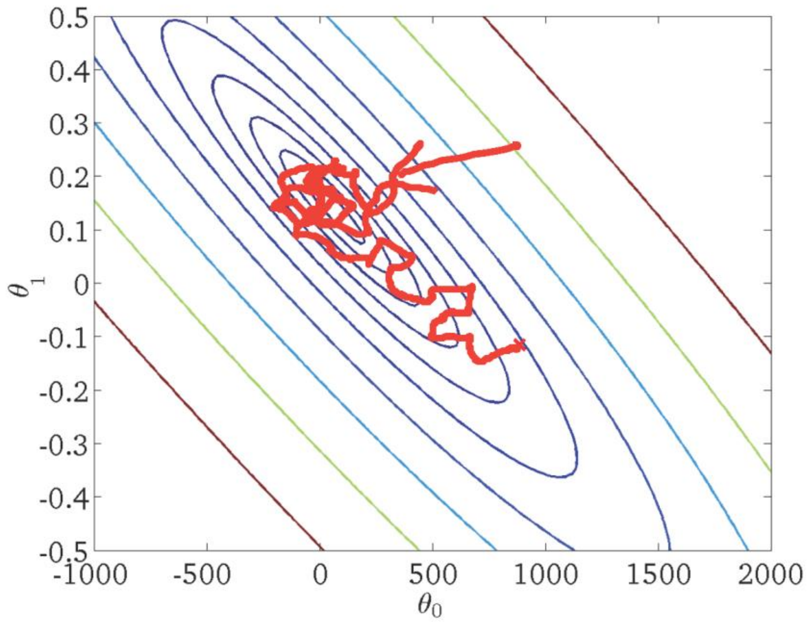
Randomly shuffle dataset

Repeat { (Typically 1 – 10x)

For $i = 1 \ldots n$, *do*

$$\theta_j \leftarrow \theta_j - \alpha \underbrace{\left(h_{\boldsymbol{\theta}}\left(\mathbf{x}_i\right) - y_i\right) x_{ij}}_{\frac{\partial}{\partial \theta_j} \mathrm{cost}_{\boldsymbol{\theta}}(\mathbf{x}_i, y_i)} \qquad \text{for } j = 0 \ldots d$$

}



Adaptive alpha

<u>Stochastic Gradient Descent</u>

Initialize θ

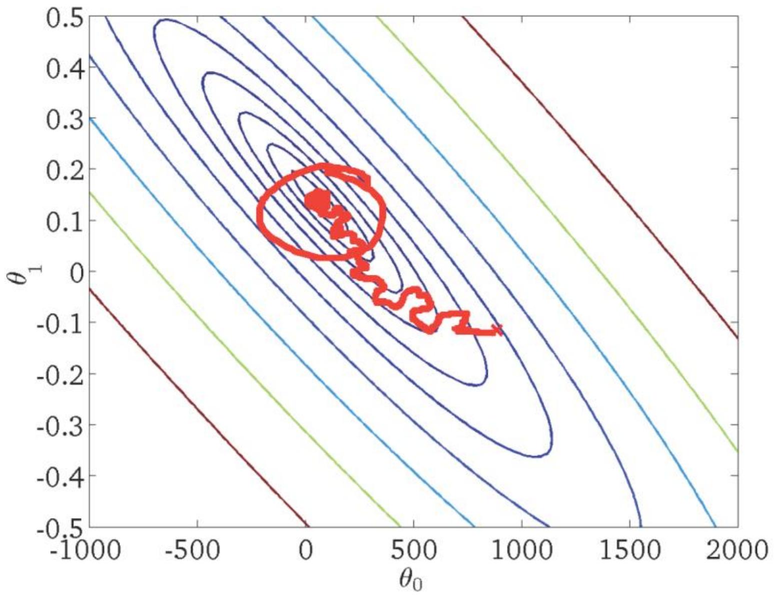
Randomly shuffle dataset

Repeat { (Typically 1 – 10x)

For $i = 1...n$, *do*

$$\theta_j \leftarrow \theta_j - \alpha \underbrace{\left(h_{\boldsymbol{\theta}}\left(\mathbf{x}_i\right) - y_i\right) x_{ij}}_{\frac{\partial}{\partial \theta_j} \mathrm{cost}_{\boldsymbol{\theta}}(\mathbf{x}_i, y_i)} \quad \text{for } j = 0...d$$

}

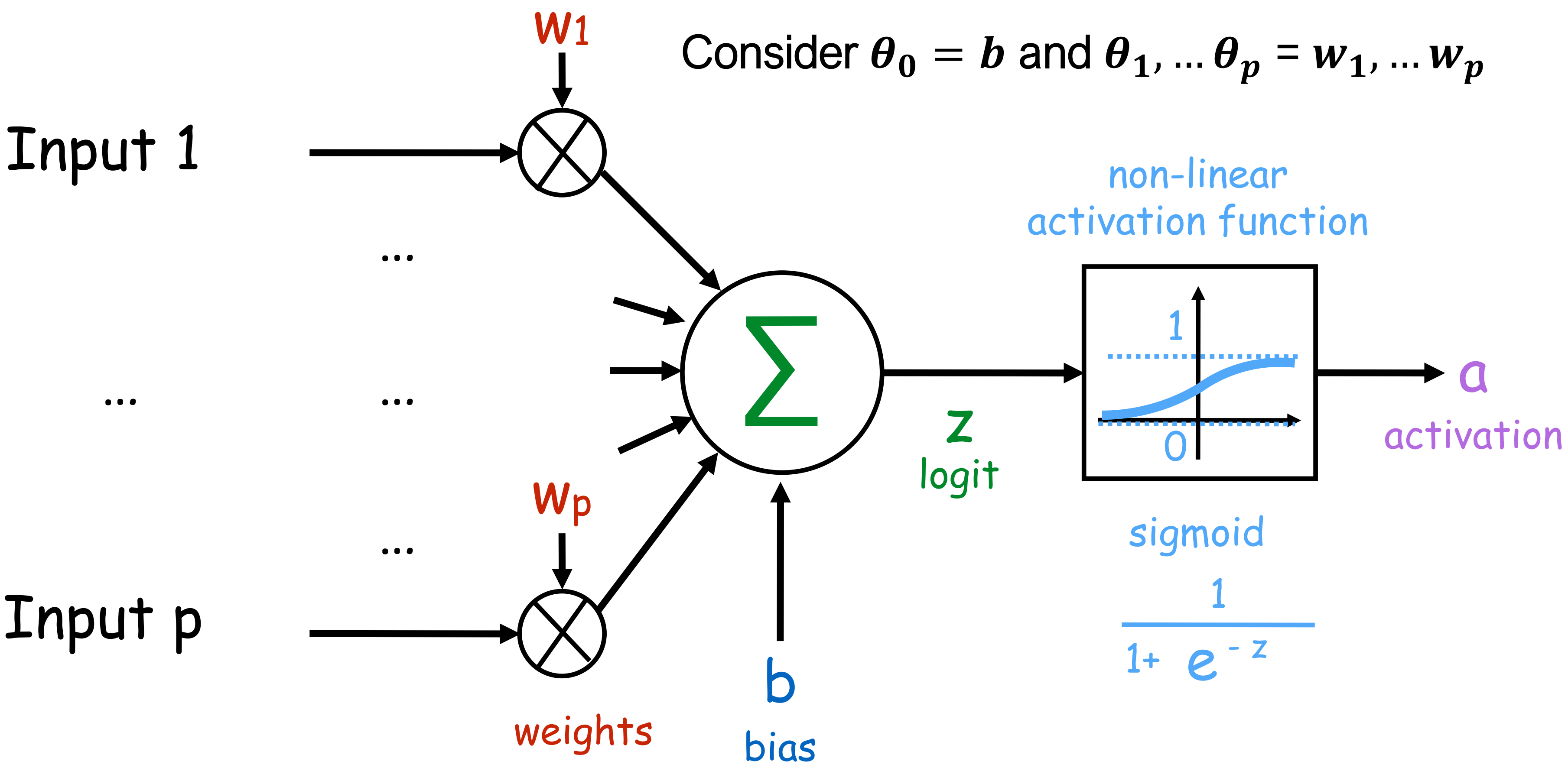


Learning rate α is typically held constant. Can slowly decrease α over time if we want θ to converge. (E.g. $\alpha = \frac{\texttt{const1}}{\texttt{iterationNumber + const2}}$)

Logistic Regression
(for hw3)

Logistic Regression for hw3

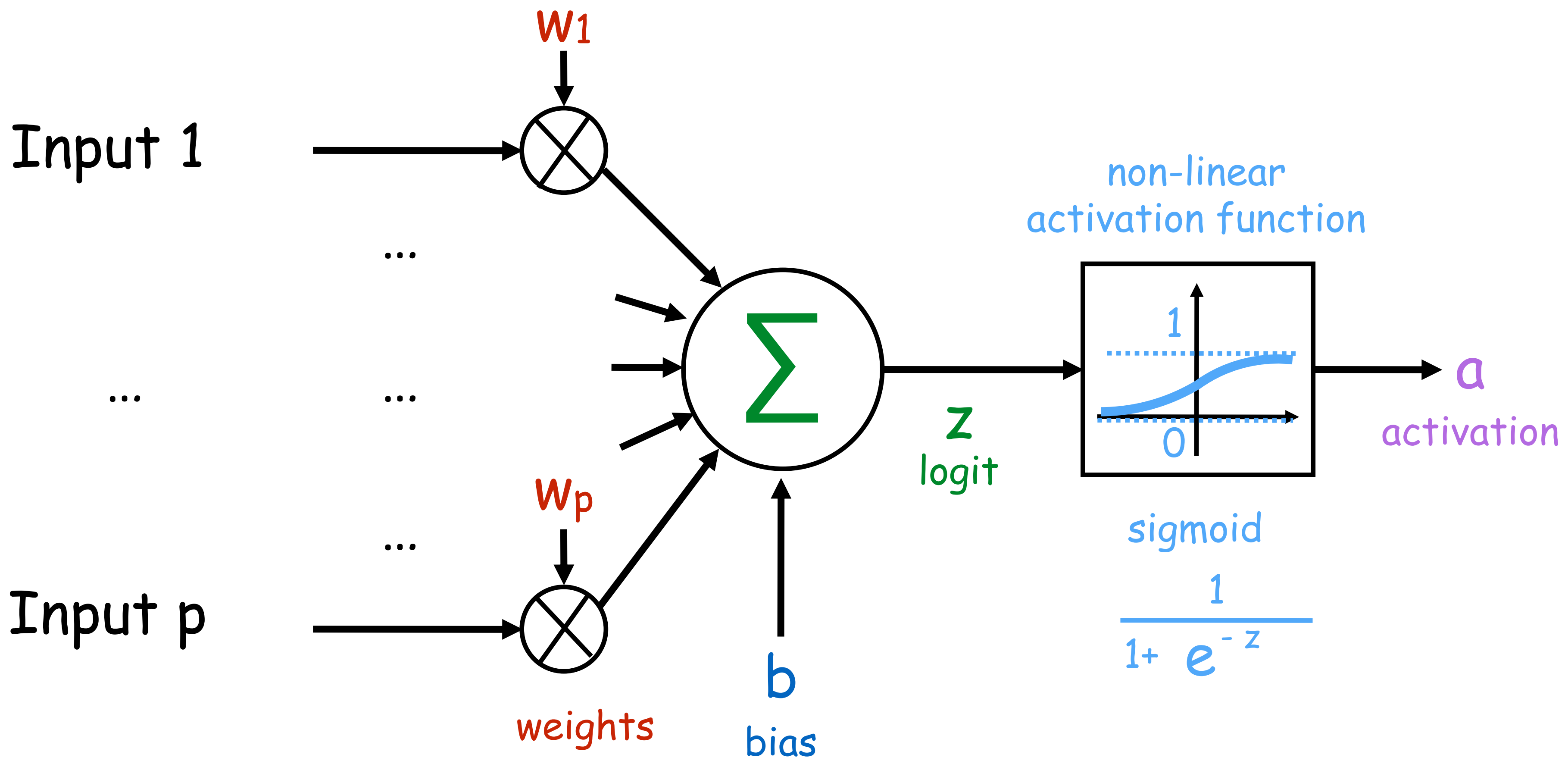
Consider $\boldsymbol{\theta_0} = \boldsymbol{b}$ and $\boldsymbol{\theta_1}, \dots \boldsymbol{\theta_p} = \boldsymbol{w_1}, \dots \boldsymbol{w_p}$



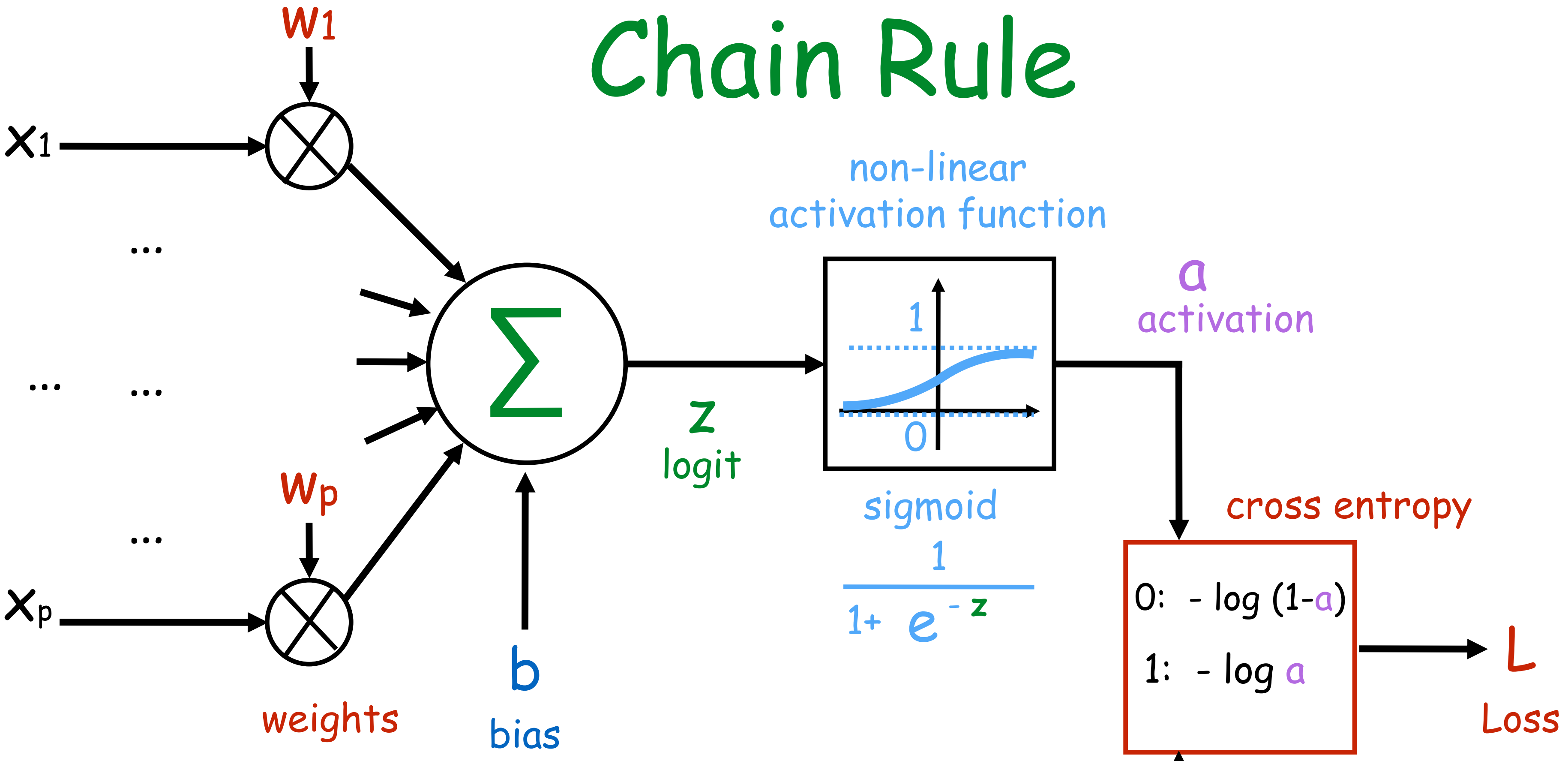
a = Pr (label = + | inputs = x)

output value represents the probability of the instance having positive label

Parameters w, b



Chain Rule



$$\frac{\partial L}{\partial w_i} = \frac{\partial L}{\partial a} \times \frac{\partial a}{\partial z} \times \frac{\partial z}{\partial w_i}$$

$$\frac{\partial L}{\partial b} = \frac{\partial L}{\partial a} \times \frac{\partial a}{\partial z} \times \frac{\partial z}{\partial b}$$

products of local gradients

Chain Rule

$$y = f(g(x)) \qquad y = f(u) \quad u = g(x)$$

$$\frac{\partial y}{\partial x} = \frac{\partial y}{\partial u} \times \frac{\partial u}{\partial x}$$

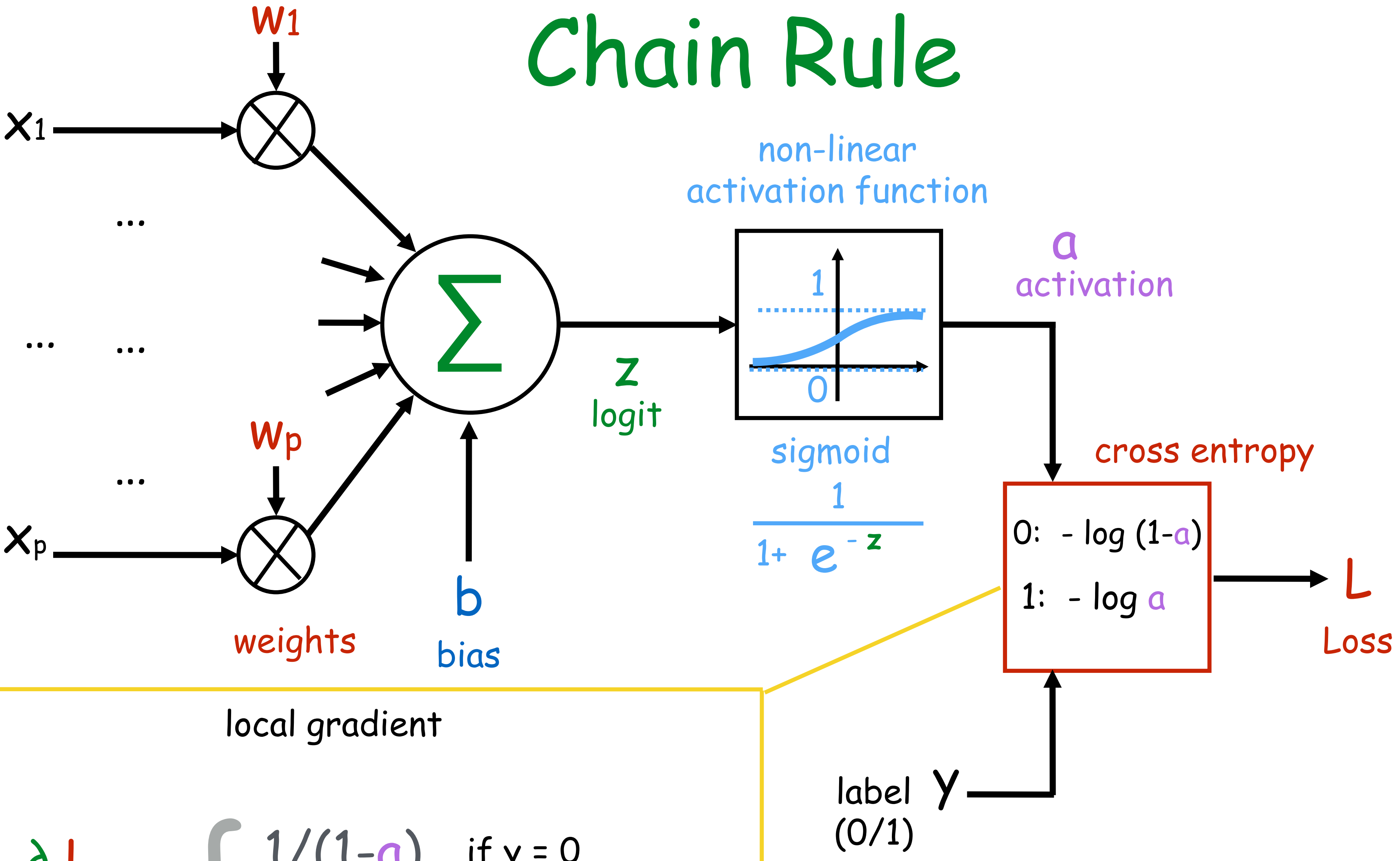
global gradient local gradient local gradient

$$y = f(g(h(x))) \qquad y = f(u) \quad u = g(v) \quad v = h(x)$$

$$\frac{\partial y}{\partial x} = \frac{\partial y}{\partial u} \times \frac{\partial u}{\partial v} \times \frac{\partial v}{\partial x}$$

global gradient local gradient local gradient local gradient

Chain Rule

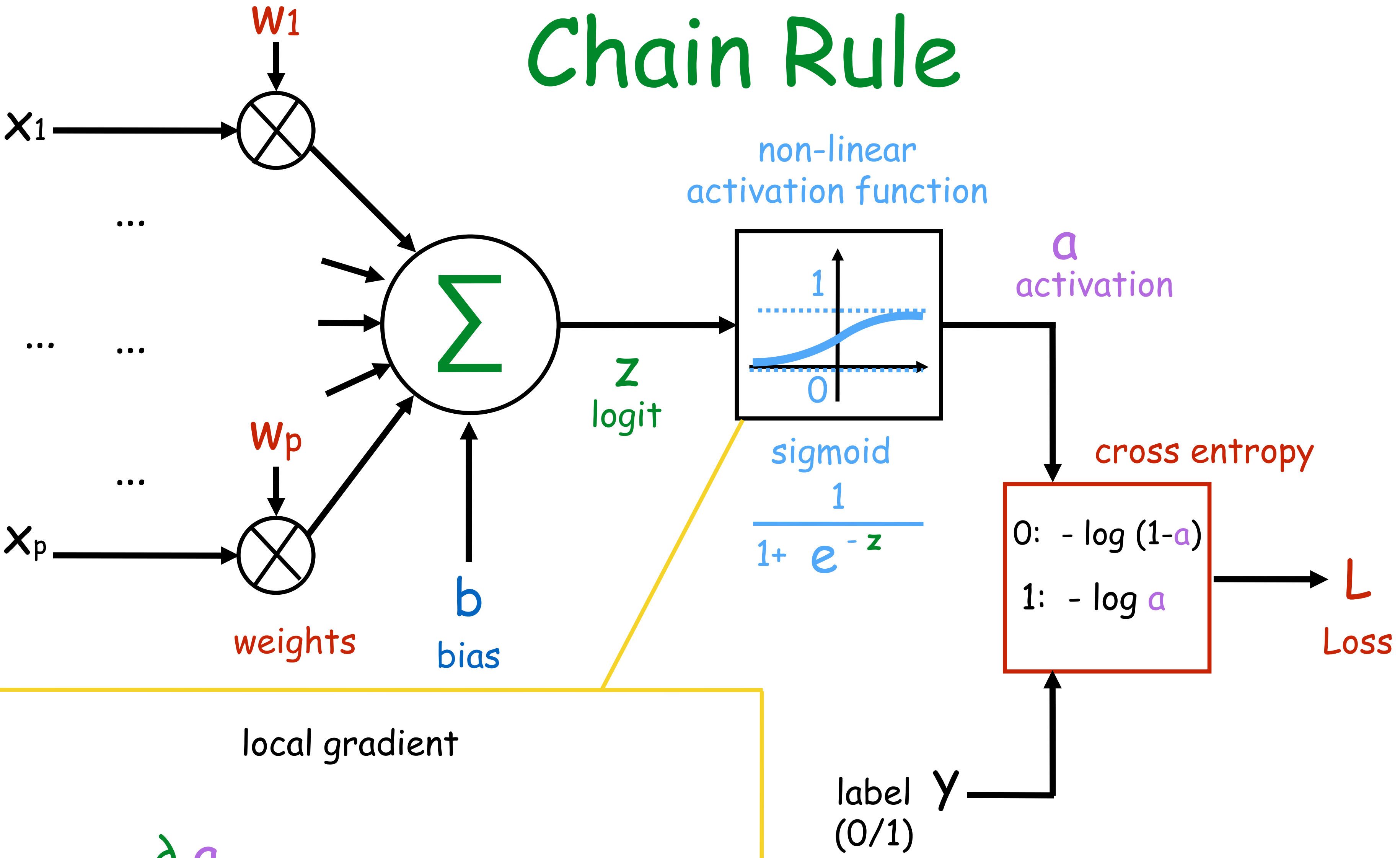


local gradient

$$\frac{\partial L}{\partial a} = \begin{cases} 1/(1-a) & \text{if } y = 0 \\ -1/a & \text{if } y = 1 \end{cases}$$

https://www.derivative-calculator.net/

Chain Rule

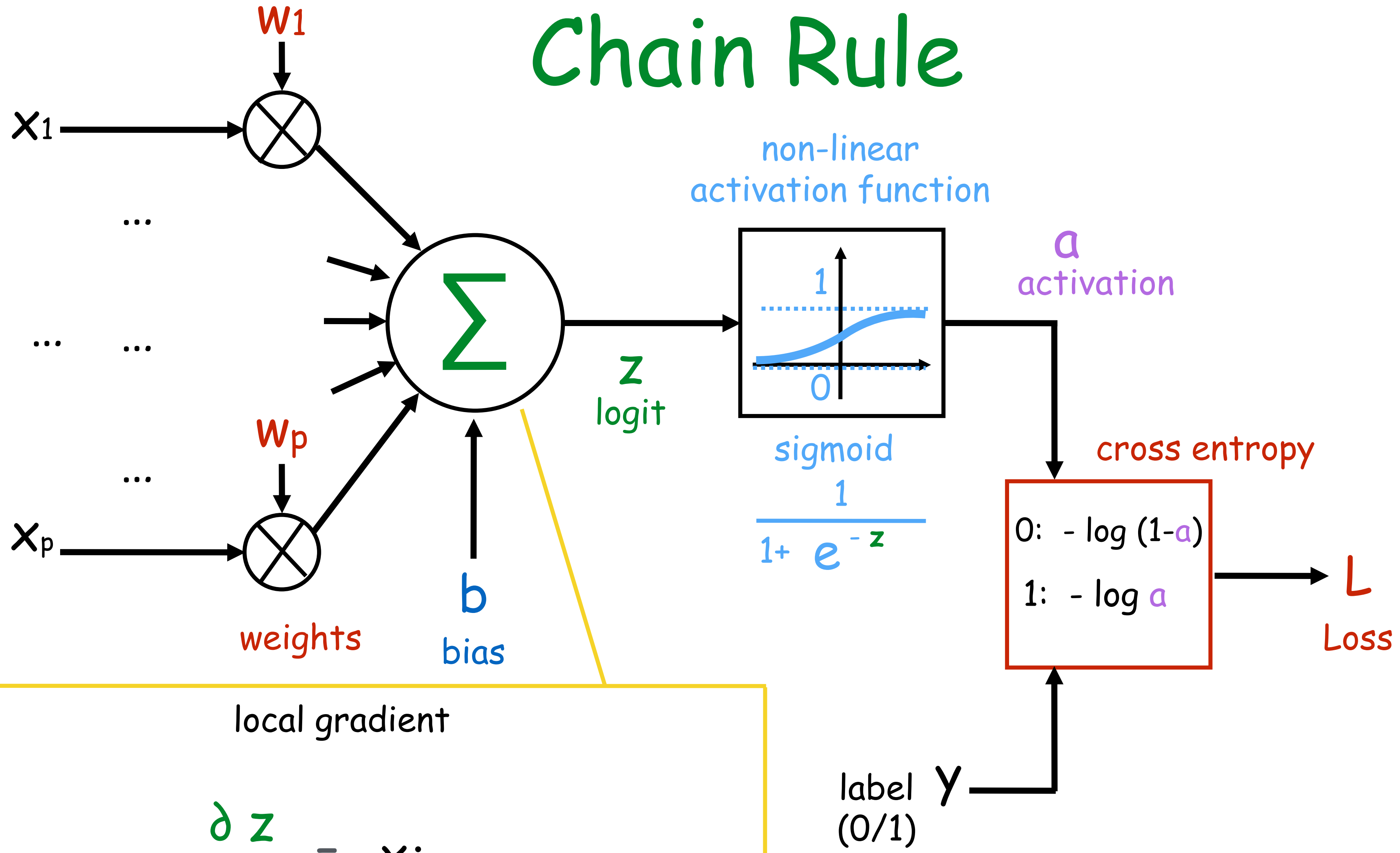


local gradient

$$\frac{\partial a}{\partial z} = a\,(1 - a)$$

https://stats.stackexchange.com/questions/278771/how-is-the-cost-function-from-logistic-regression-derivated

Chain Rule

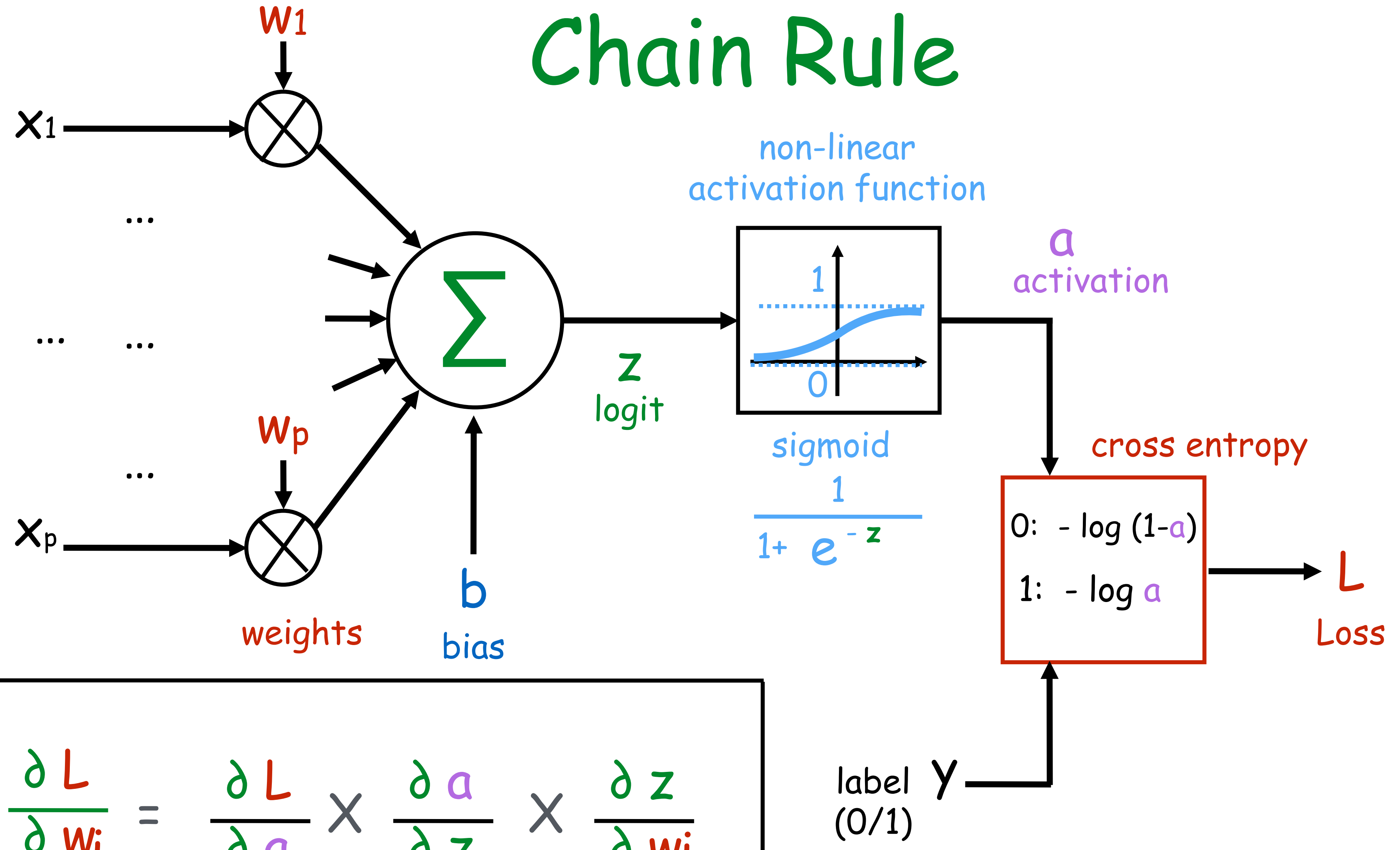


local gradient

$$\frac{\partial z}{\partial w_i} = x_i$$

$$\frac{\partial z}{\partial b} = 1$$

Chain Rule



$$\frac{\partial L}{\partial w_i} = \frac{\partial L}{\partial a} \times \frac{\partial a}{\partial z} \times \frac{\partial z}{\partial w_i}$$

$$\frac{\partial L}{\partial b} = \frac{\partial L}{\partial a} \times \frac{\partial a}{\partial z} \times \frac{\partial z}{\partial b}$$

products of local gradients

Logistic Regression (train)

initialize $\mathbf{w}$ and b

Loop for n_epoch iterations:

Loop for each training instance ($\mathbf{x}$, y) in training set

forward pass to compute z, a and L for the instance

backward pass to compute local gradients

$$\frac{\partial L}{\partial a} \quad \frac{\partial a}{\partial z} \quad \frac{\partial z}{\partial b} \quad \frac{\partial z}{\partial \mathbf{w}}$$

compute global gradients using chain rule

$$\frac{\partial L}{\partial \mathbf{w}} \quad \frac{\partial L}{\partial b}$$

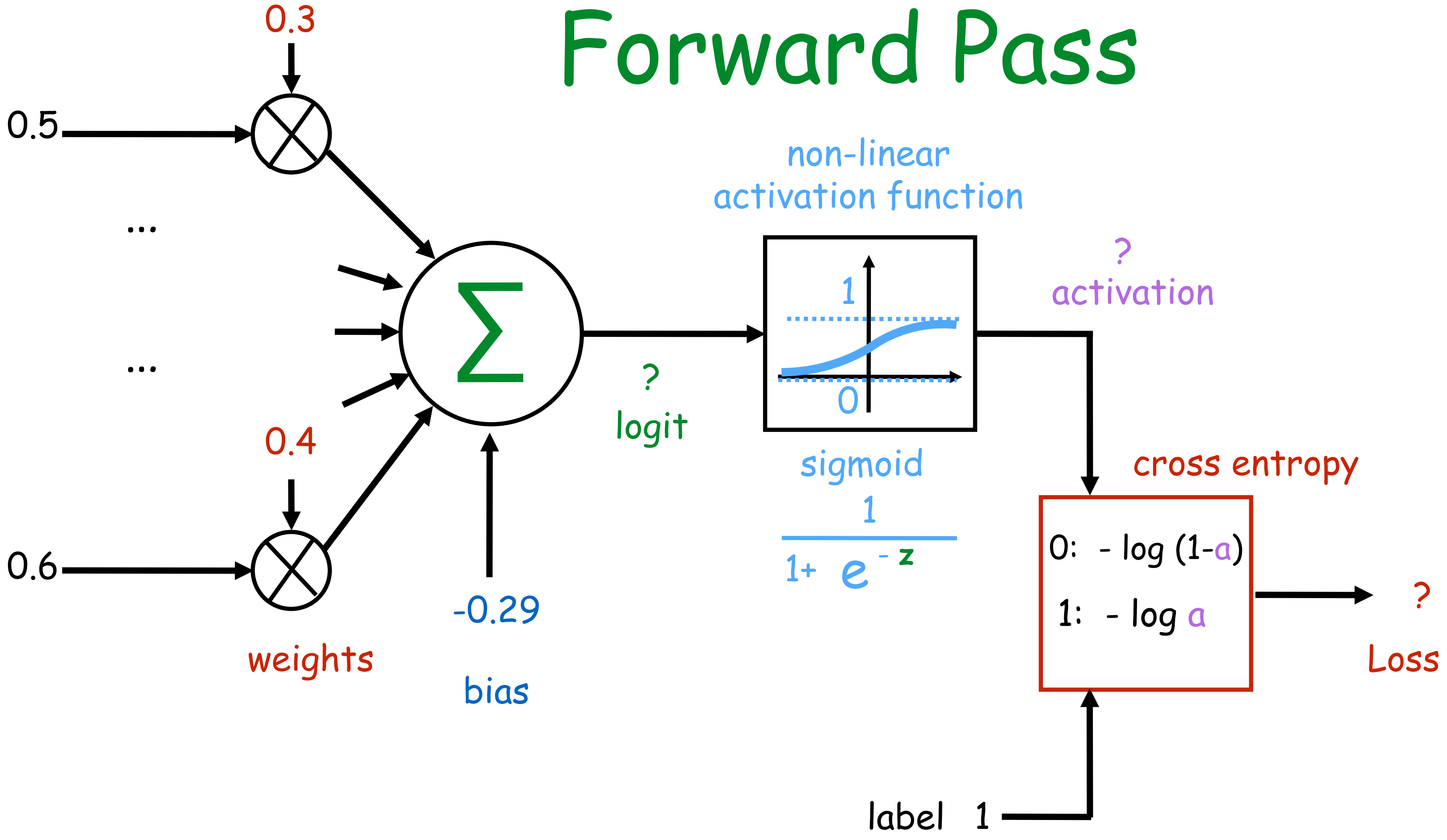
update the parameters $\mathbf{w}$ and b

$$\mathbf{w} \leftarrow \mathbf{w} - a\frac{\partial L}{\partial \mathbf{w}}$$

$$b \leftarrow b - a\frac{\partial L}{\partial b}$$

Example
(for your reference)

Forward Pass

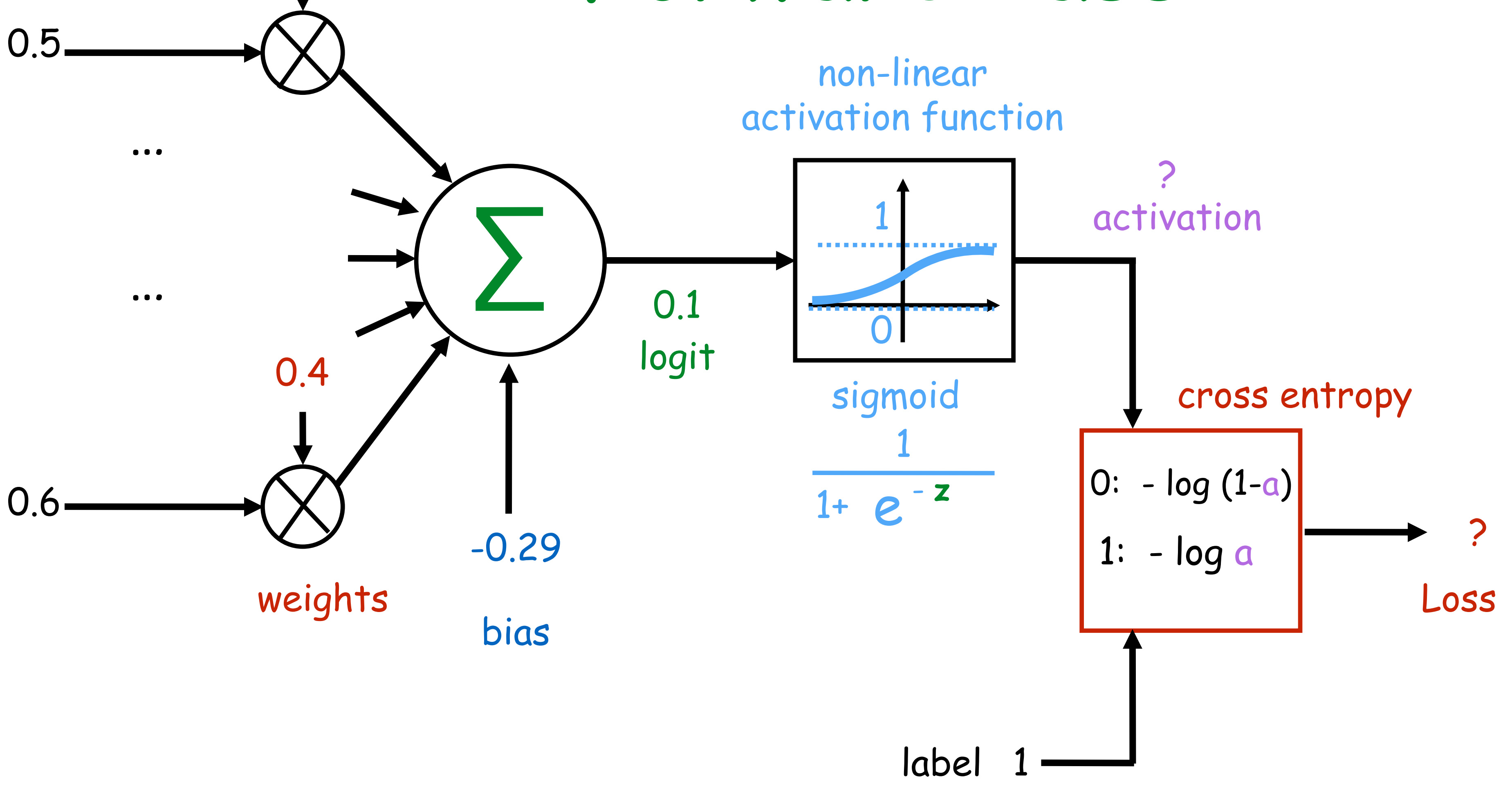


Given a training instance **x**, y

x = (0.5, ..., 0.6)

y = 1

Forward Pass

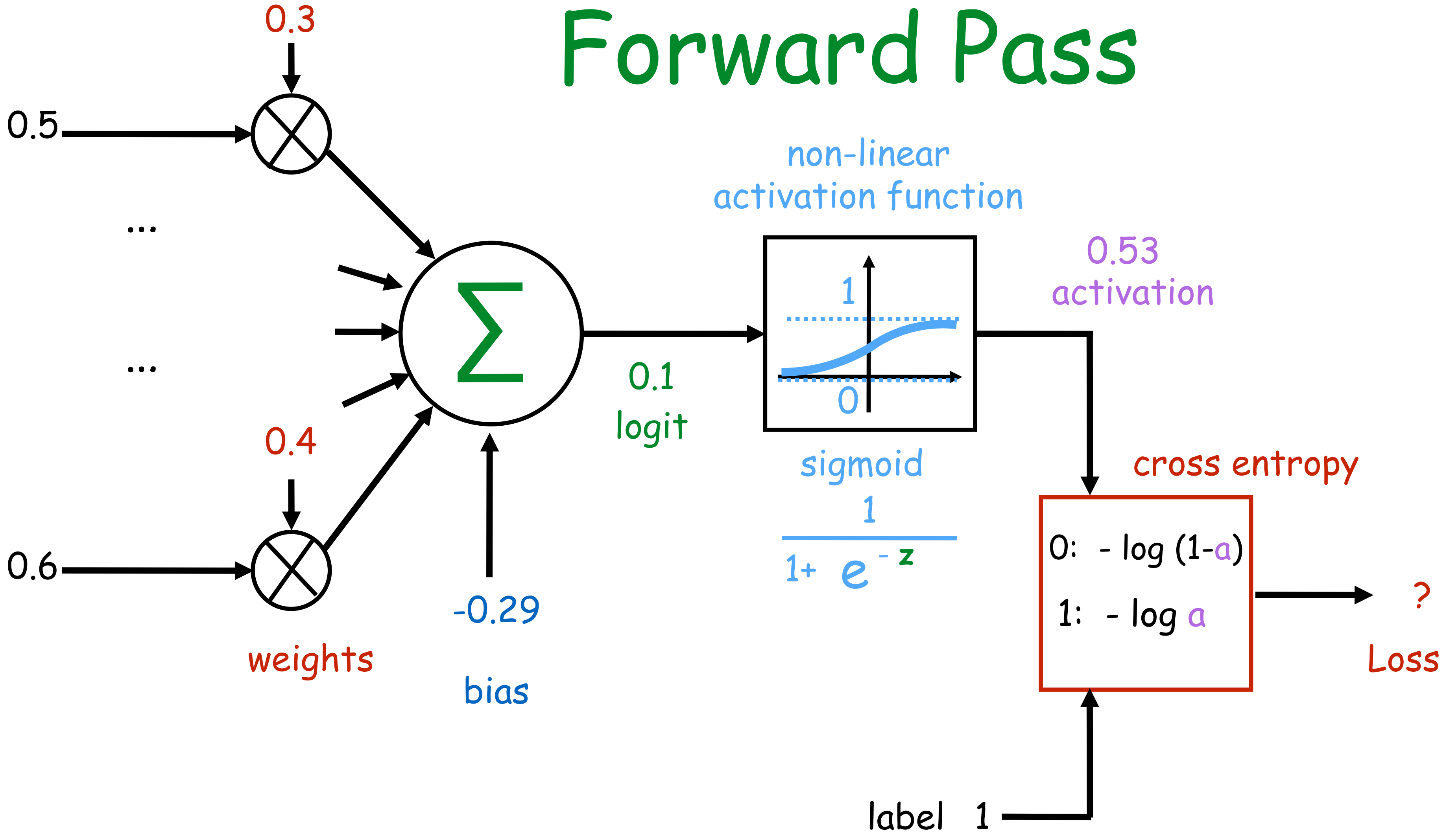


Given a training instance **x**, y

x = (0.5, ..., 0.6)

y = 1

Forward Pass

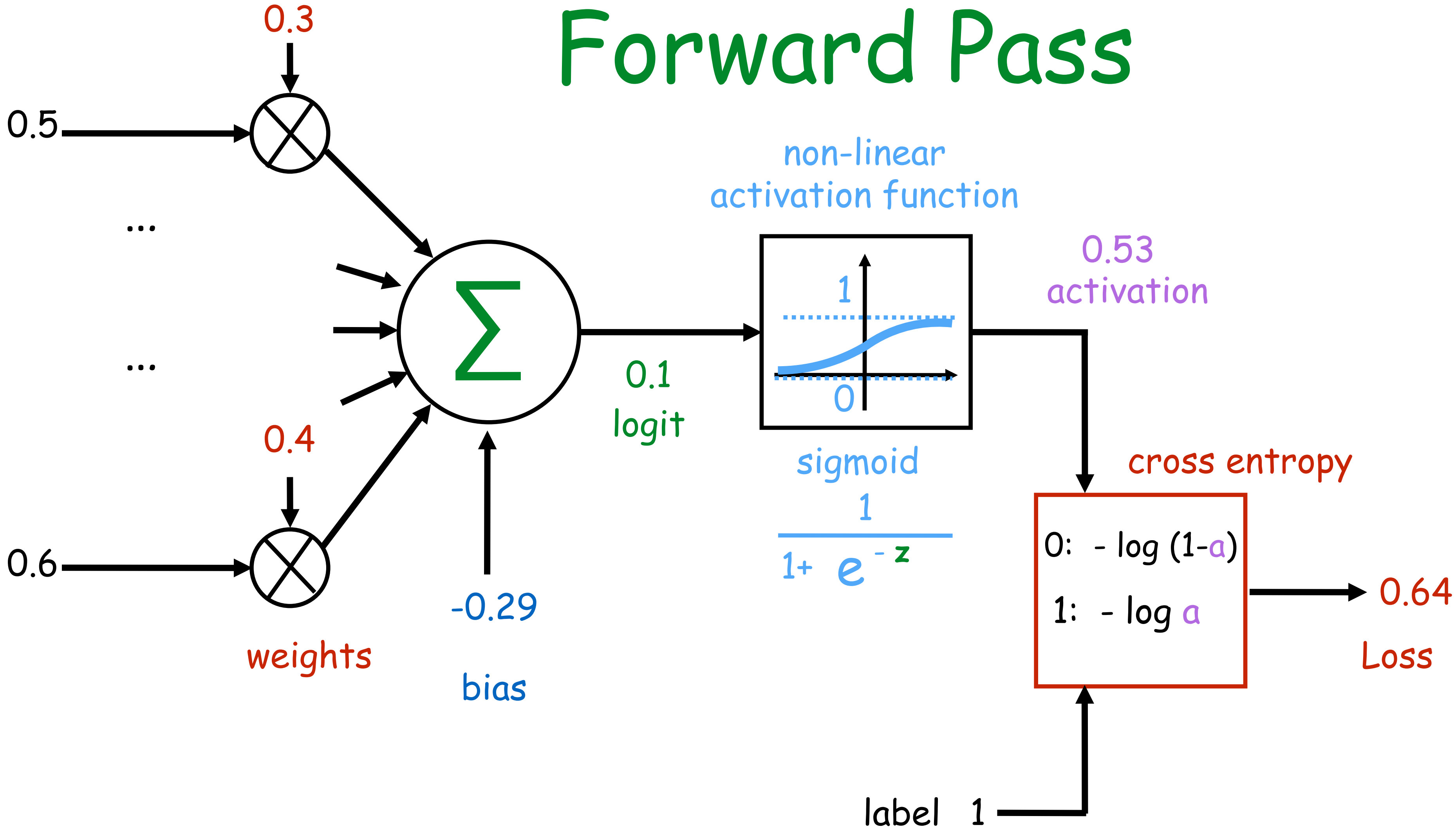


Given a training instance **x**, y

x = (0.5, …, 0.6)

y = 1

Forward Pass

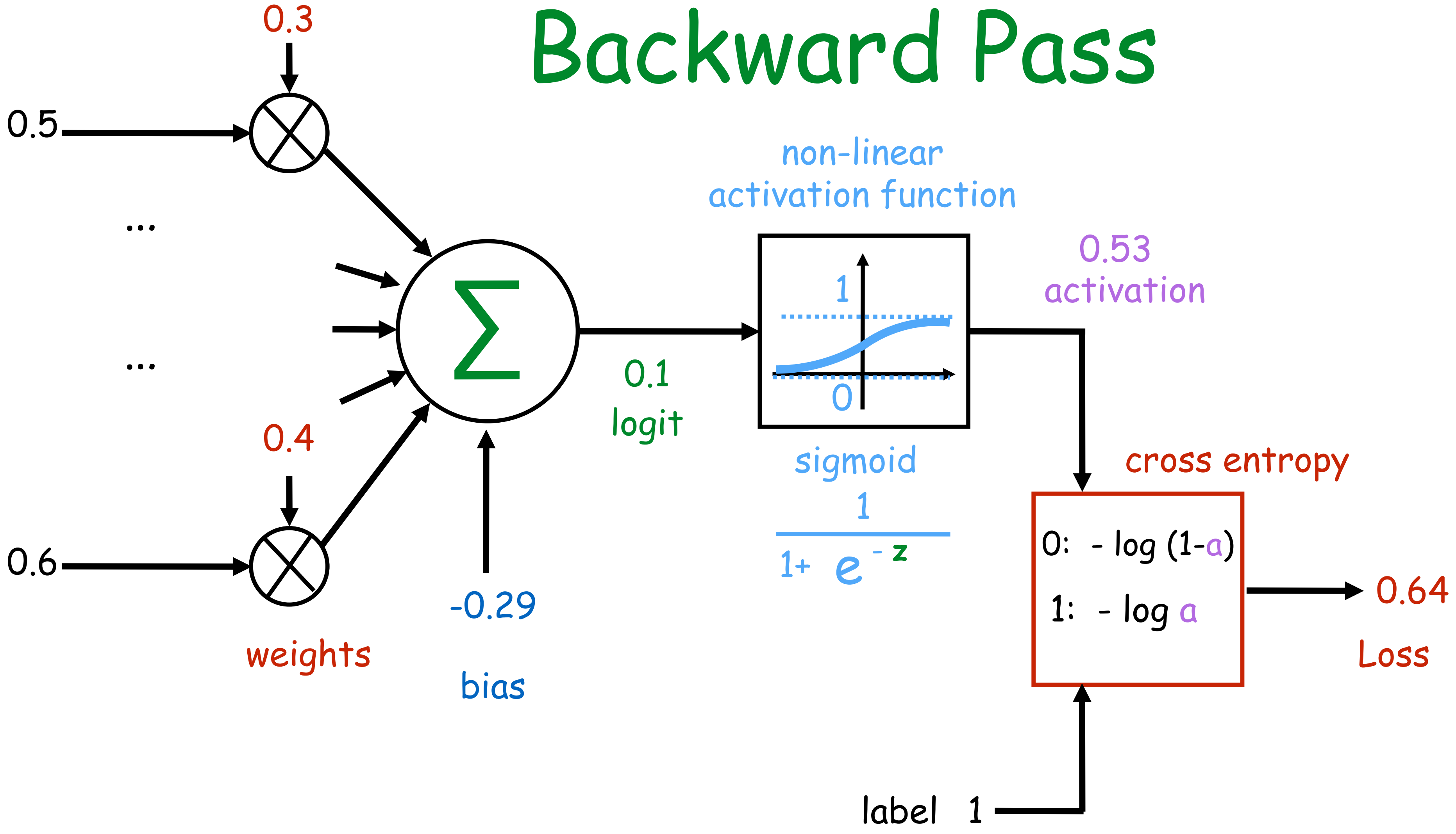


Given a training instance **x**, y

x = (0.5, …, 0.6)

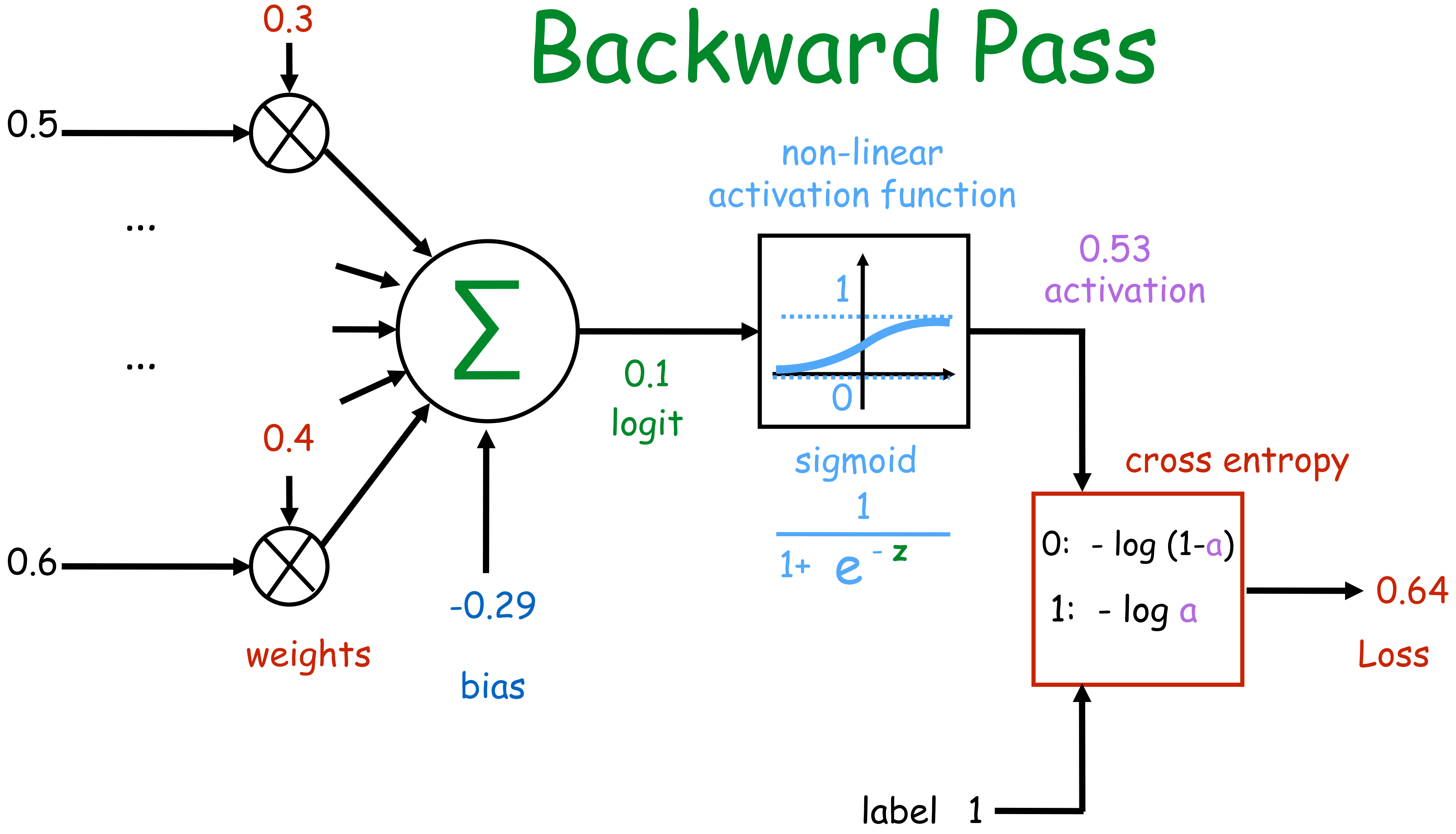
y = 1

Backward Pass



$$\frac{\partial L}{\partial a} = -1/a$$
$$= -1.9$$

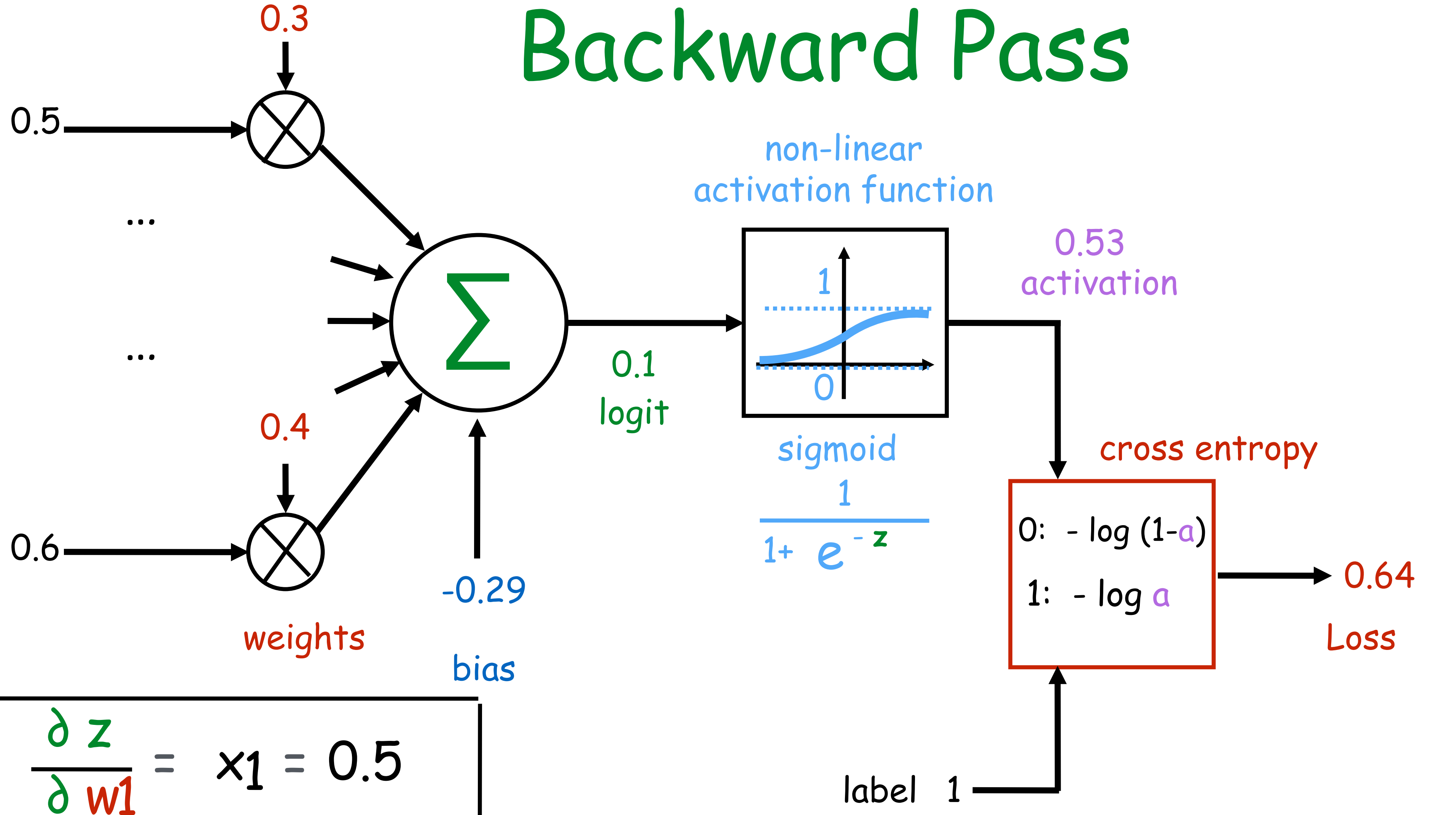
Backward Pass



$$\frac{\partial a}{\partial z} = a(1 - a) = 0.24$$

$$\frac{\partial L}{\partial a} = -1.9$$

Backward Pass



$$\frac{\partial z}{\partial w_1} = x_1 = 0.5$$

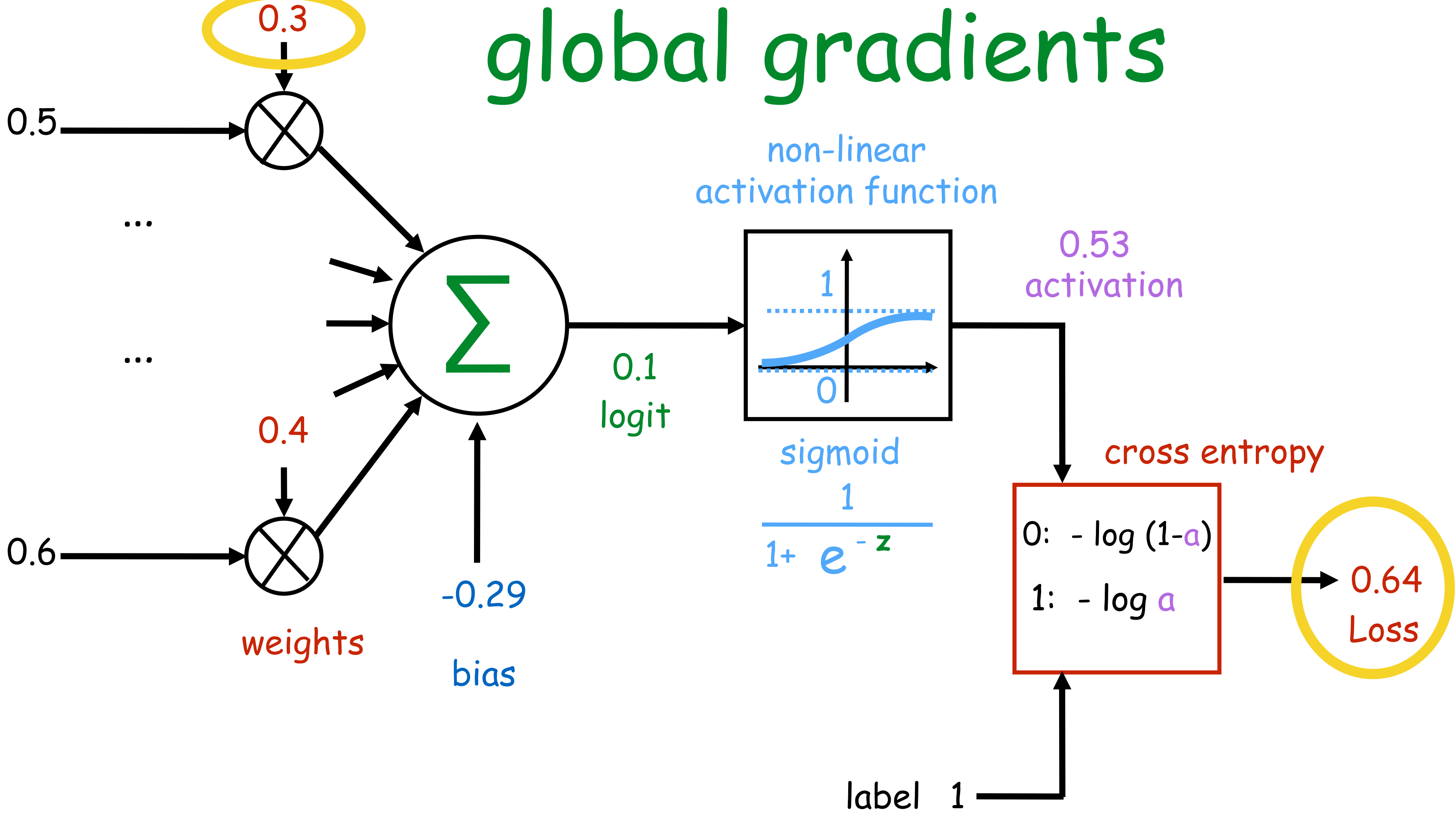
$$\frac{\partial z}{\partial w_p} = x_p = 0.6$$

$$\frac{\partial z}{\partial b} = 1$$

$$\frac{\partial a}{\partial z} = 0.24$$

$$\frac{\partial L}{\partial a} = -1.9$$

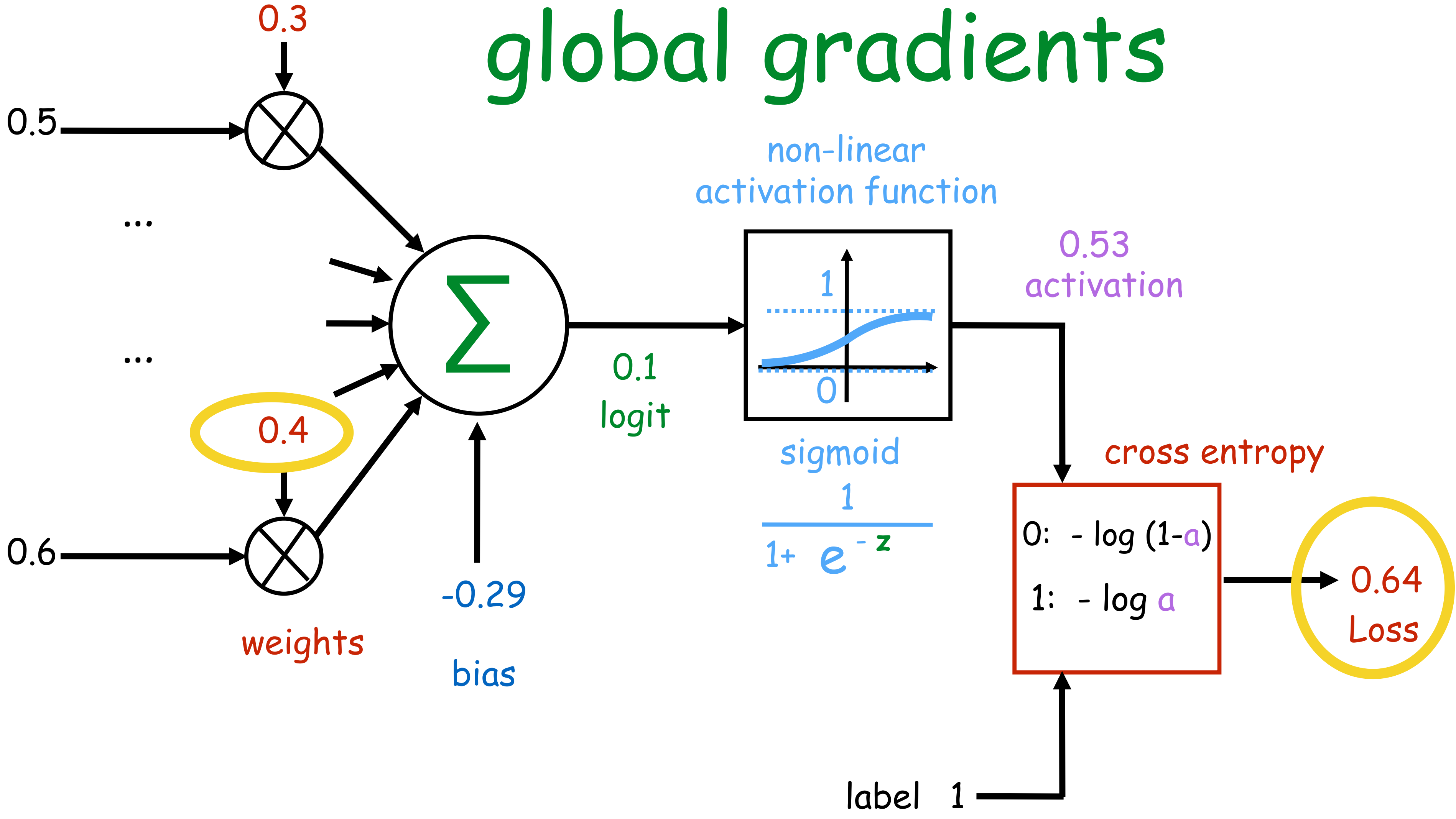
global gradients



$$\frac{\partial L}{\partial w_1} = \frac{\partial z}{\partial w1} \times \frac{\partial a}{\partial z} \times \frac{\partial L}{\partial a}$$

-0.22   0.5   0.24   -1.9

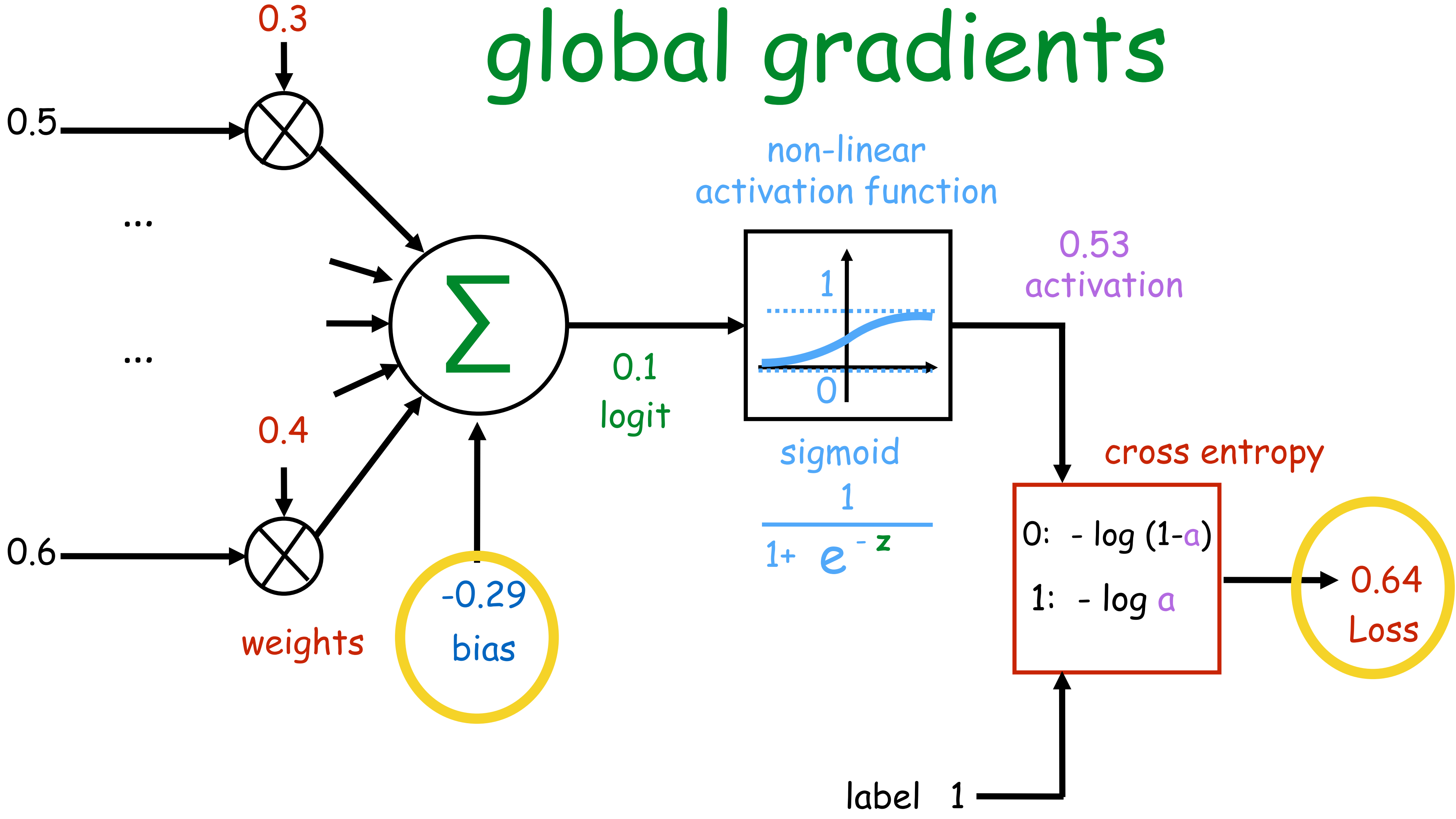
global gradients



$$\frac{\partial L}{\partial w_p} = \frac{\partial z}{\partial w_p} \times \frac{\partial a}{\partial z} \times \frac{\partial L}{\partial a}$$

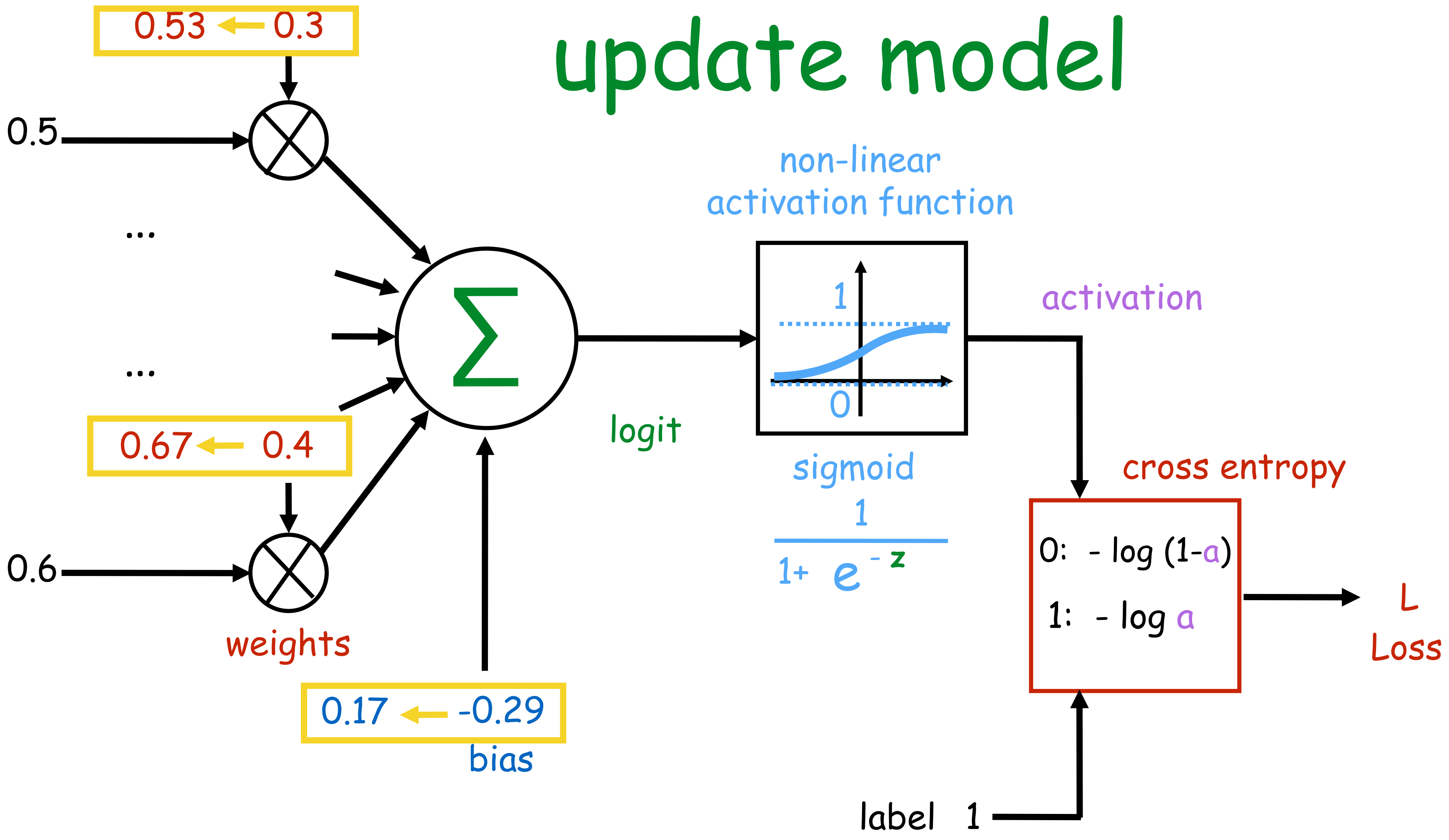
-0.27   0.6   0.24   -1.9

global gradients



$$\underset{-0.46}{\frac{\partial L}{\partial b}} = \underset{1}{\frac{\partial z}{\partial b}} \times \underset{0.24}{\frac{\partial a}{\partial z}} \times \underset{-1.9}{\frac{\partial L}{\partial a}}$$

update model



$$\mathbf{w} \leftarrow \mathbf{w} - a\frac{\partial L}{\partial \mathbf{w}}$$

$$b \leftarrow b - a\frac{\partial L}{\partial b}$$

for example

$a = 1$