Natural Language Processing

第二周 机器学习简介

庞彦

yanpang@gzhu.edu.cn

Spring 2023

Overview

CONTENTS

01 机器学习简介

02 线性逻辑回归

03 反向传播网络

01 Machine Learning

机器学习简介

Machine Learning

推理时期	1960s	赋予机器逻辑推理能力使机器获得智能；当时的AI程序证明力一些著名的数学定理，但由于缺乏知识，远不能实现真正的智能。
知识时期	1970s	将人类的知识总结出来教给机器使机器获得智能；即“专家系统”，在很多领域获得大量进展，但由于人类知识量巨大，故出现“知识工程瓶颈”。
机器学习时期	1980s	连接主义较为流行；代表方法为神经网络。
	1990s	统计学习占据舞台。代表方法包括支持向量机等。
	21Cent.	深度神经网络被提出，连接主义卷土重来。随着数据量和计算能力的不断提升，以深度学习为基础的诸多AI应用逐渐成熟。

Machine Learning

What is Machine Learning?

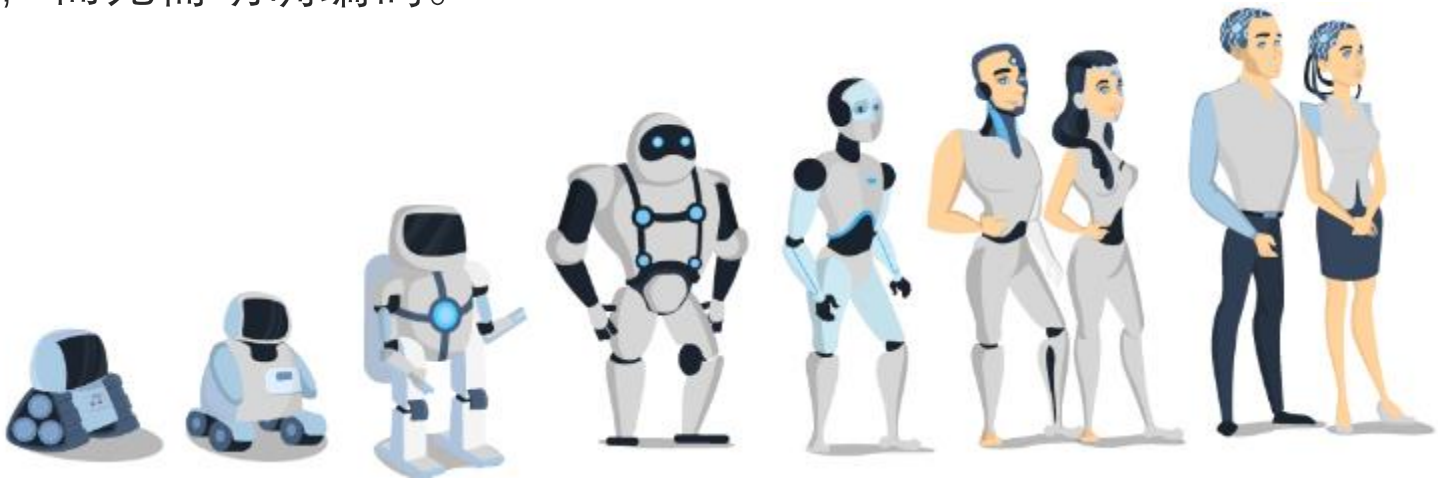
Machine Learning is the science (and art) of programming computers so they can **learn** from data.

机器学习可以让机器从数据中学习知识。

General Definition:

Machine Learning is the field of study that gives computers the ability to **learn** without being explicitly programmed. [Arthur Samuel, 1959]

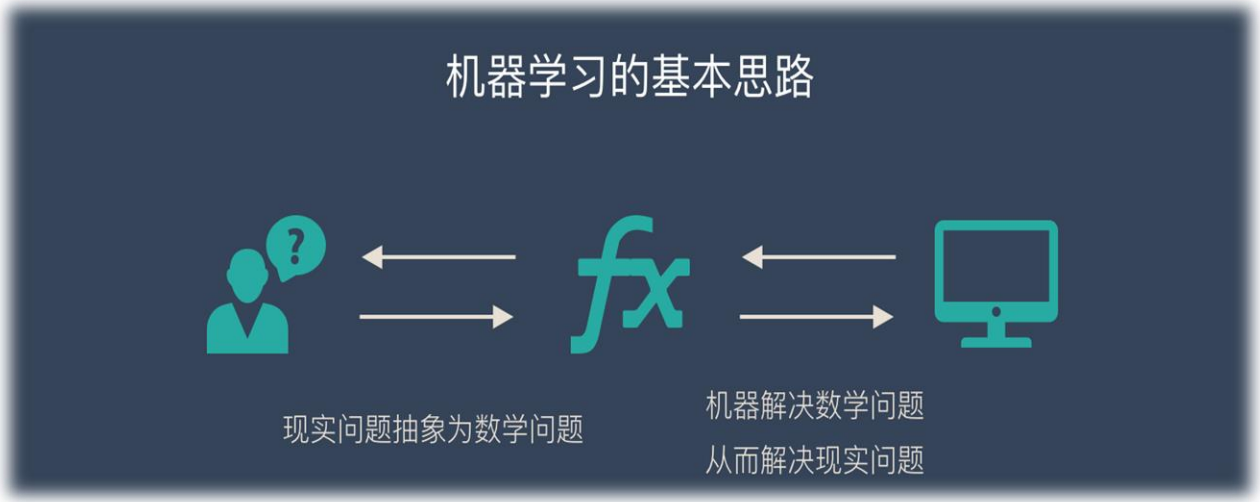
机器学习可以让机器具备自我学习能力，而无需明确编码。



Machine Learning

Steps of Machine Learning：

1) 把现实生活中的问题抽象成数学模型，并且很清楚模型中不同参数的作用
2) 利用数学方法对这个数学模型进行求解，从而解决现实生活中的问题
3) 评估这个数学模型，是否真正的解决了现实生活中的问题，解决的如何?



Machine Learning

Why Machine Learning?

Consider how you would write a spam filter using traditional programming techniques.

如何用传统的编程技术来写一个垃圾邮件分类器呢?

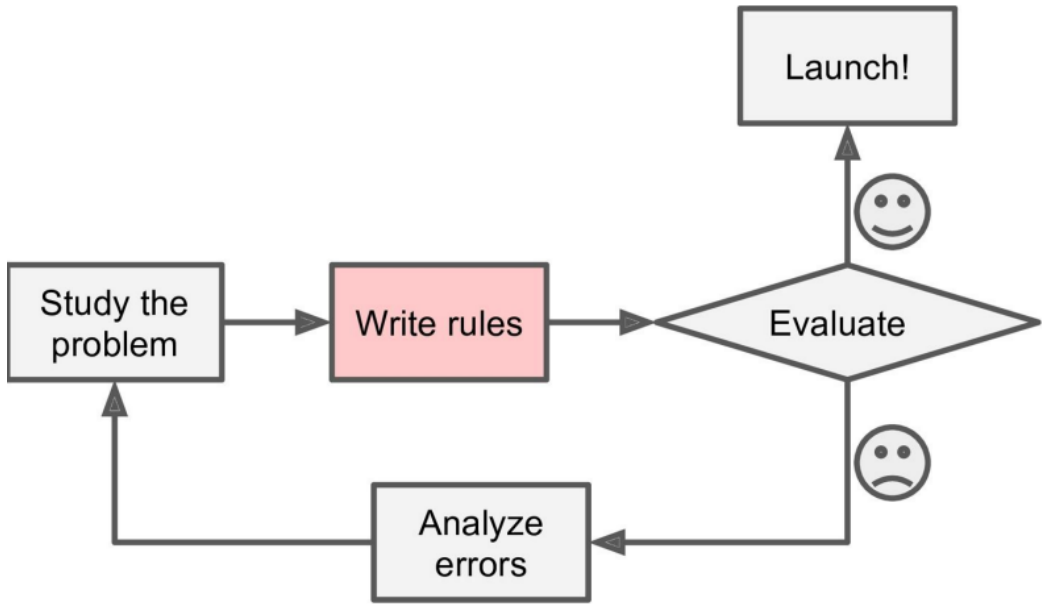
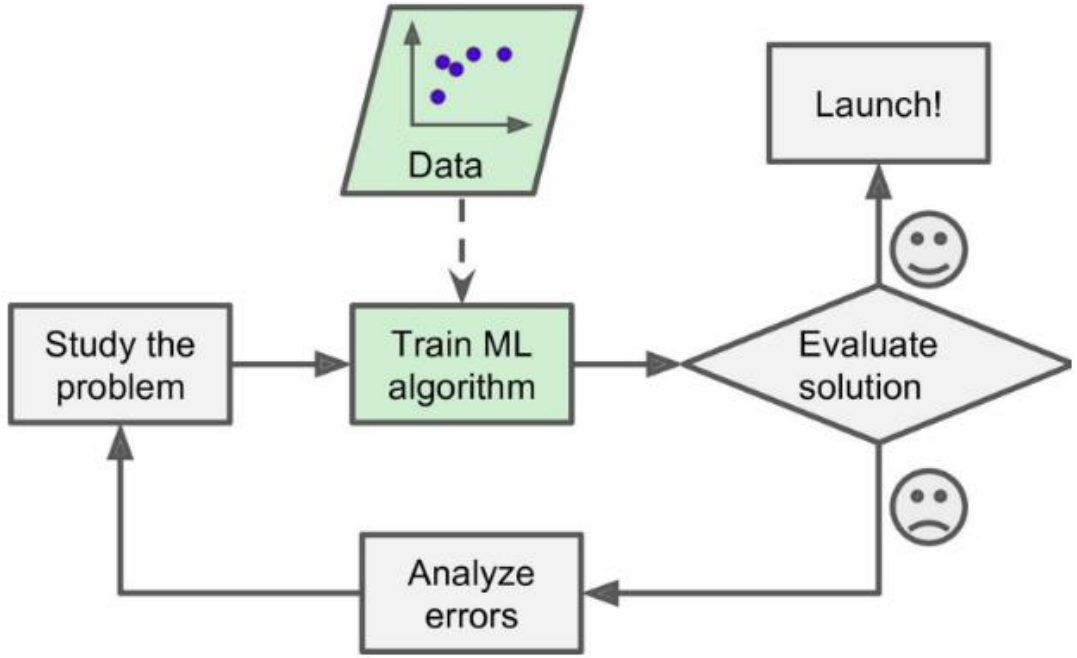


Figure 1-1. The traditional approach



Machine Learning

Machine Learning is Great For:

- Problems for which existing solutions require a lot of hand-tuning.
  代替大量简单的重复性人工操作
- For complex problems which can not solved by using traditional approach.
  传统难以解决的复杂问题
- For fluctuating environments where machine learning can adopt a new data easily.
  复杂环境，但机器学习可以很容易的获取数据

Review

函数集合

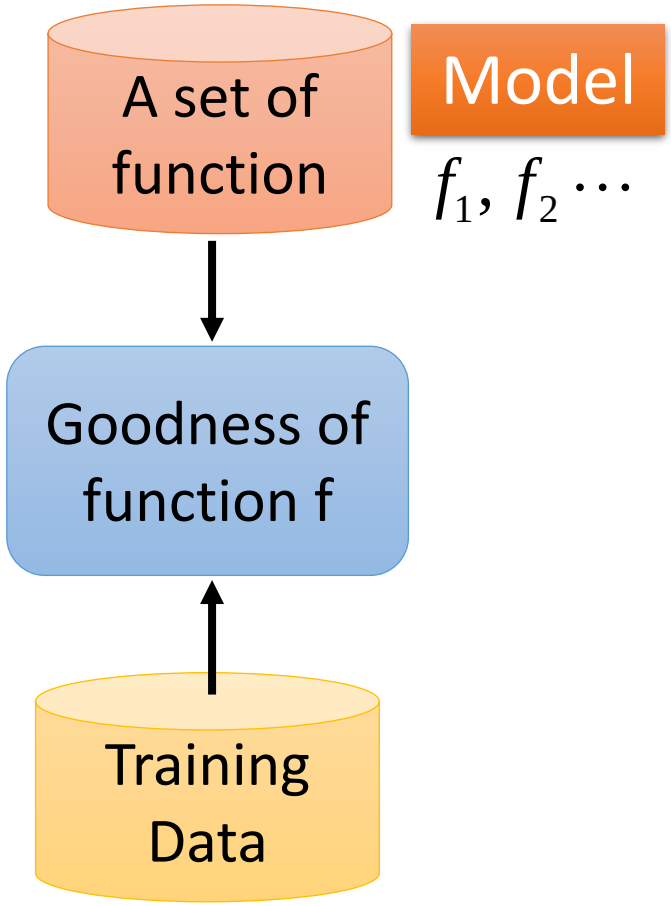
目标识别

$f($  $)=$ “cat”

$f_1($  $)=$ “cat”　　$f_2($  $)=$ “flamingo”

$f_1($  $)=$ “dog”　　$f_2($  $)=$ “lynx”

Review



$f($  $)=$ "cat"

$f_1($  $)=$ "cat"

$f_2($  $)=$ "flamingo"

Better!

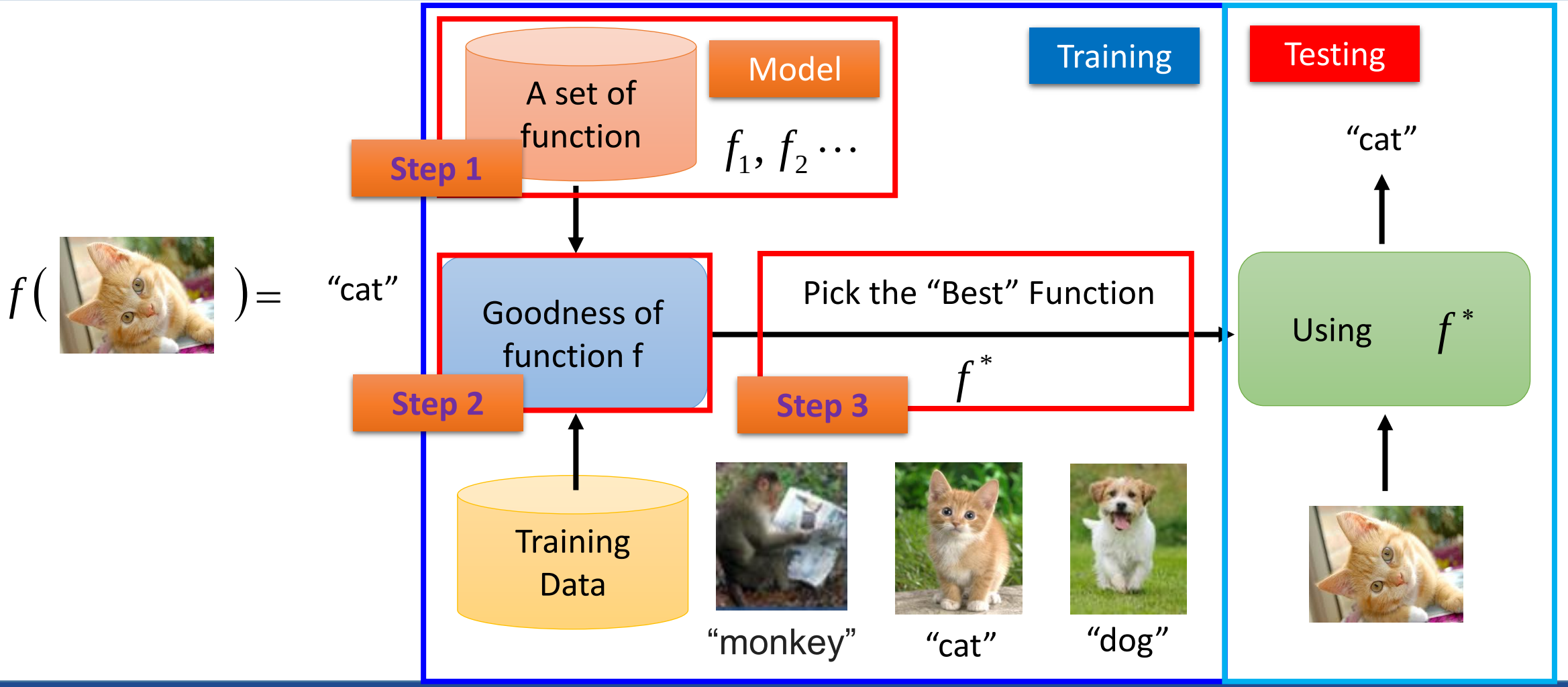
$f_1($  $)=$ "dog"

$f_2($  $)=$ "lynx"

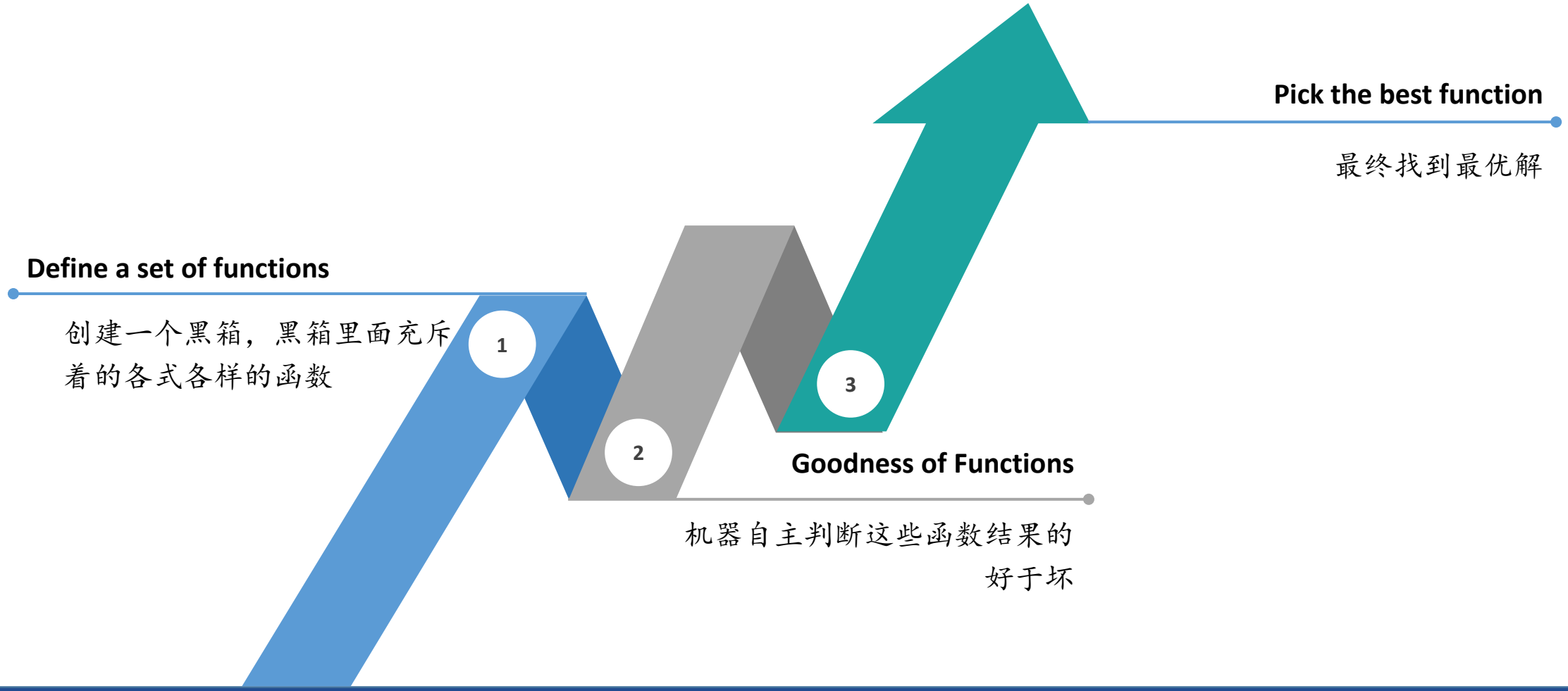
function input:   

function output: "monkey" "cat" "dog"

Review: Three Steps



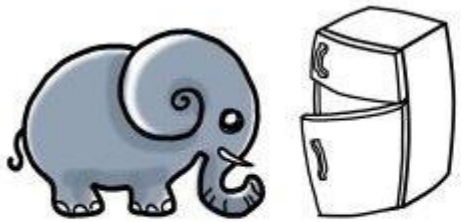
Review: Three Steps



Review: Three Steps

Define a set of functions

创建一个黑箱，黑箱里面充斥着的各式各样的函数

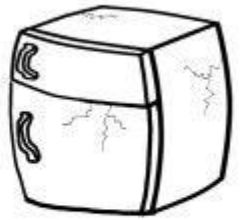


Goodness of Functions

机器自主判断这些函数结果的好于坏

Pick the best function

最终找到最优解



02 Linear and Logistic Regression

逻辑回归原理

Regression: Output a scalar

- 股票交易预测

$f($  $)=$ 明天的道琼斯工业指数均值

- 无人驾驶

$f($  $)=$ 方向盘调整角度

- 推荐系统

$f($  $)=$ 当前用户对某产品潜在购买最大概率

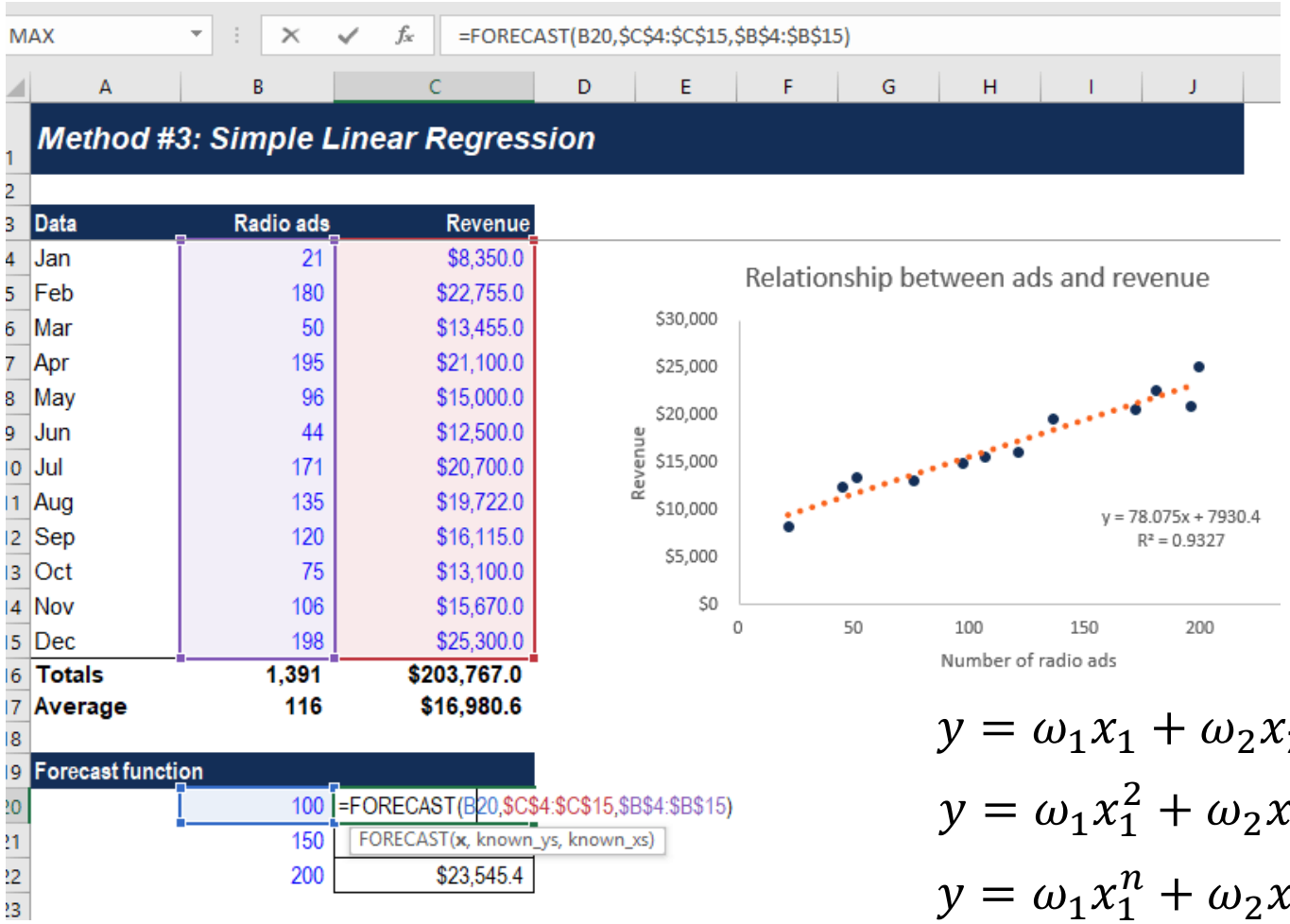
Regression

回归的底层逻辑

通过大量历史数据总结出「套路」

通过总结的「套路」预测数据

Regression Application



$$y = \omega x + b$$

$$y = \omega_1 x_1 + \omega_2 x_2 + \cdots + \omega_n x_n + b$$

$$y = \omega_1 x_1^2 + \omega_2 x_2 + b$$

$$y = \omega_1 x_1^n + \omega_2 x_2^{n-1} + \cdots \omega_n x_n^1 + b$$

Step 1: model

$y = \omega x + b$ w and b 参数(可以是任意值)

A set of function

Model

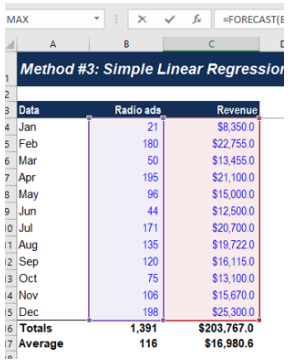
$f_1, f_2 \cdots$

f_1: y = 10.0 + 9.0 · x

f_2: y = 9.8 + 9.2 · x

f_3: y = - 0.8 - 1.2 · x

…… infinite无限

$f($  $x) =$ Revenue 收入 y

Method #3: Simple Linear Regression

Data	Radio ads	Revenue
Jan	21	$8,350.0
Feb	180	$22,755.0
Mar	50	$13,455.0
Apr	195	$21,100.0
May	96	$15,000.0
Jun	44	$12,500.0
Jul	171	$20,700.0
Aug	135	$19,722.0
Sep	120	$16,115.0
Oct	75	$13,100.0
Nov	106	$15,670.0
Dec	198	$25,300.0
Totals	1,391	$203,767.0
Average	116	$16,980.6

x_i: $x_{ra}, x_{tva}, x_{tr}, \ldots$

feature

Linear model: $y = b + \sum w_i x_i$

w_i: weight权重, b: bias 偏差

Step 2: Goodness of Function

$y = \omega x + b$

A set of function

Model

$f_1, f_2 \cdots$

function input:	function Output (scalar):
x^1 21	$\hat{y}^1$ $8,350.0
x^2 198	$\hat{y}^2$ $25,300.0

Training Data

Step 2: Goodness of Function

Training Data: 12 months
训练数据：12个月实际营收

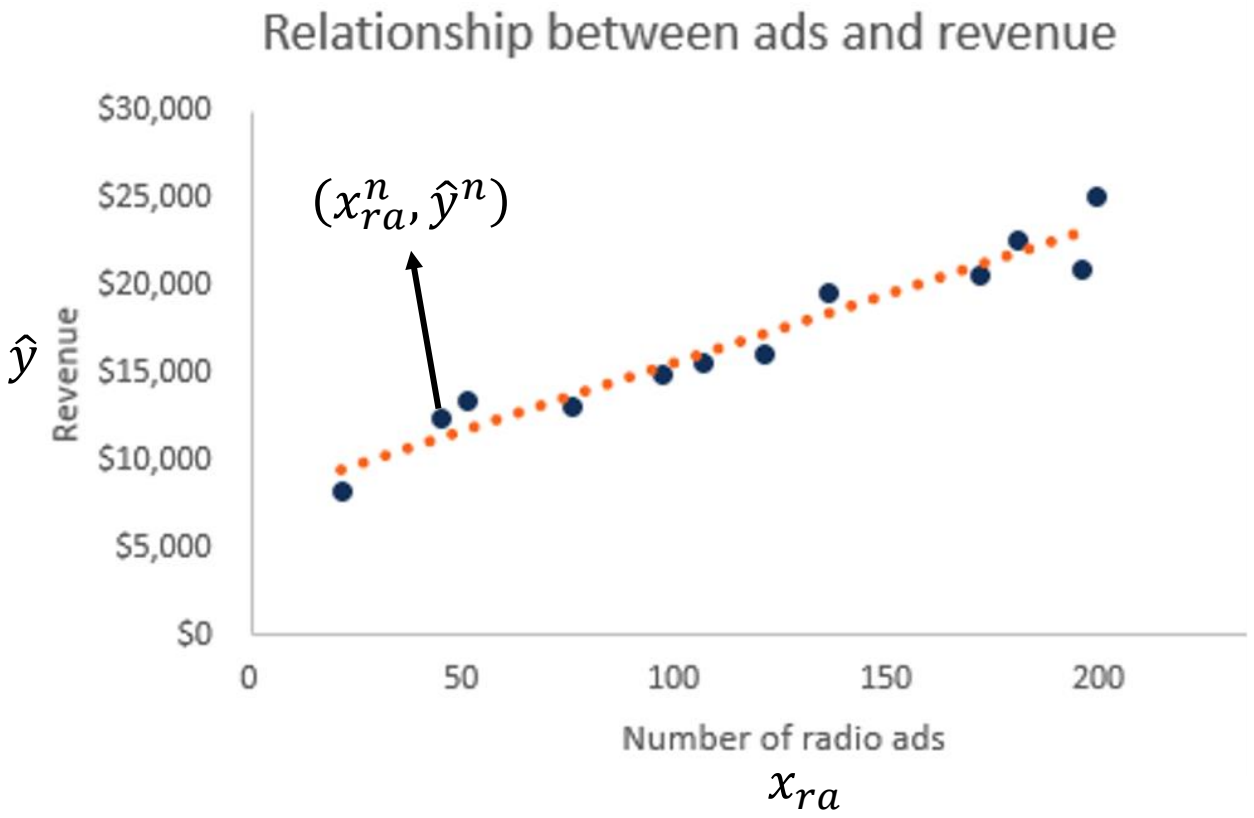
$(x^1, \hat{y}^1)$

$(x^2, \hat{y}^2)$

⋮

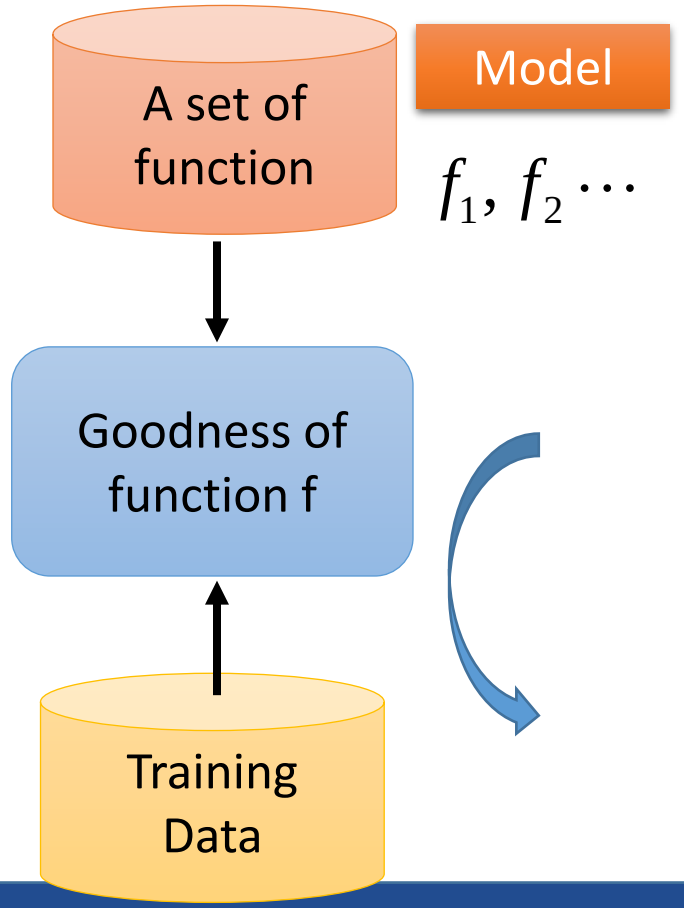
$(x^{12}, \hat{y}^{12})$

This is real data.
真实数据



Step 2: Goodness of Function

$y = \omega x + b$



Loss function 损失函数L:

Input: a function, output: how bad it is
输入：一个函数；输出：函数值

Estimation error 估计误差

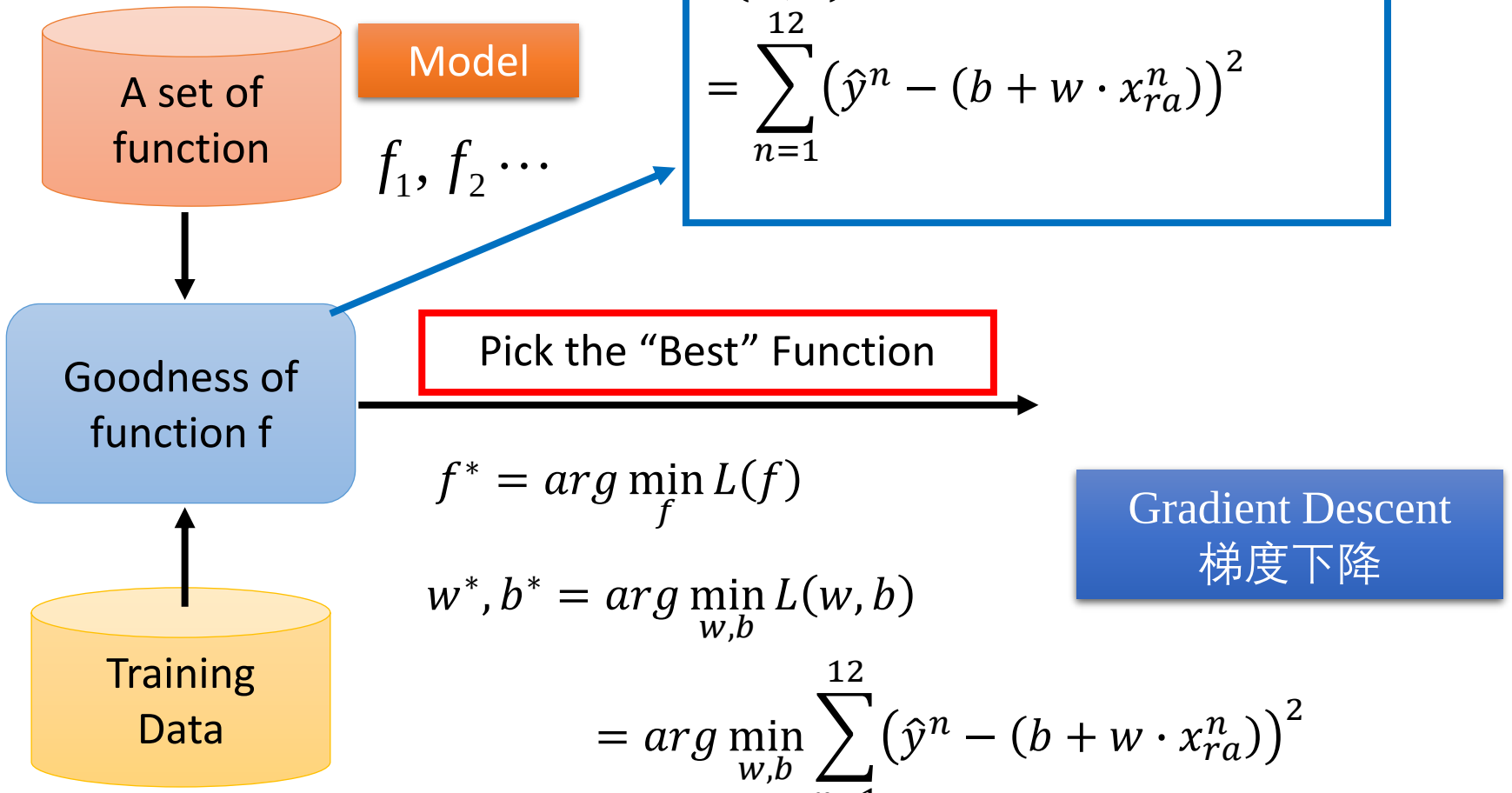
$$L(f) = \sum_{n=1}^{12} \left(\hat{y}^n - f(x_{ra}^n)\right)^2$$

Sum over examples 求和

Estimated y based on input function 函数值

$$L(w, b) = \sum_{n=1}^{12} \left(\hat{y}^n - (b + w \cdot x_{ra}^n)\right)^2$$

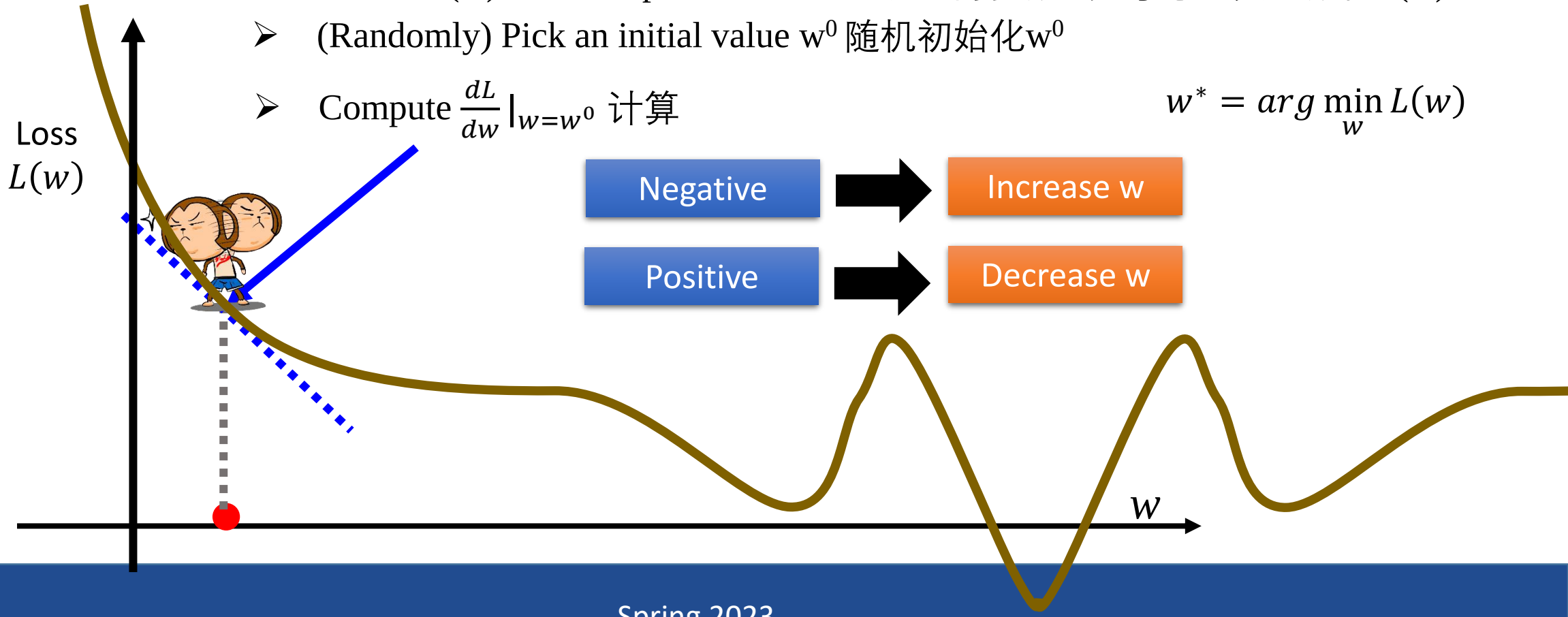
Step 3: Pick the Best Function



Step 3: Pick the Best Function

- Consider loss function $L(w)$ with one parameter w: 对于当前参数w，考虑损失函数值$L(w)$
  - (Randomly) Pick an initial value w^0 随机初始化w^0
  - Compute $\frac{dL}{dw}|_{w=w^0}$ 计算

$$w^* = arg\min_w L(w)$$

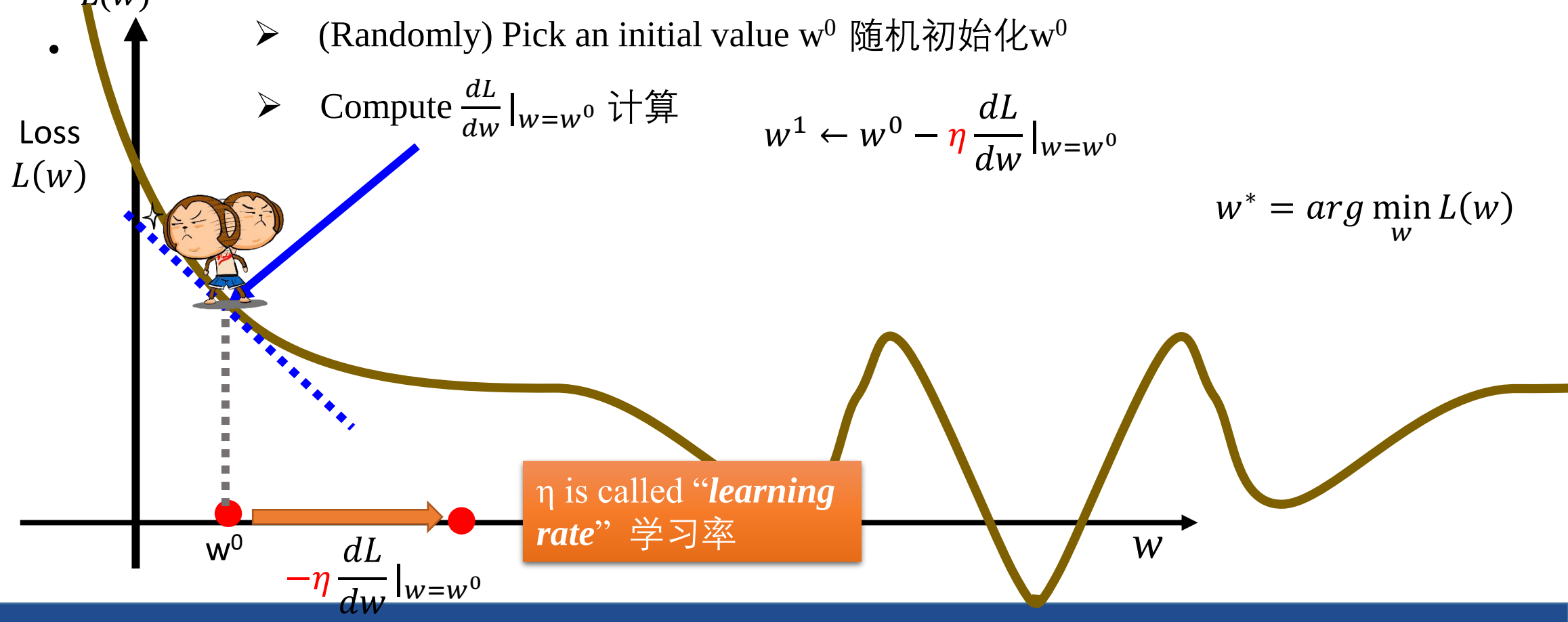


Step 3: Pick the Best Function

- Consider loss function $L(w)$ with one parameter w: 对于当前参数w，考虑损失函数值 $L(w)$
-
  - ➢ (Randomly) Pick an initial value w^0 随机初始化w^0
  - ➢ Compute $\frac{dL}{dw}|_{w=w^0}$ 计算

$$w^1 \leftarrow w^0 - \eta \frac{dL}{dw}|_{w=w^0}$$

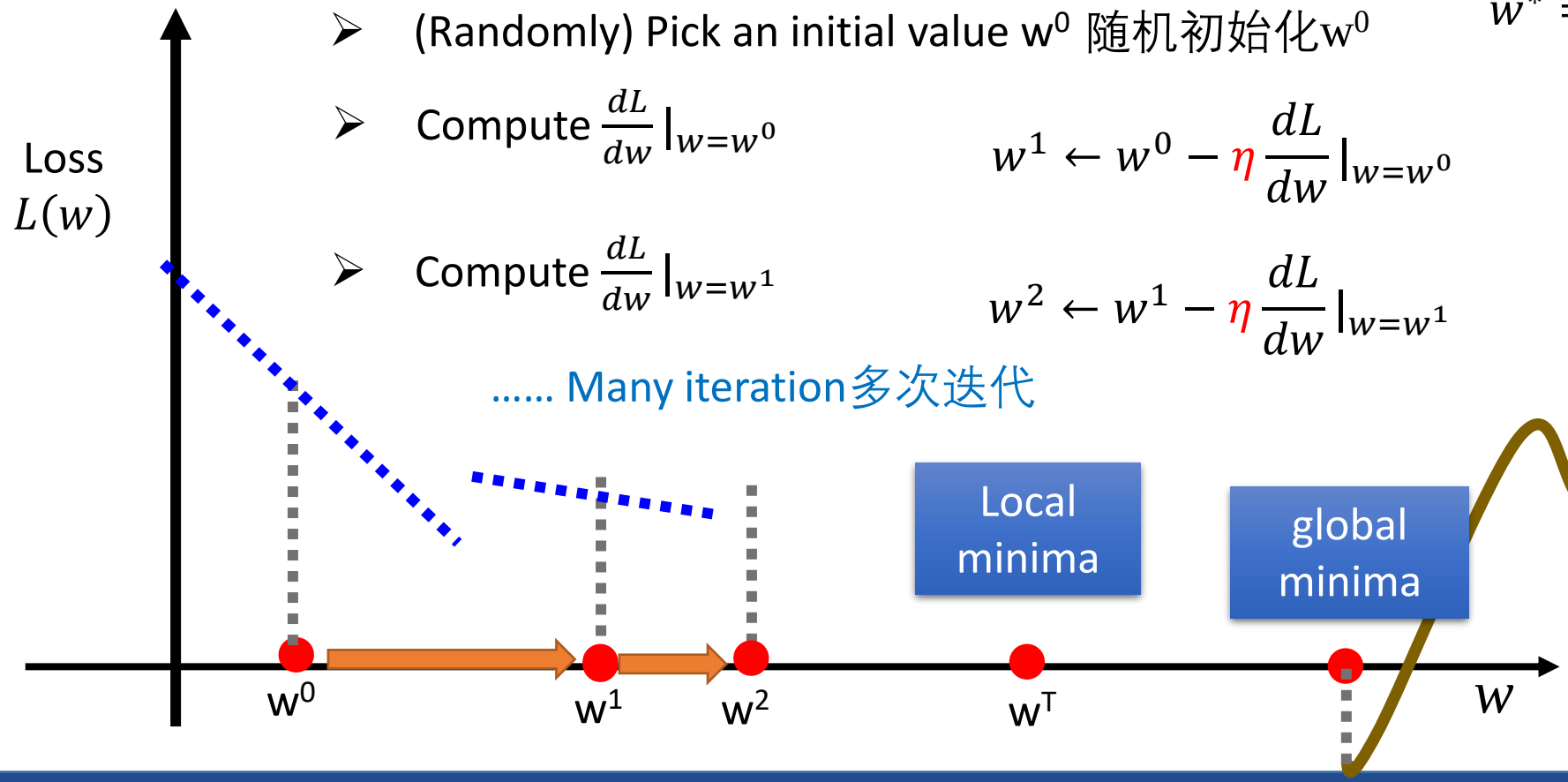
$$w^* = arg \min_w L(w)$$



Step 3: Pick the Best Function

- Consider loss function $L(w)$ with one parameter w: 对于当前参数w，考虑损失函数值

$$w^* = arg\min_w L(w)$$



Step 3: Pick the Best Function

$$\begin{bmatrix} \frac{\partial L}{\partial w} \\ \frac{\partial L}{\partial b} \end{bmatrix}$$ Gradient 梯度

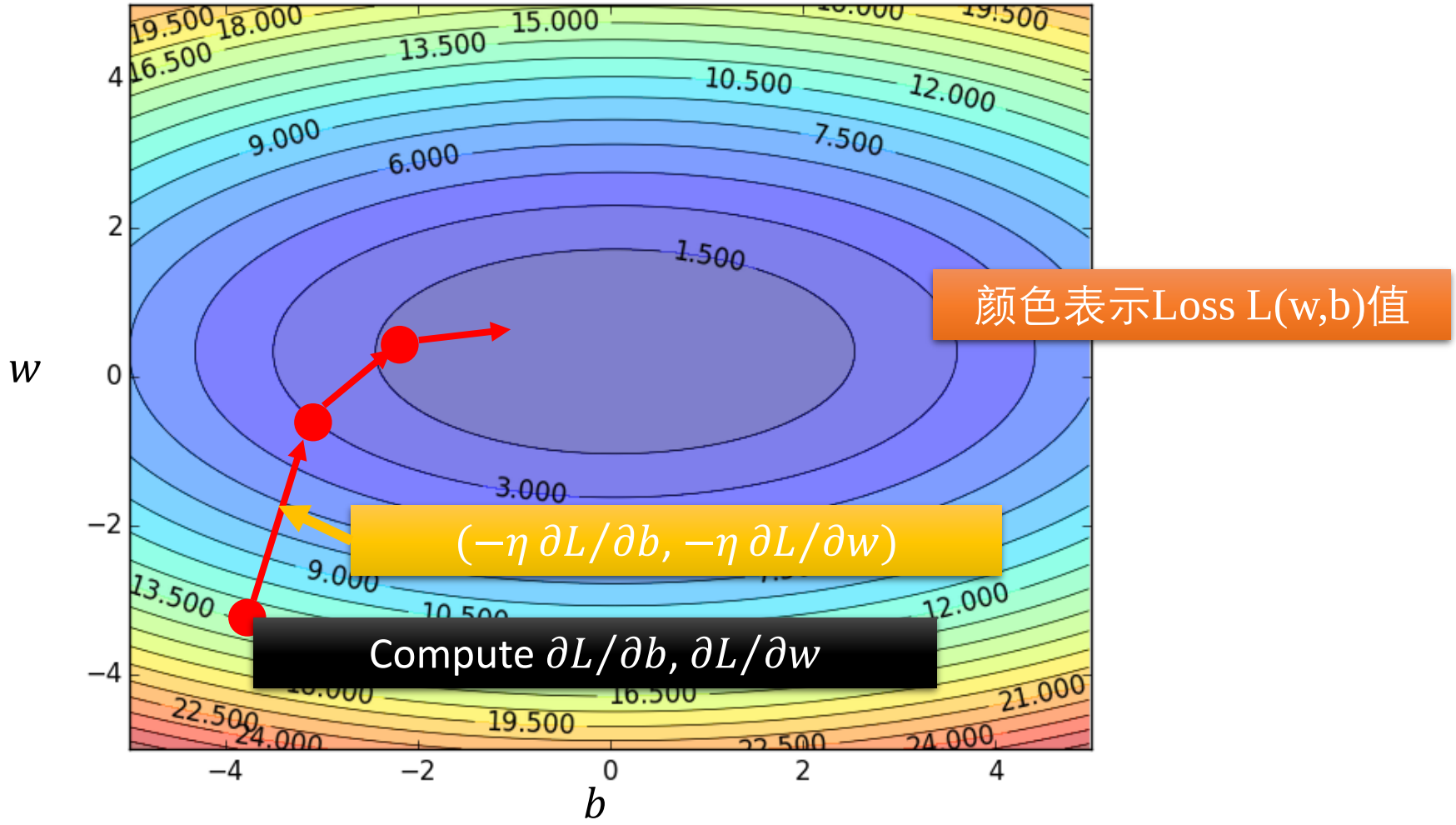
- How about two parameters?双参数 $w^*, b^* = arg \min_{w,b} L(w,b)$
  - (Randomly) Pick an initial value 随机初始化 w^0, b^0
  - Compute 计算 $\frac{\partial L}{\partial w}|_{w=w^0,b=b^0}, \frac{\partial L}{\partial b}|_{w=w^0,b=b^0}$

$$w^1 \leftarrow w^0 - \eta \frac{\partial L}{\partial w}|_{w=w^0,b=b^0} \qquad b^1 \leftarrow b^0 - \eta \frac{\partial L}{\partial b}|_{w=w^0,b=b^0}$$

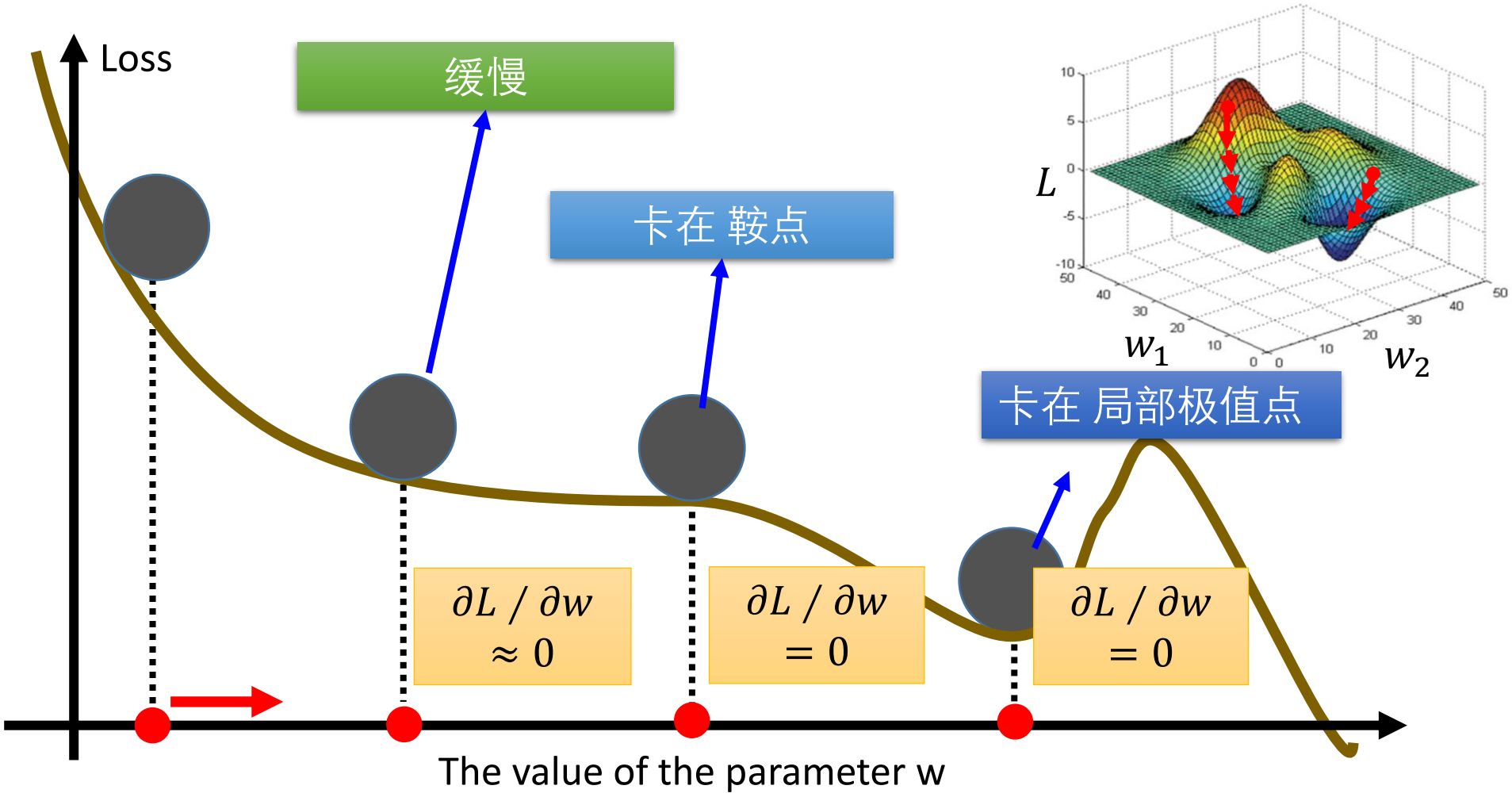
  - Compute 计算 $\frac{\partial L}{\partial w}|_{w=w^1,b=b^1}, \frac{\partial L}{\partial b}|_{w=w^1,b=b^1}$

$$w^2 \leftarrow w^1 - \eta \frac{\partial L}{\partial w}|_{w=w^1,b=b^1} \qquad b^2 \leftarrow b^1 - \eta \frac{\partial L}{\partial b}|_{w=w^1,b=b^1}$$

Step 3: Pick the Best Function



Step 3: Pick the Best Function



Three steps

Define a set of functions A

Goodness of function B

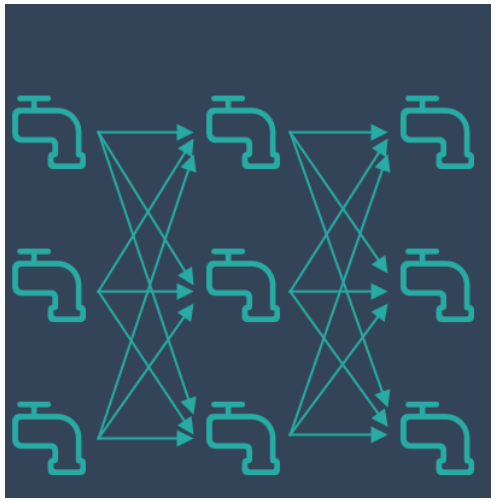
Pick the best function C

03 Back Propagation

高效快速必备：反向传播

Review: Neural Networks

函数集合

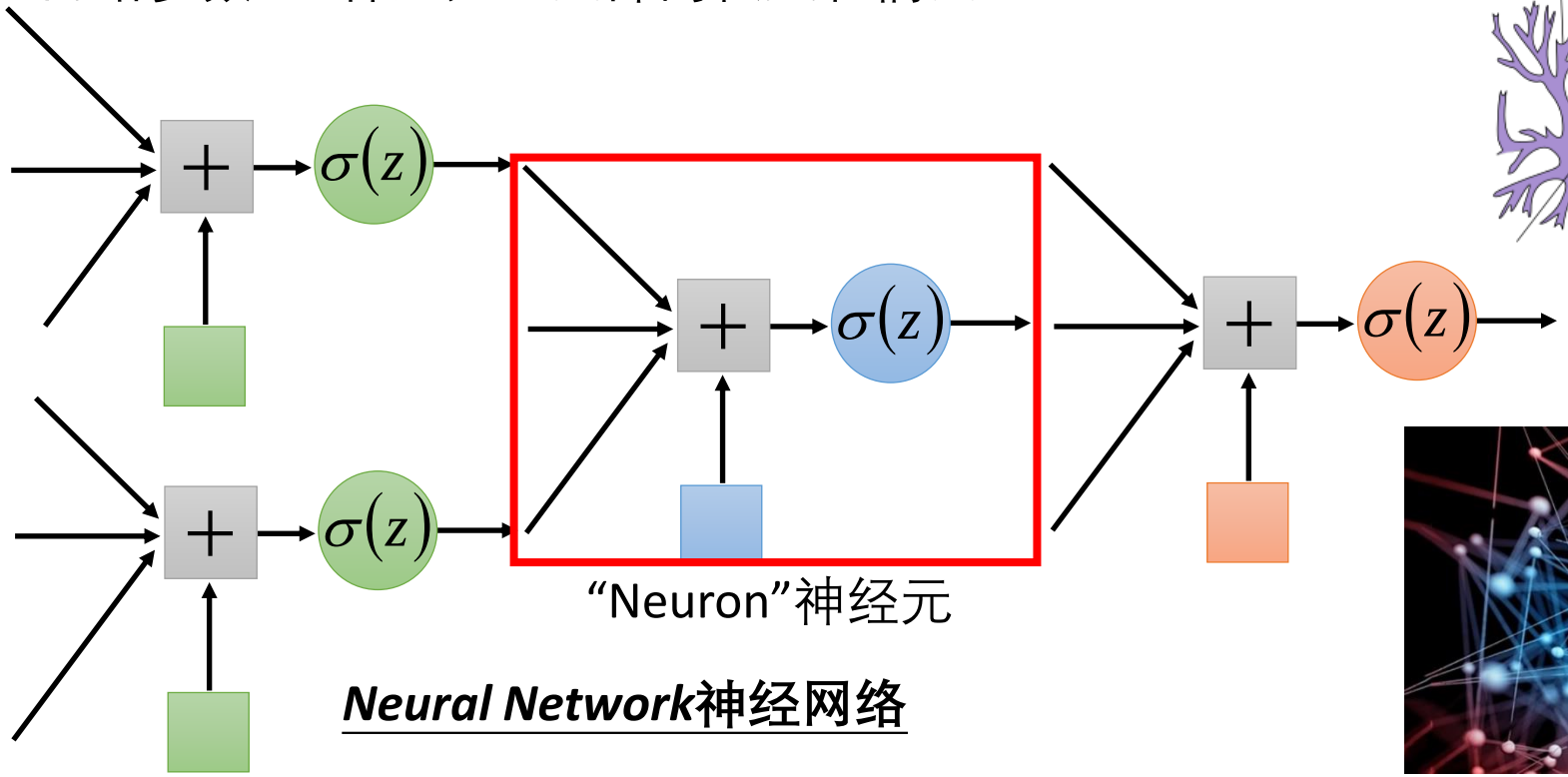


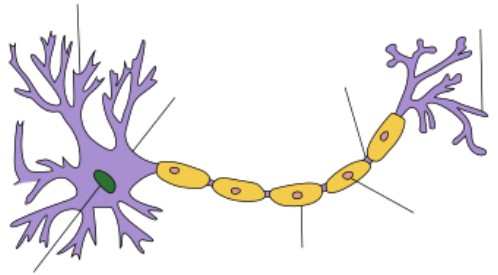
神经网络

Step 3: Pick the Best Function

Network parameter θ: all the weights and biases in the "neurons"

神经网络参数θ：神经元里全部的权重和偏差



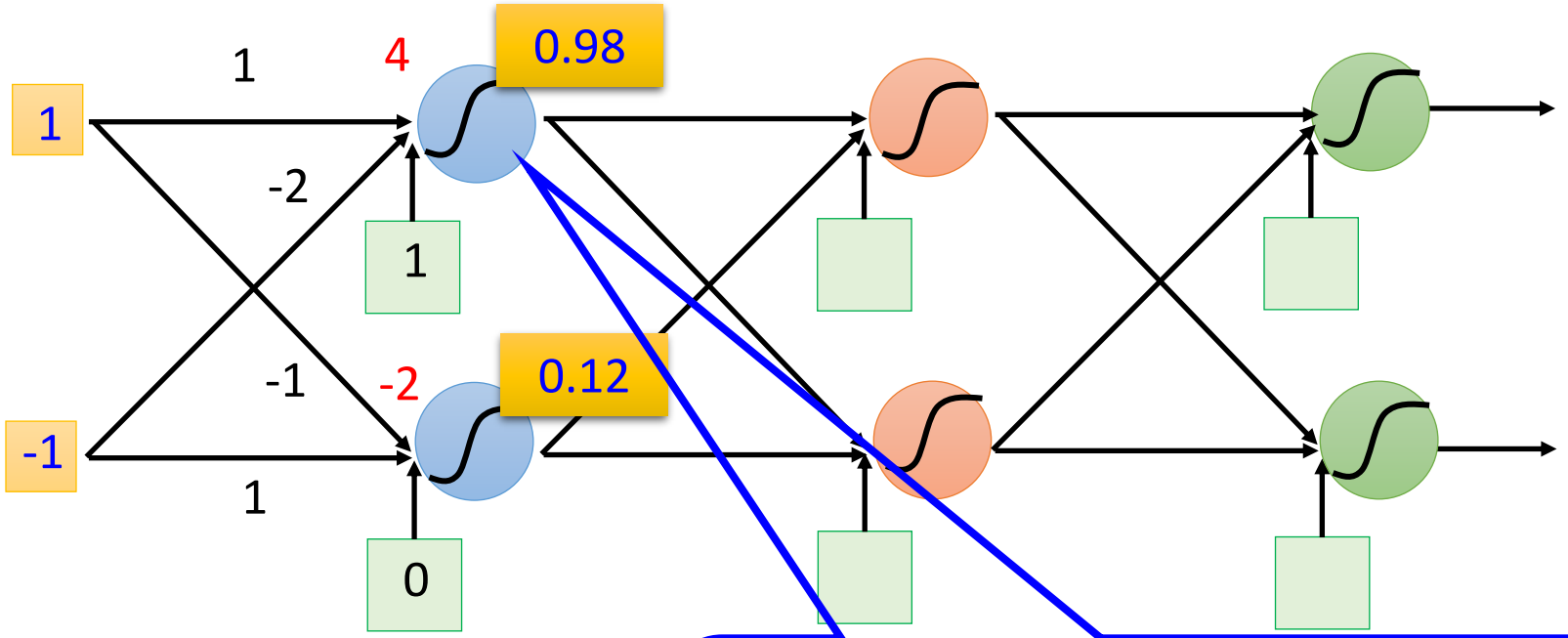


"Neuron"神经元

***Neural Network*神经网络**

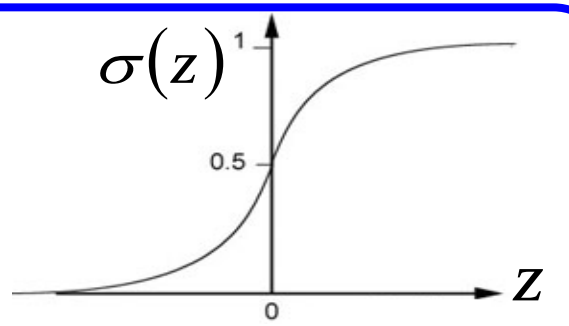
Different connection leads to different network structures不同连接方式导致网络结构差异

Fully Connect Network (FCN): Feedforward

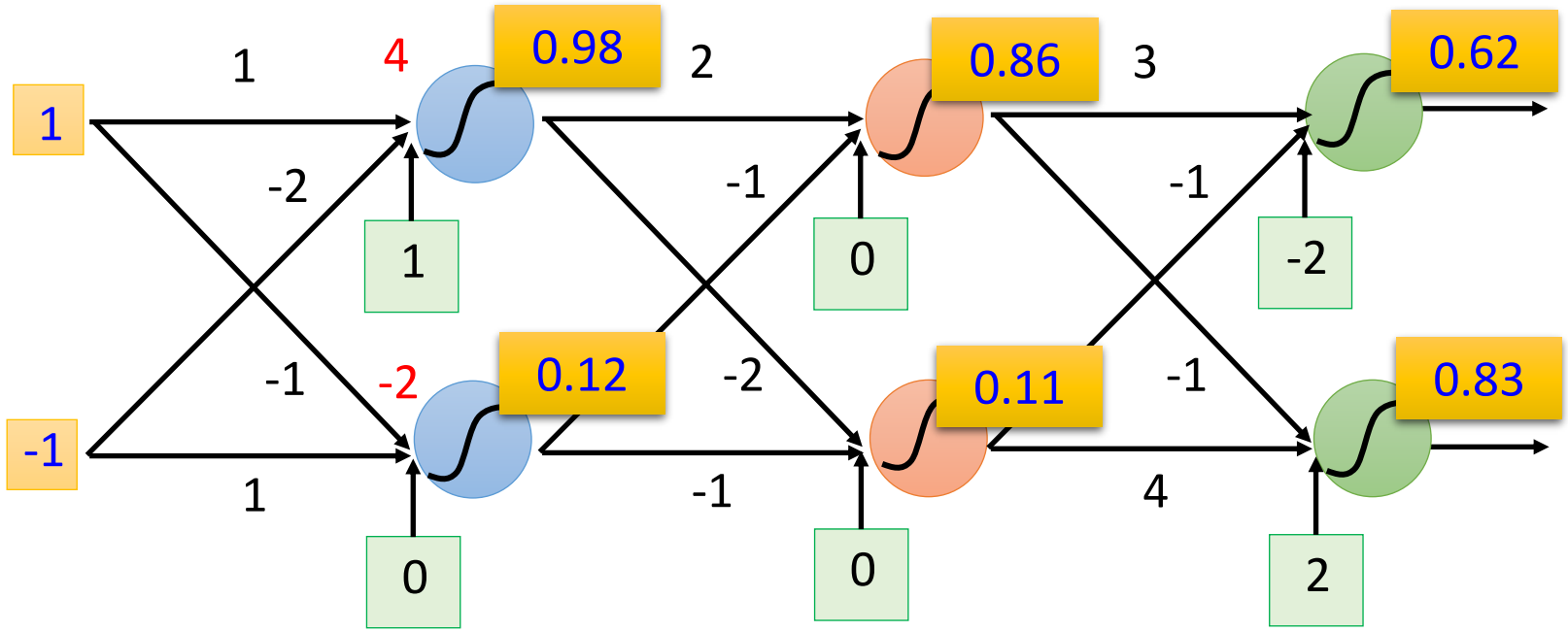


Sigmoid Function

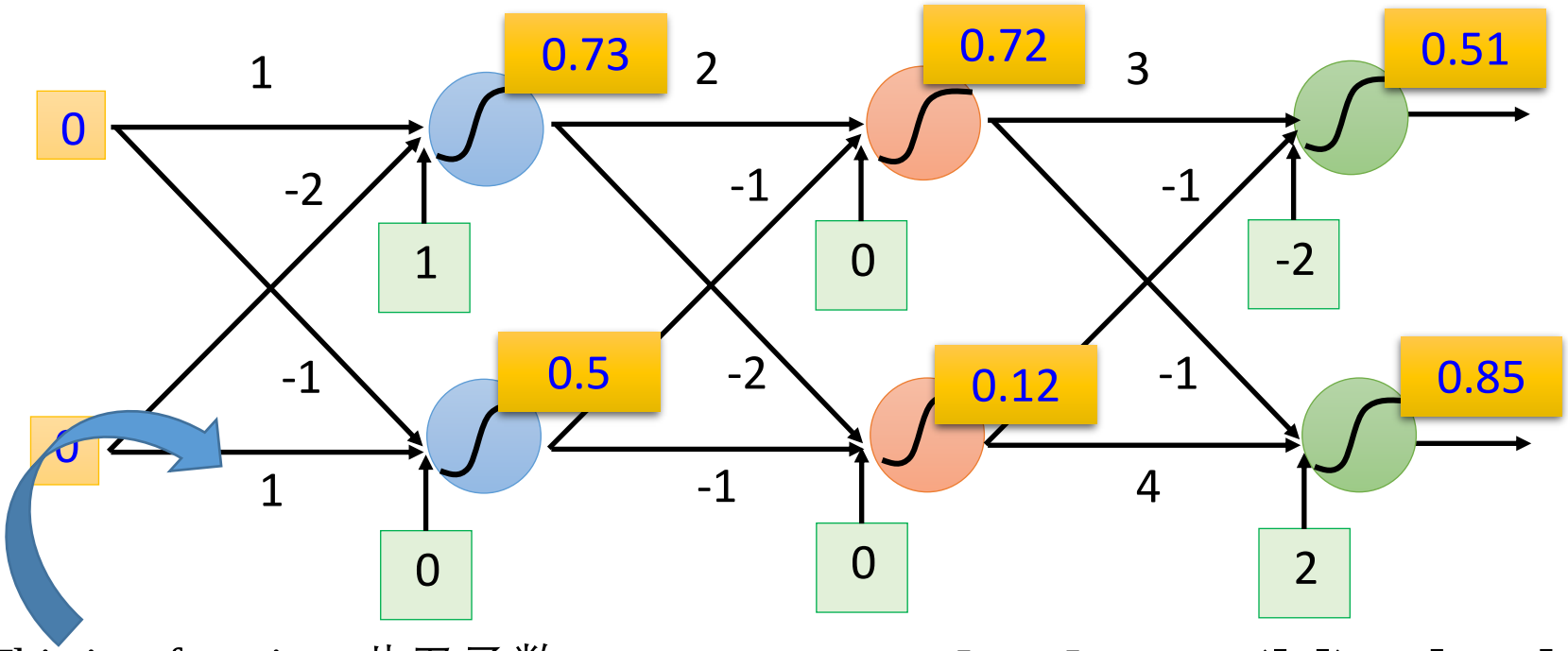
$$\sigma(z) = \frac{1}{1+e^{-z}}$$



Fully Connect Network (FCN): Feedforward



Fully Connect Network (FCN): Feedforward



This is a function. 此乃函数

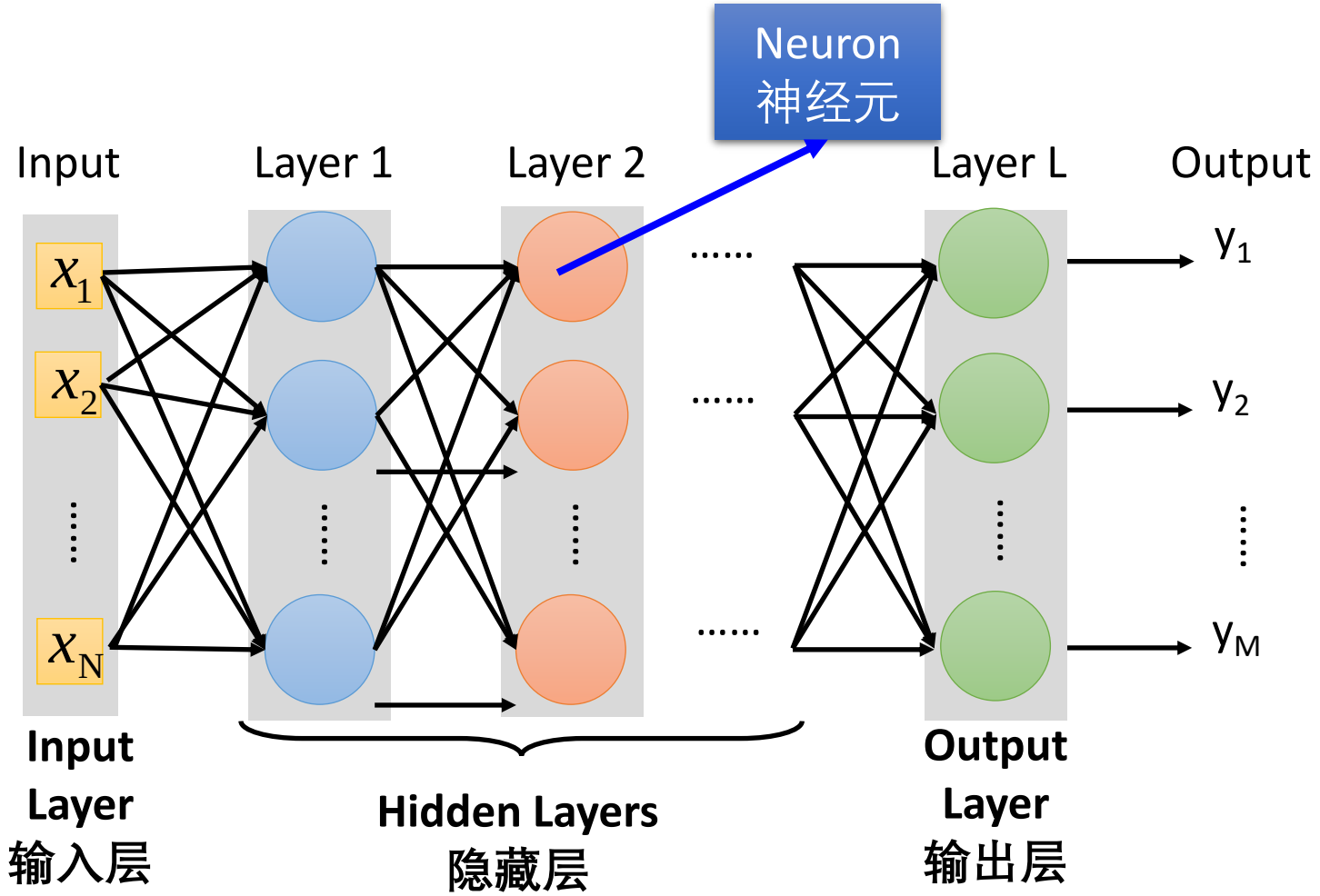
Input vector, output vector

输入矢量， 输出矢量

$$f\left(\begin{bmatrix}1\\-1\end{bmatrix}\right)=\begin{bmatrix}0.62\\0.83\end{bmatrix} \qquad f\left(\begin{bmatrix}0\\0\end{bmatrix}\right)=\begin{bmatrix}0.51\\0.85\end{bmatrix}$$

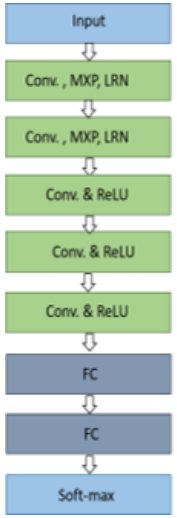
Given network structure, define ***a function set***

Fully Connect Network (FCN): Feedforward



Deep Neural Networks

8 layers



Inaccuracy: 16.4%

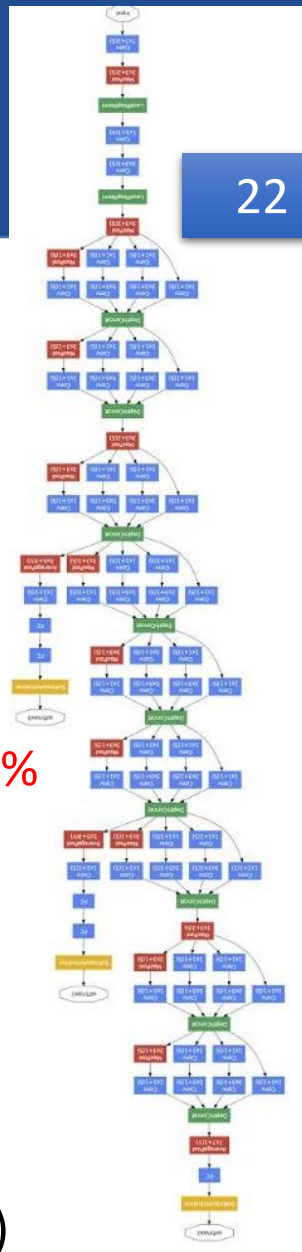
AlexNet (2012)

19 layers

7.3%

VGG (2014)

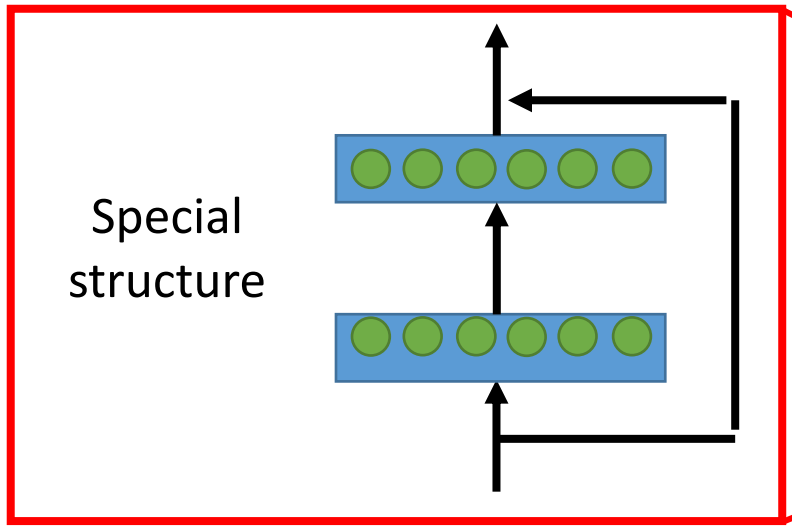
22 layers



6.7%

GoogleNet (2014)

Deep Neural Networks



16.4% AlexNet (2012)

7.3% VGG (2014)

6.7% GoogleNet (2014)

3.57% Residual Net (2015)

Dense Net (2017)

Review: Chain Rule

Case 1

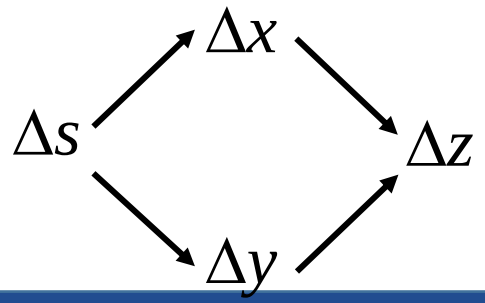
$$y = g(x) \qquad z = h(y)$$

$$\Delta x \to \Delta y \to \Delta z$$

$$\frac{dz}{dx} = \frac{dz}{dy}\frac{dy}{dx}$$

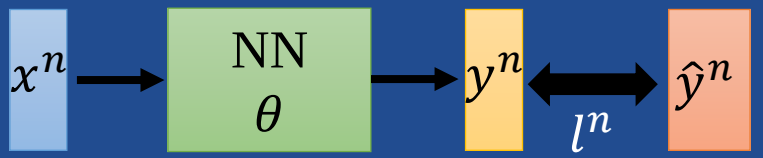
Case 2

$$x = g(s) \qquad y = h(s) \qquad z = k(x, y)$$



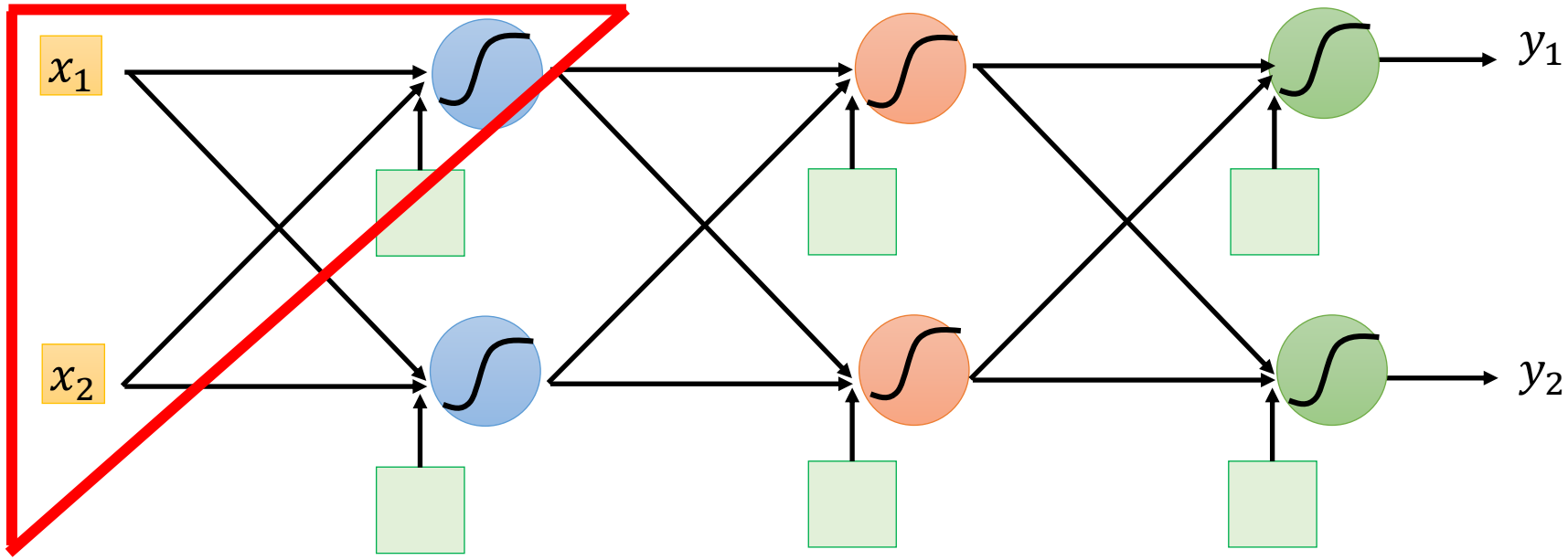
$$\frac{dz}{ds} = \frac{\partial z}{\partial x}\frac{dx}{ds} + \frac{\partial z}{\partial y}\frac{dy}{ds}$$

Backpropagation

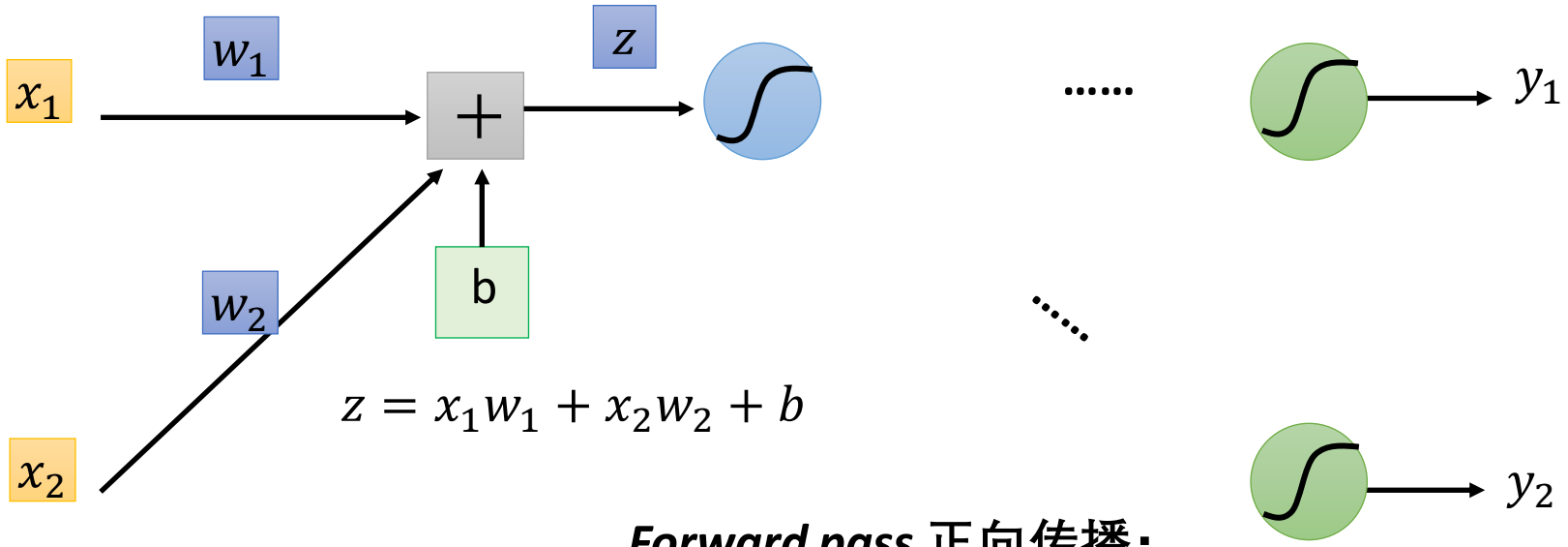


$$L(\theta) = \sum_{n=1}^{N} l^n(\theta)$$

$$\frac{\partial L(\theta)}{\partial w} = \sum_{n=1}^{N} \frac{\partial l^n(\theta)}{\partial w}$$



Backpropagation



$$\frac{\partial l}{\partial w} = ? \qquad \frac{\partial z}{\partial w}\frac{\partial l}{\partial z}$$

(链式法则)

***Forward pass* 正向传播：**

Compute $\partial z/\partial w$ for all parameters 计算偏导数（所有参数）

***Backward pass* 反向传播：**

Compute $\partial l/\partial z$ for all activation function inputs z 计算偏导数（激活函数）

Deep Learning Frameworks

Q&A