Python人工智能

讲师：覃秉丰

人工智能/机器学习/神经网络/深度学习介绍

人工智能(AI)

AI Magazine Volume 27 Number 4 (2006) (© AAAI)

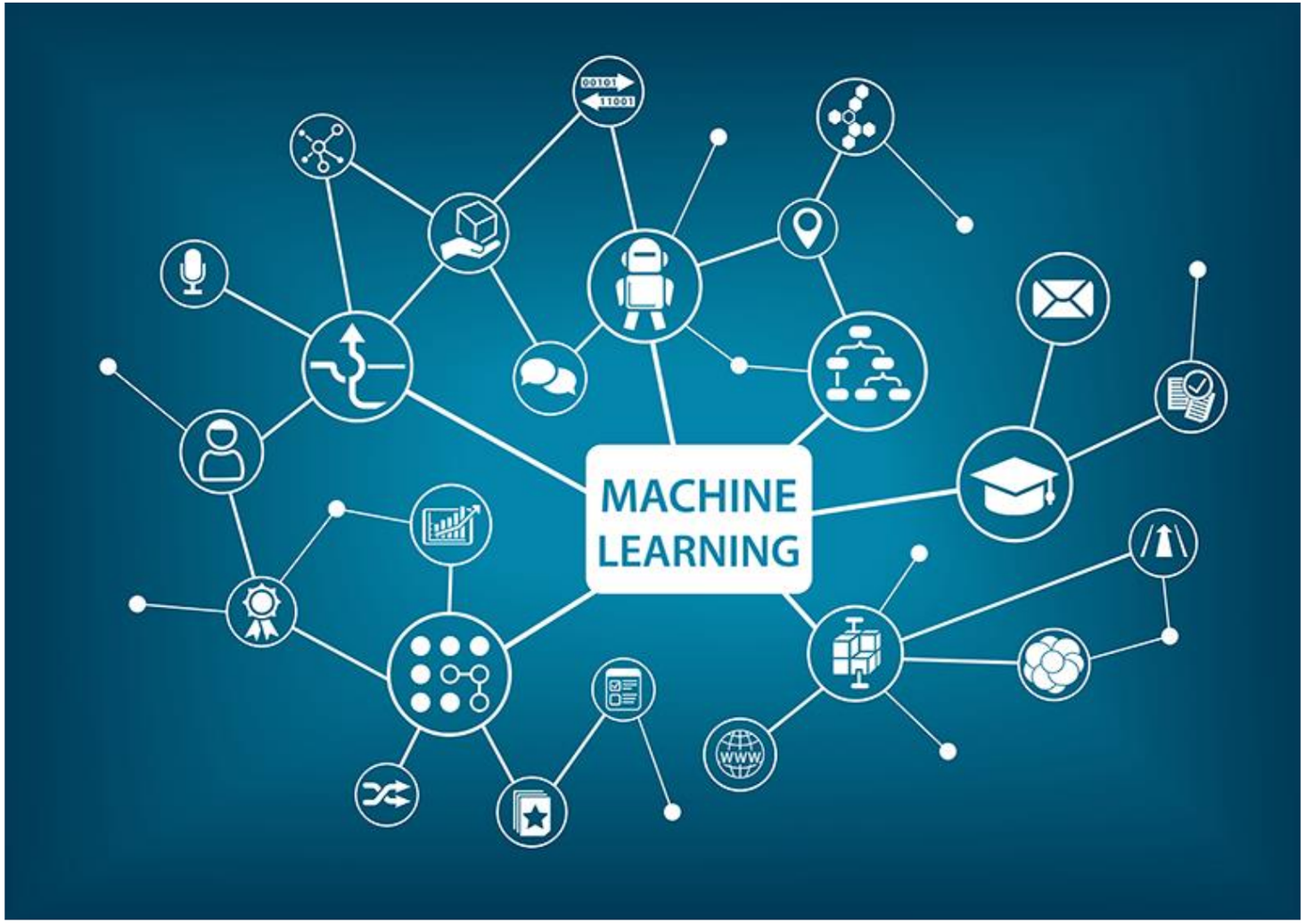
A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence

August 31, 1955

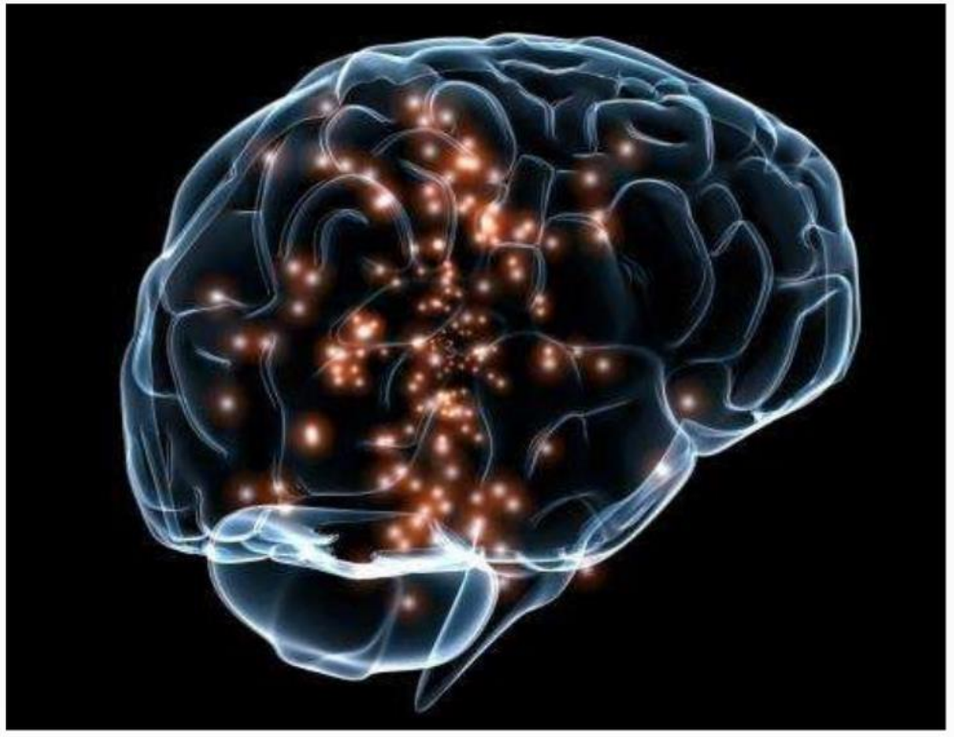
John McCarthy, Marvin L. Minsky,
Nathaniel Rochester,
and Claude E. Shannon

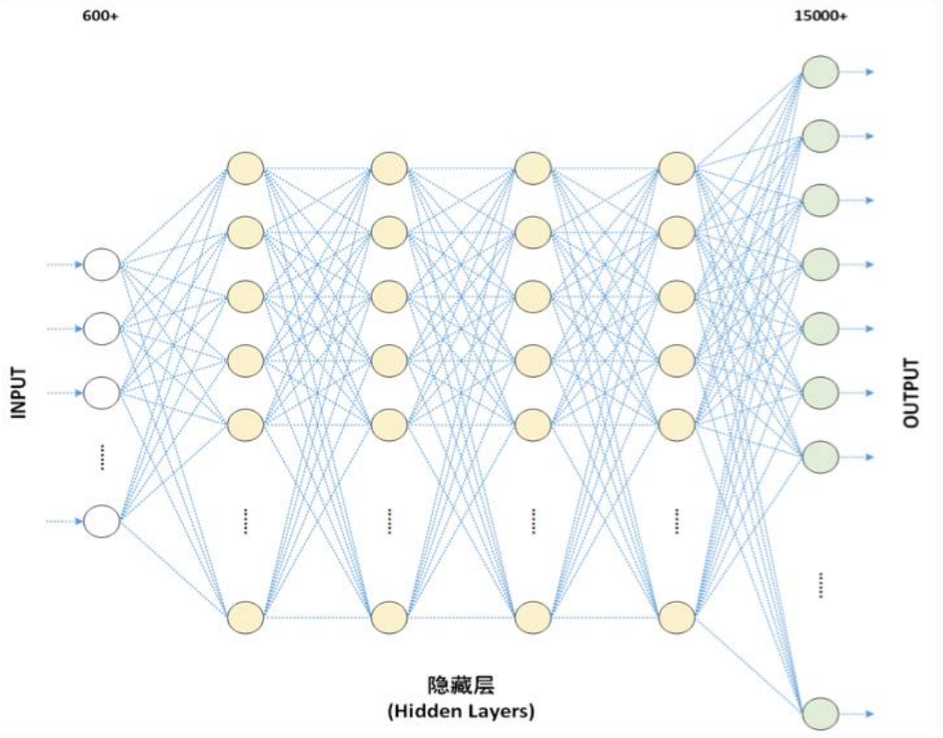
在人工智能50年大会上，5位1956年Dartmouth人工智能夏季研究会的与会者再相聚

机器学习(ML)

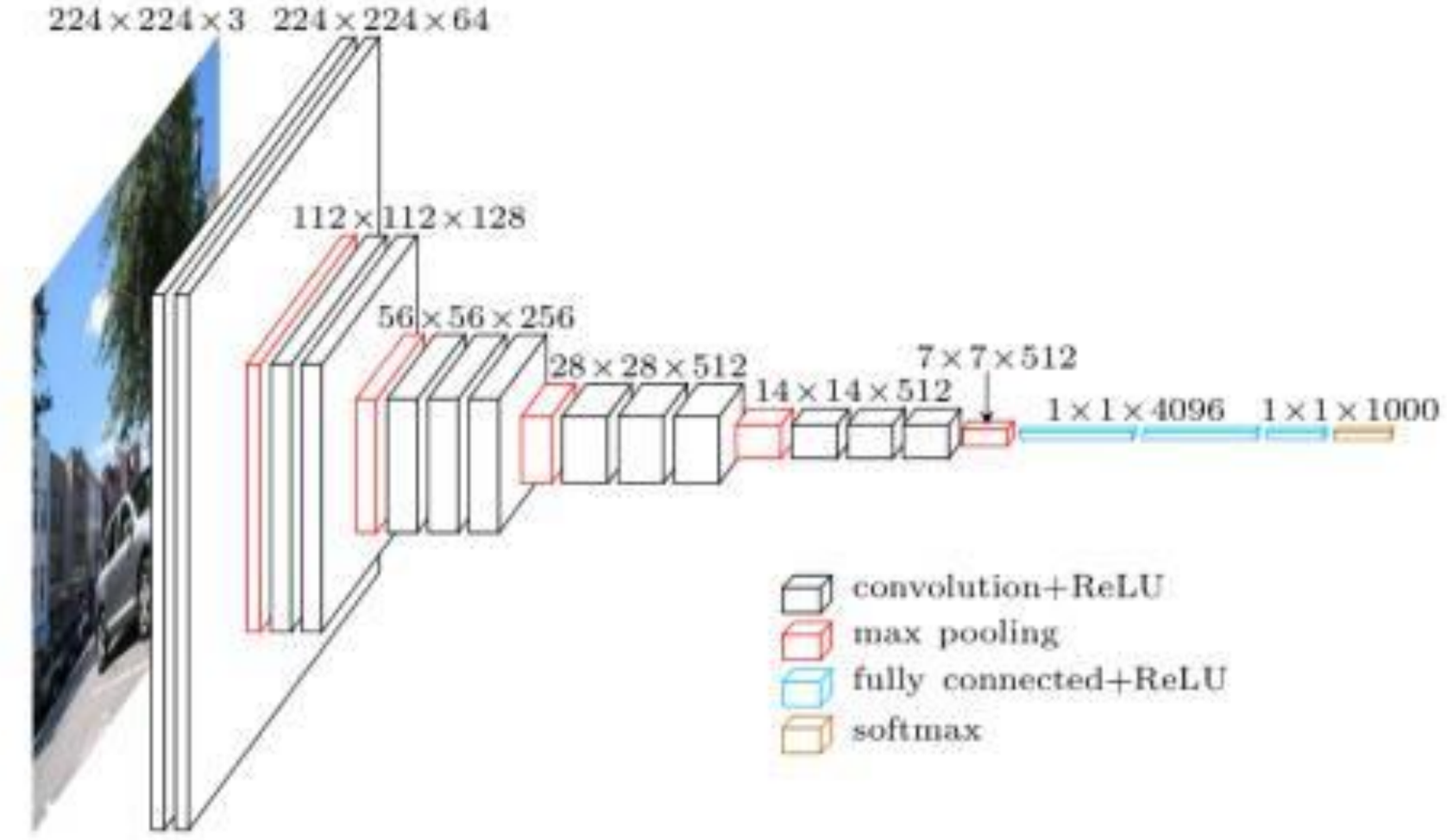


神经网络(NN)

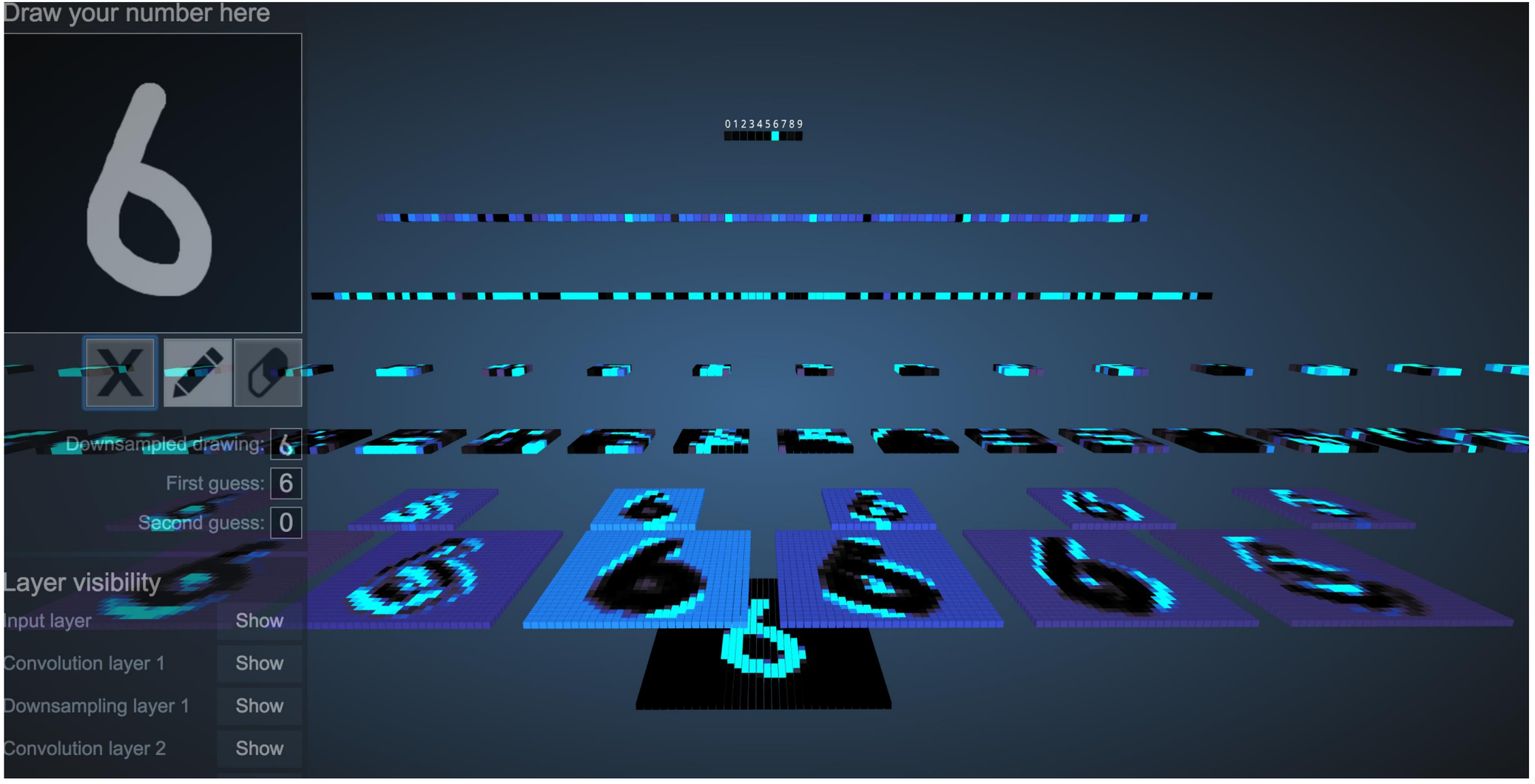




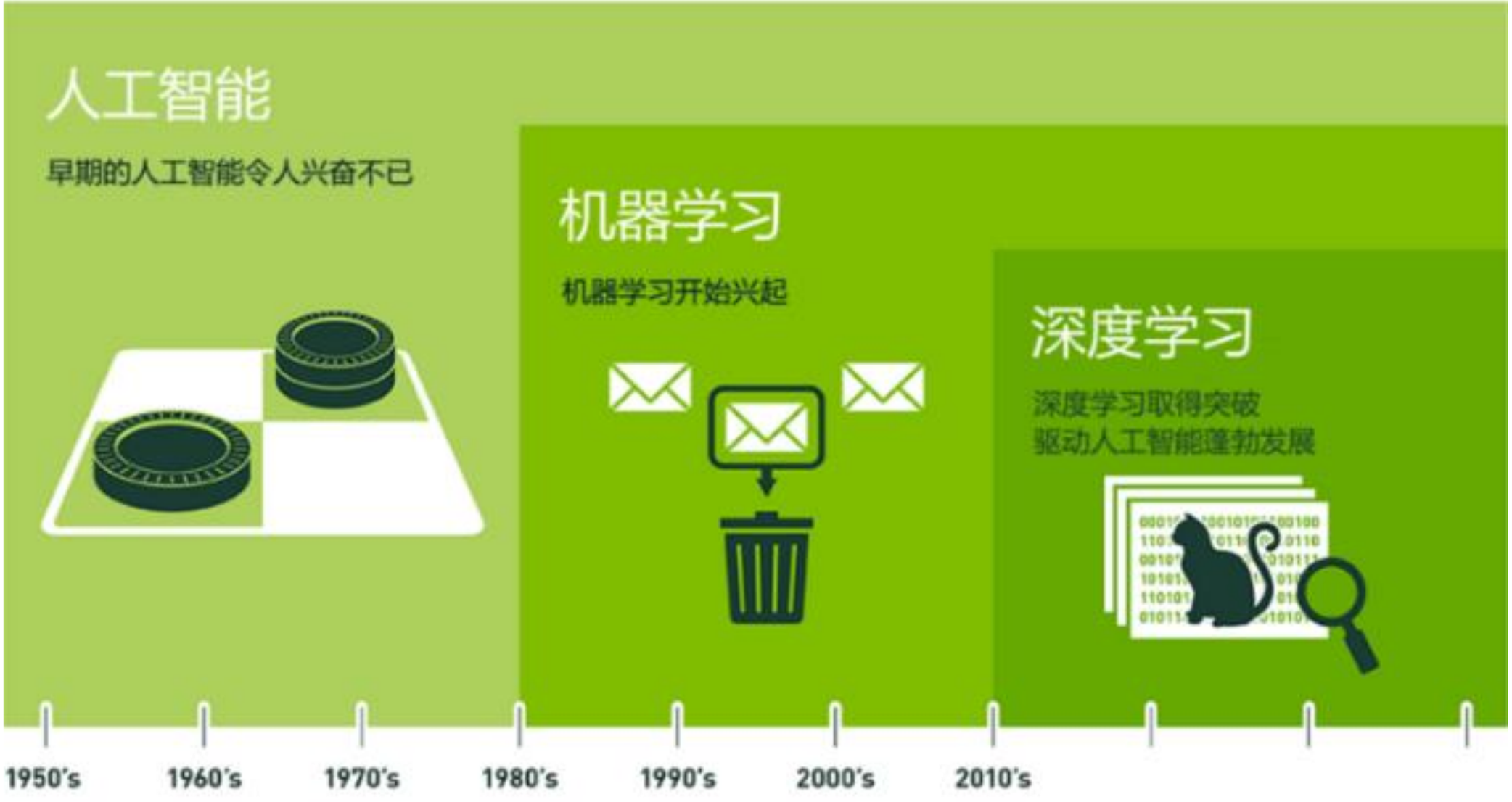
深度学习(DL)


224×224×3
224×224×64
112×112×128
56×56×256
28×28×512
14×14×512
7×7×512
1×1×4096
1×1×1000
convolution+ReLU
max pooling
fully connected+ReLU
softmax

CNN可视化

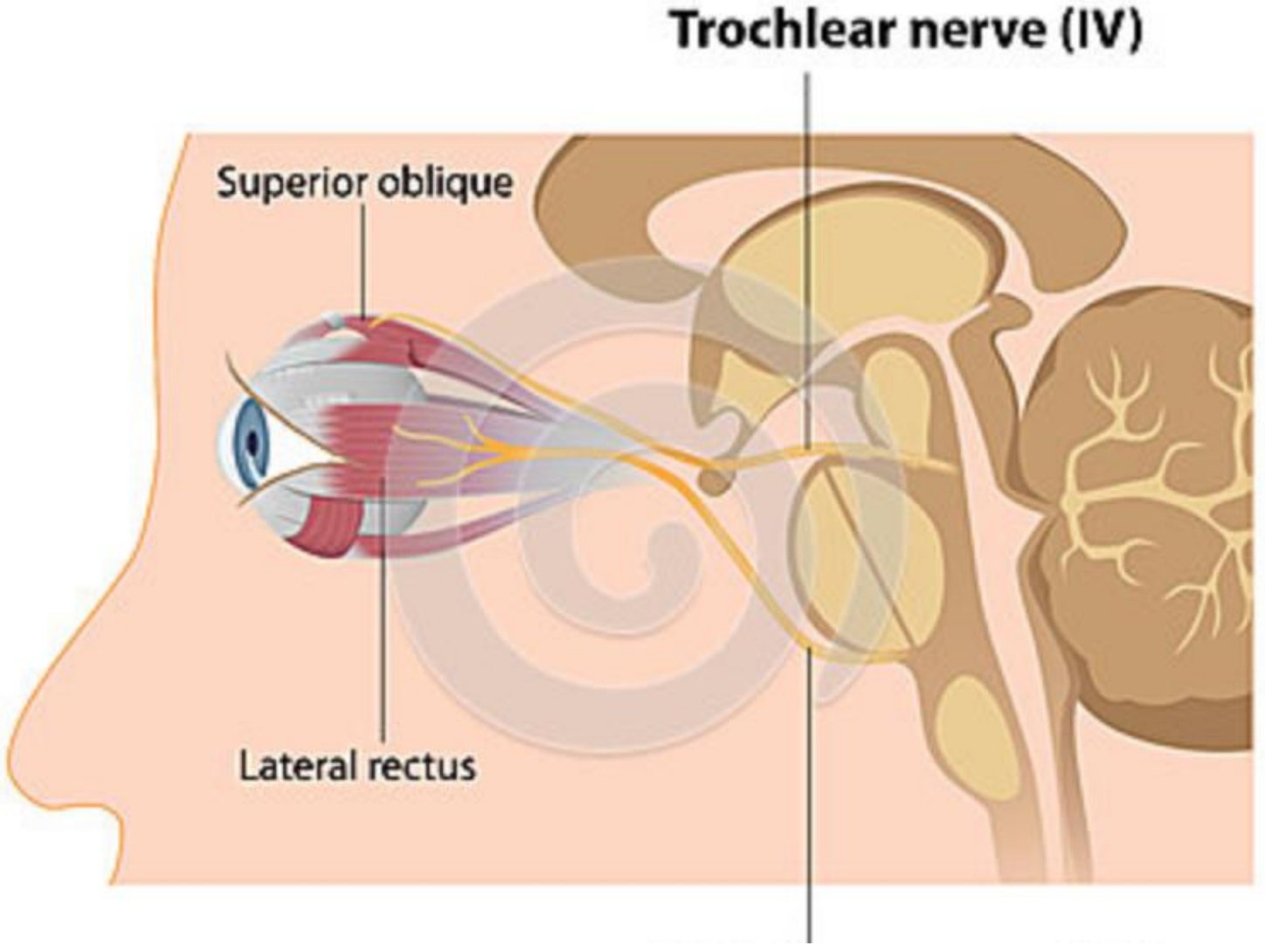

Draw your number here
0 1 2 3 4 5 6 7 8 9
Downsampled drawing:
First guess: 6
Second guess: 0
Layer visibility
Input layer Show
Convolution layer 1 Show
Downsampling layer 1 Show
Convolution layer 2 Show

人工智能/机器学习/神经网络/深度学习

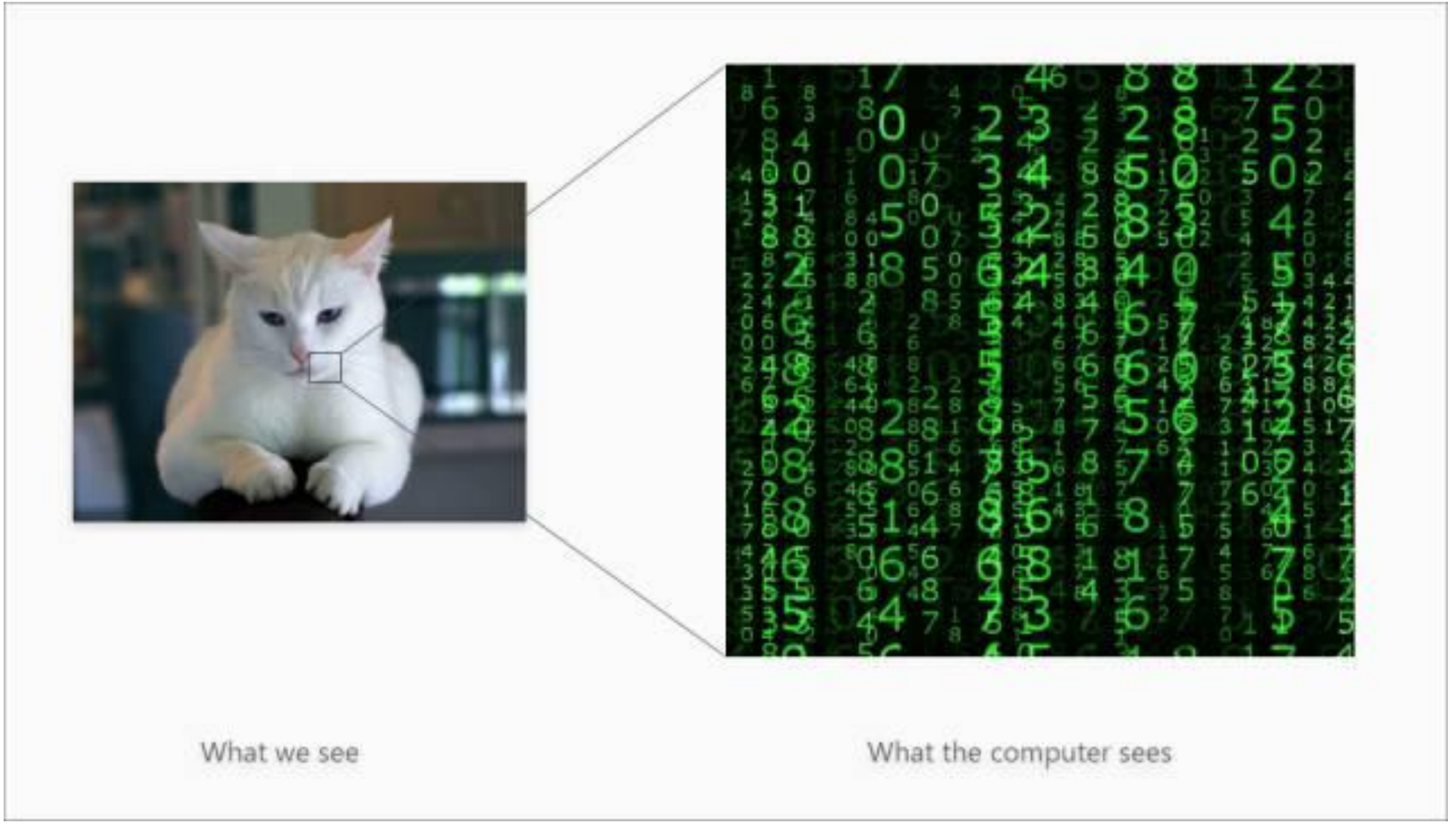


理解深度学习与传统算法的区别

看见≠看懂



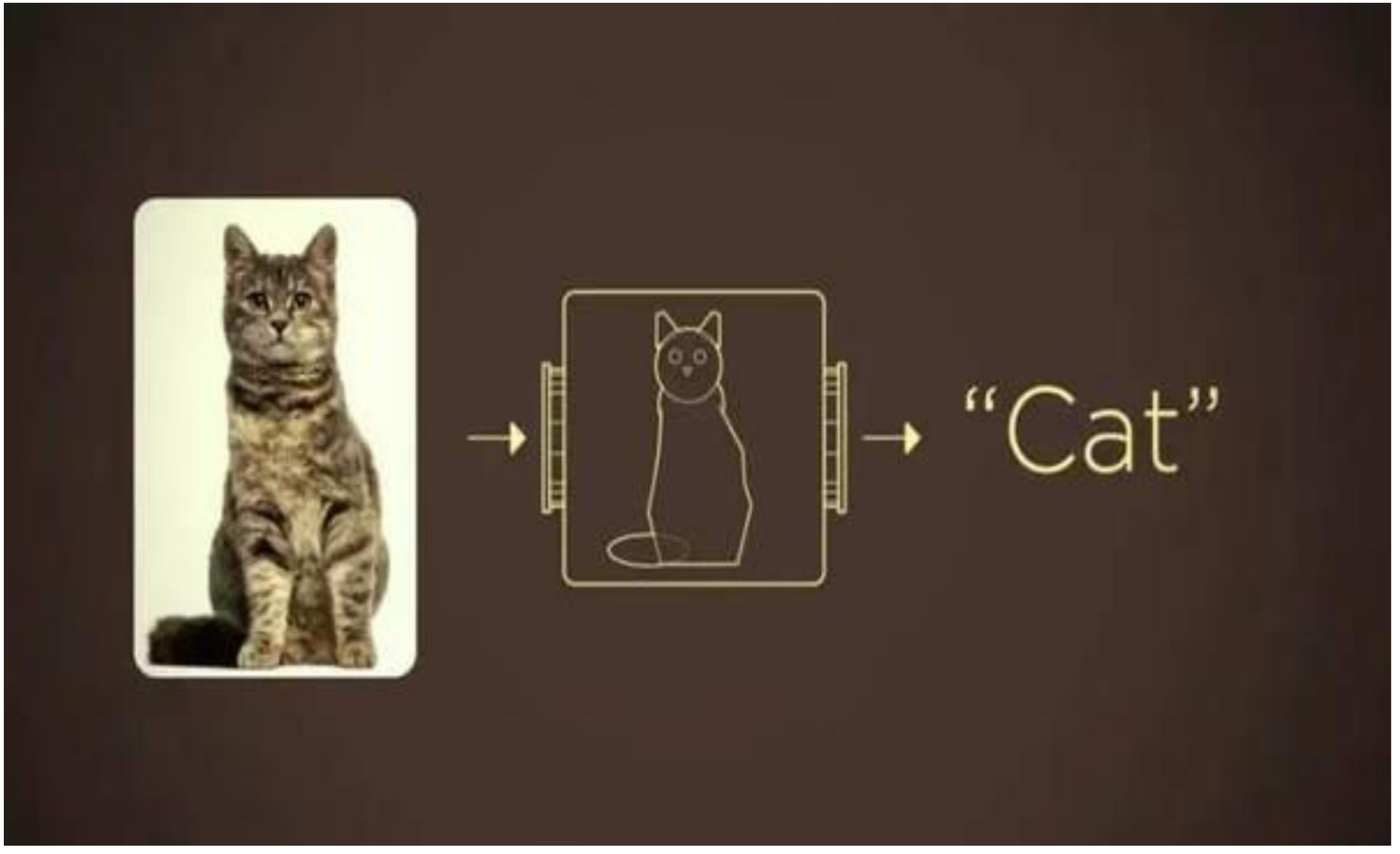
识别猫



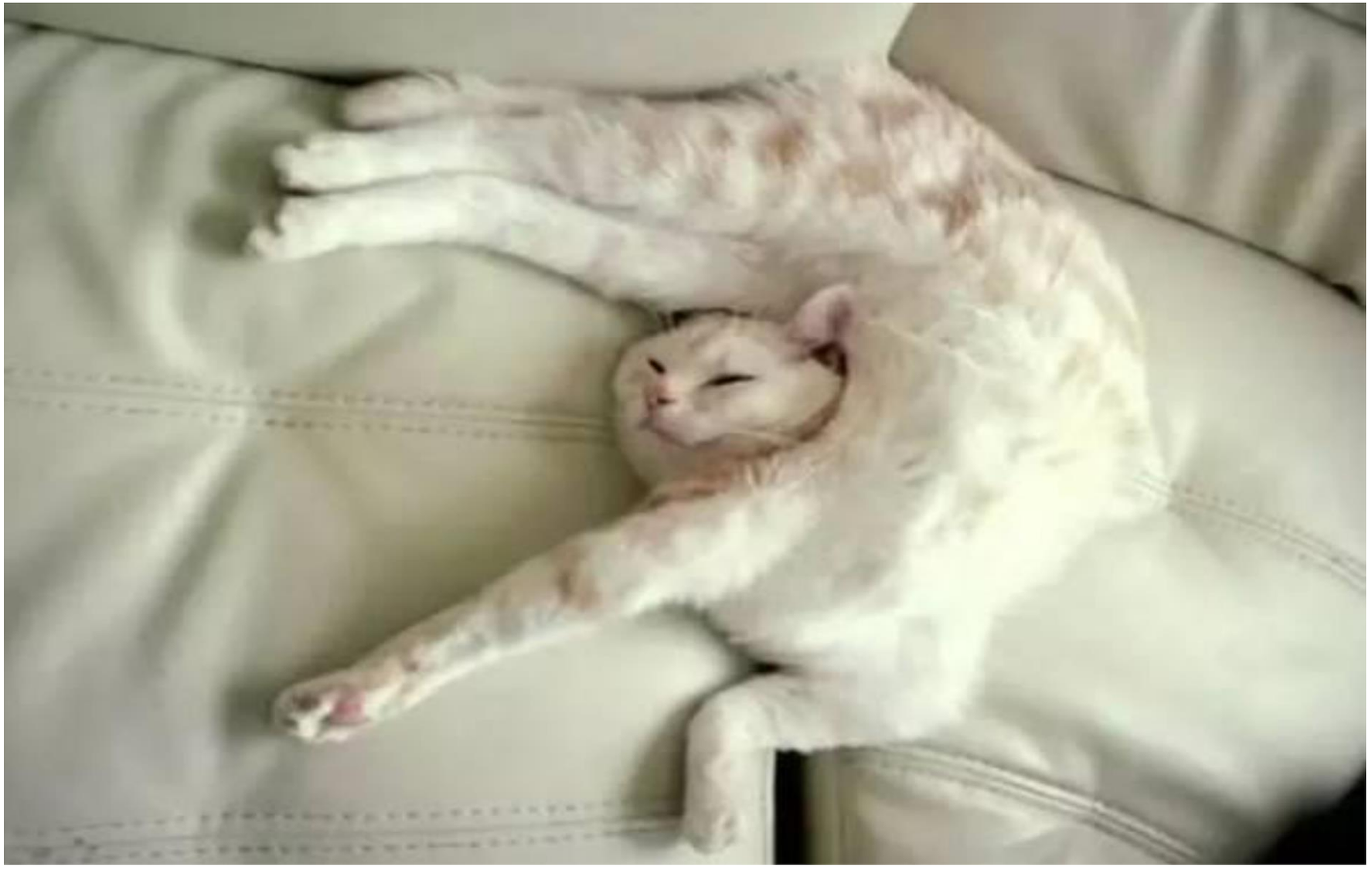
识别猫

识别猫

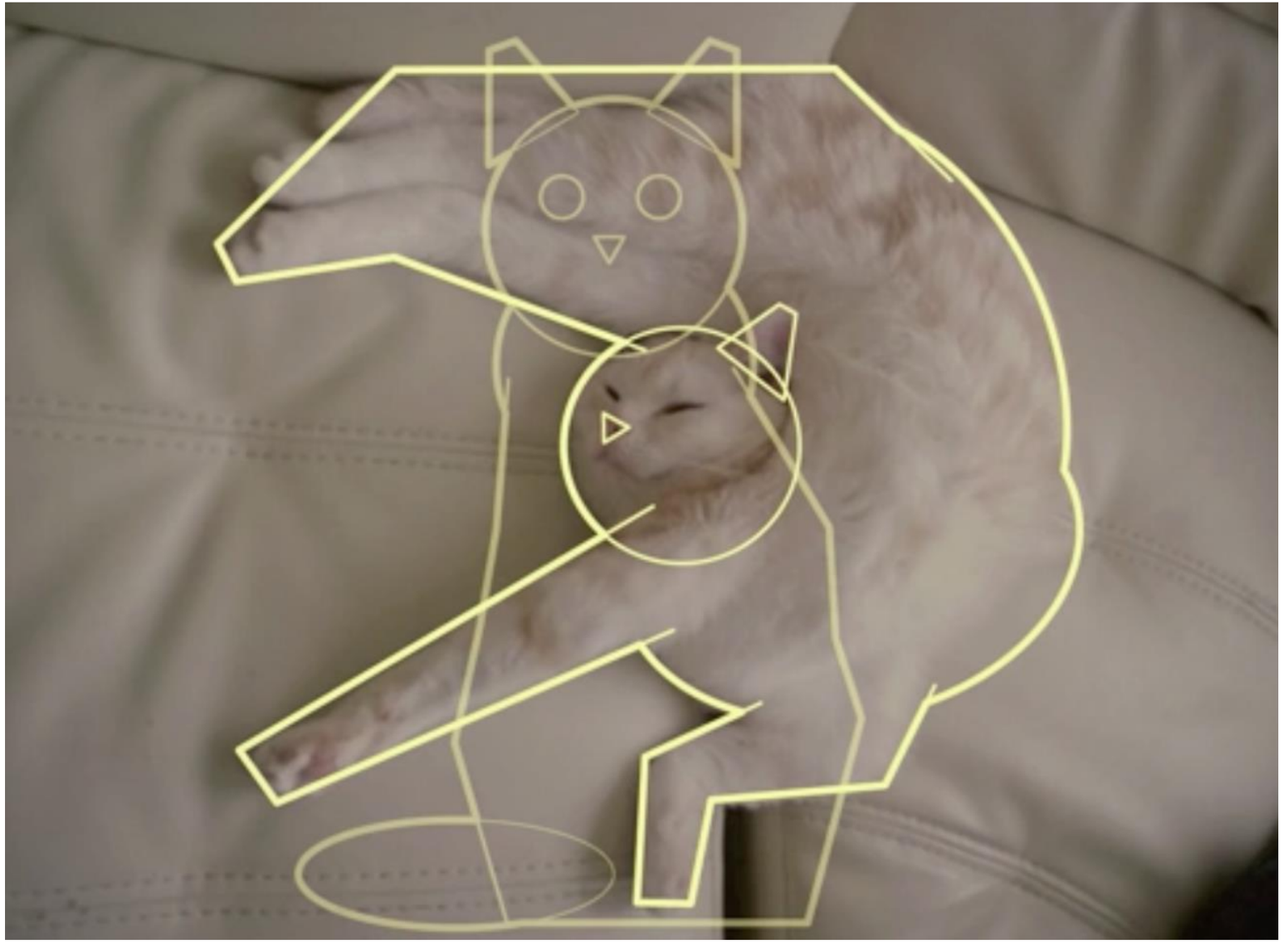
识别猫



识别猫



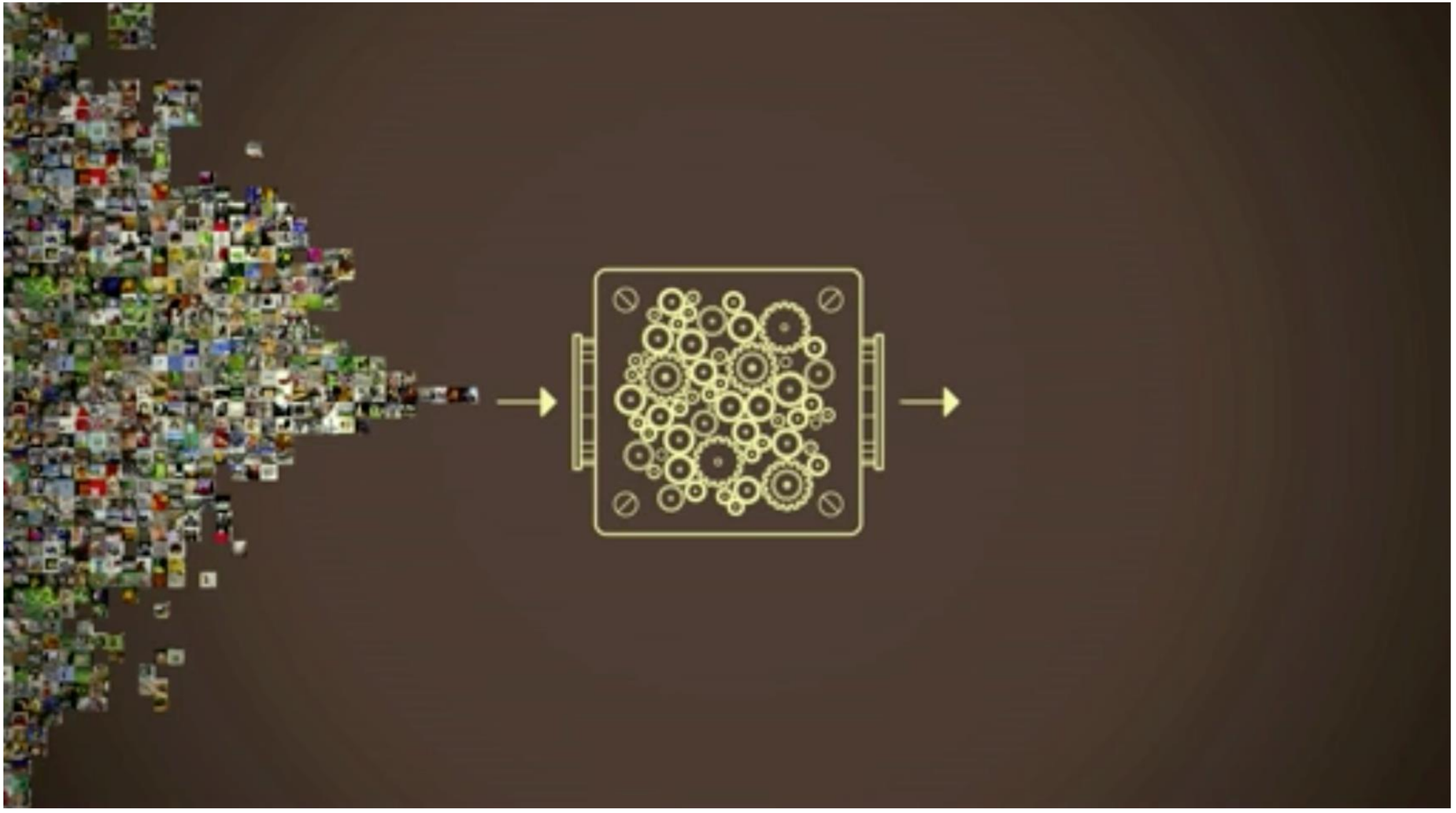
识别猫



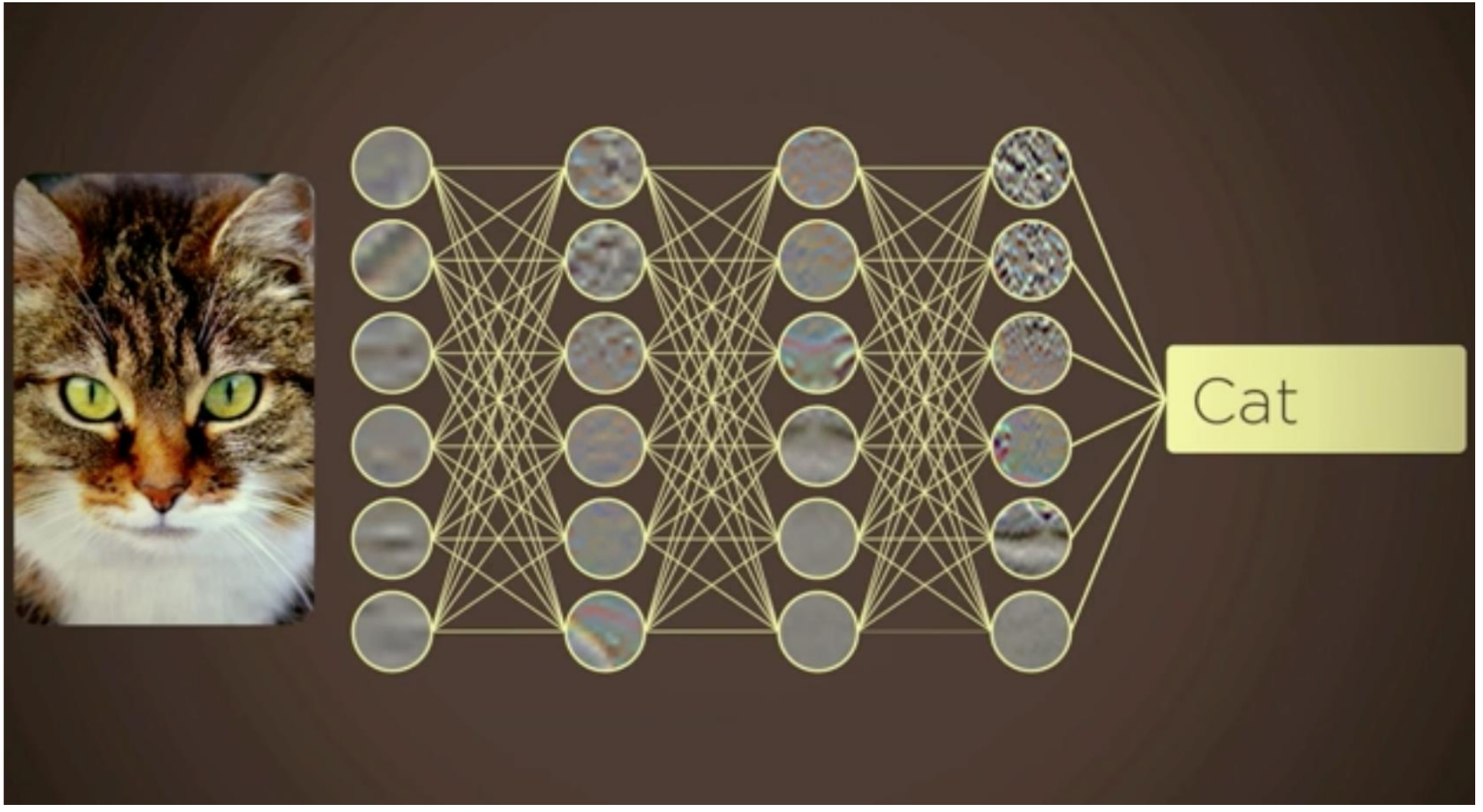
识别猫

识别猫

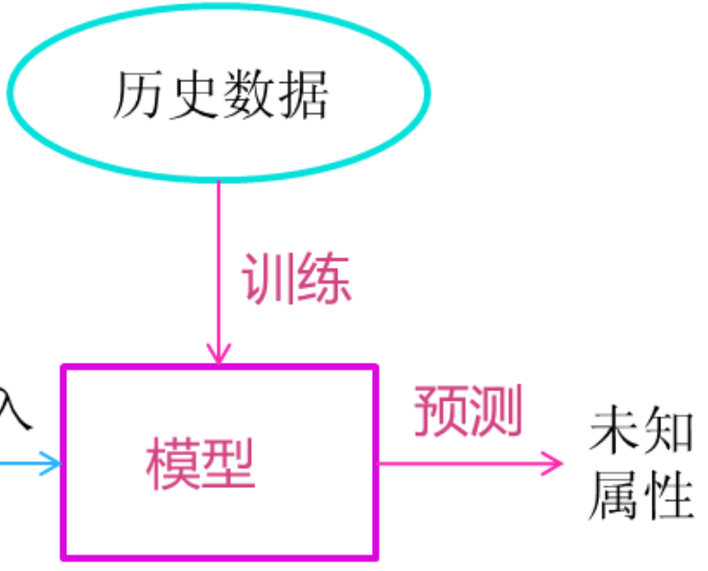
识别猫

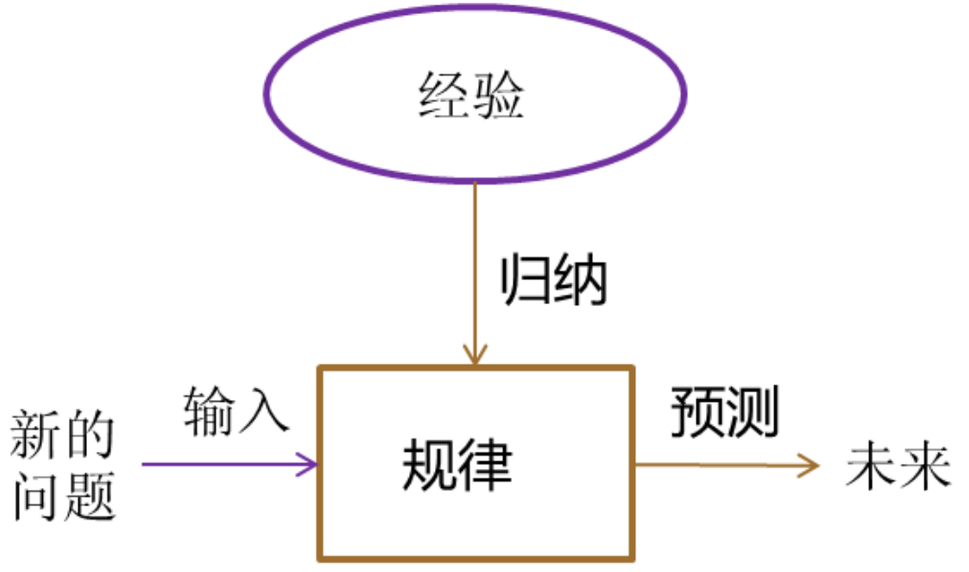


识别猫



机器学习与人类思维



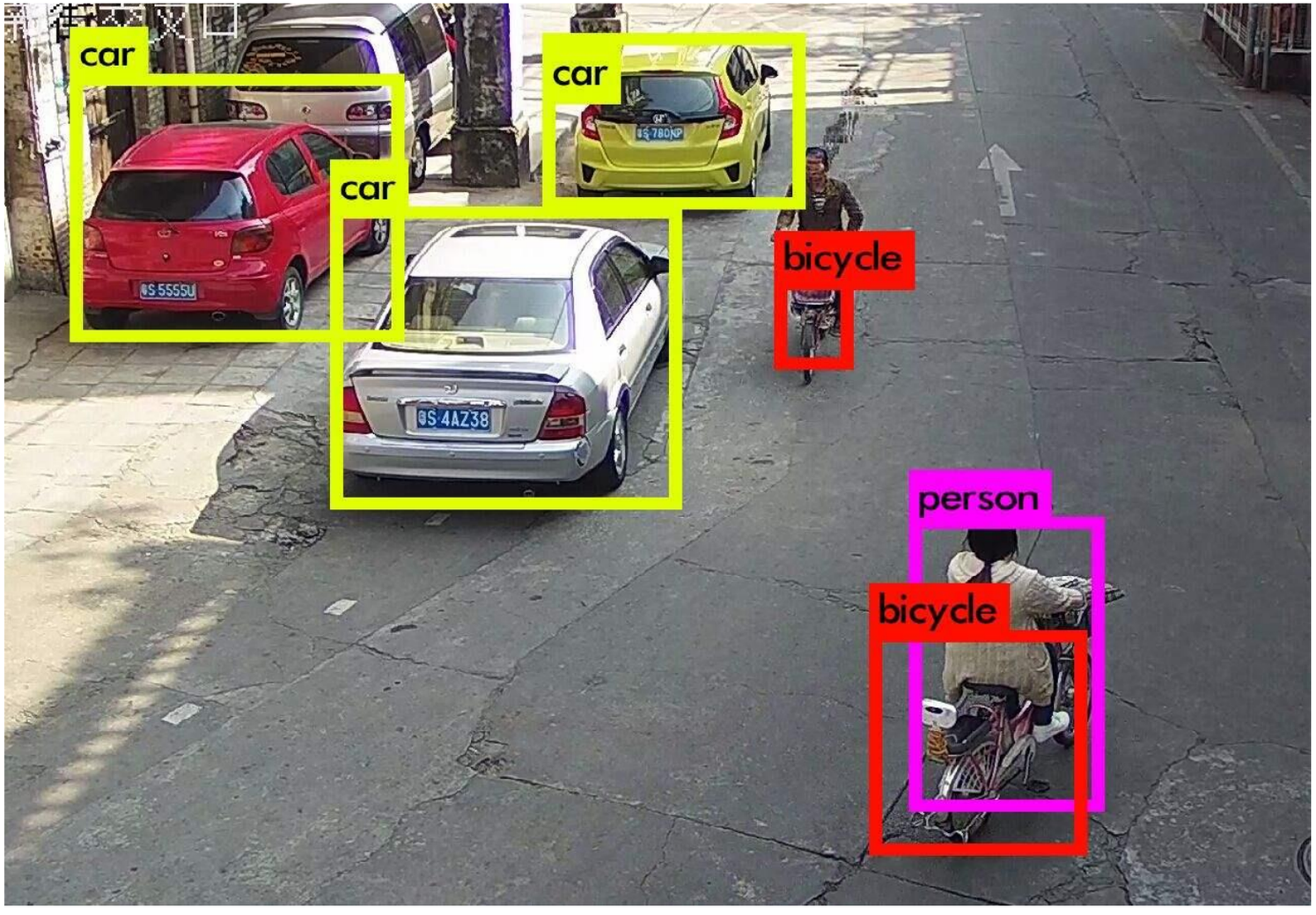


深度学习的应用

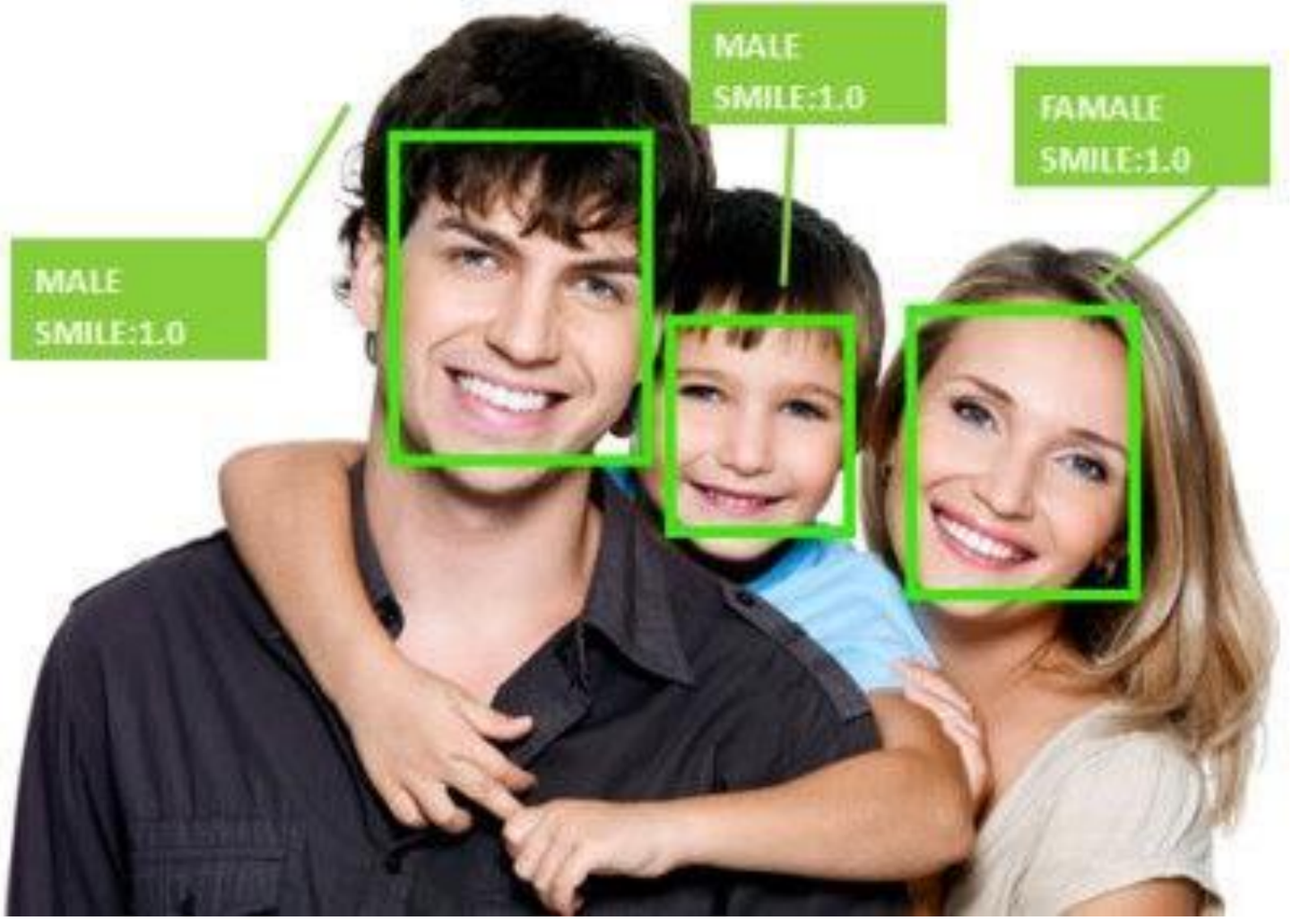
图像识别


mite
container ship
motor scooter
leopard
mite
black widow
cockroach
tick
starfish
container ship
lifeboat
amphibian
fireboat
drilling platform
motor scooter
go-kart
moped
bumper car
golfcart
leopard
jaguar
cheetah
snow leopard
Egyptian cat

目标识别



人脸识别



图片描述

A person on a beach flying a kite.

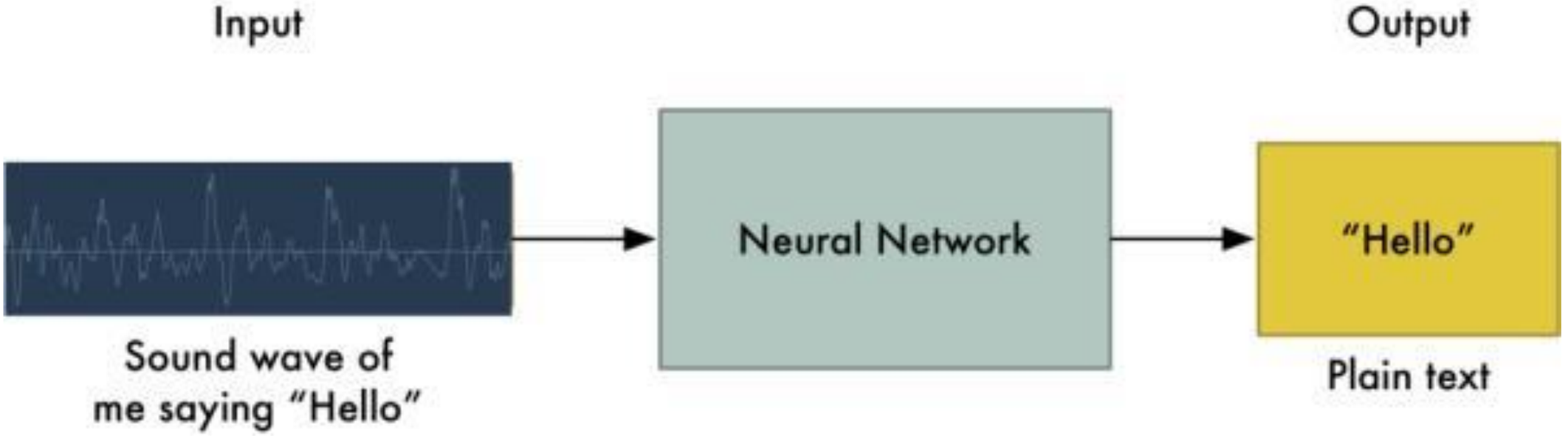
A black and white photo of a train on a train track.

A person skiing down a snow covered slope.

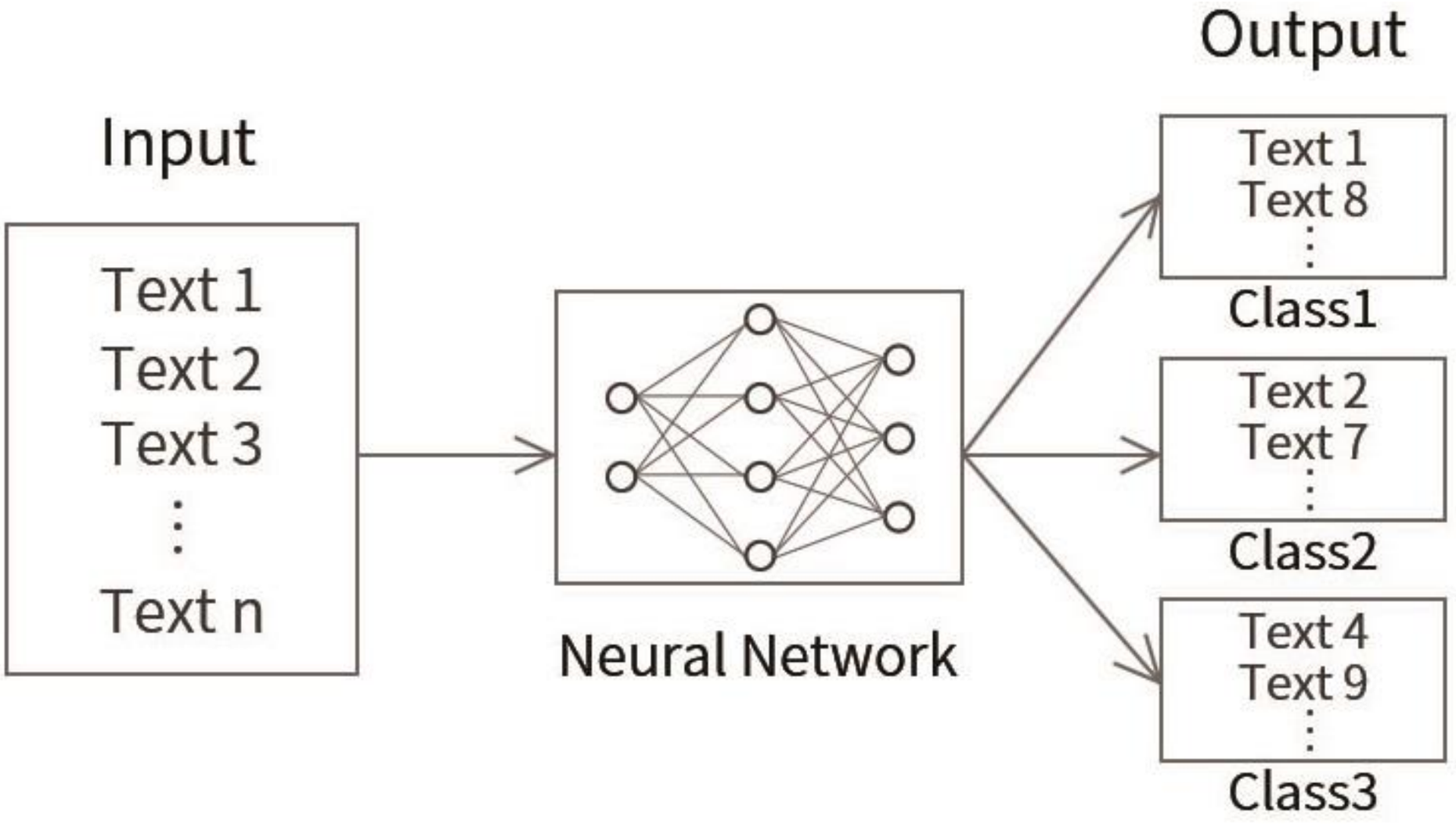
A group of giraffe standing next to each other.

图像风格转换

语音识别


Input
Output
Neural Network
"Hello"
Sound wave of
me saying "Hello"
Plain text

文本分类


Output
Input
Text 1
Text 2
Text 3
⋮
Text n
Text 1
Text 8
⋮
Class1
Text 2
Text 7
⋮
Class2
Text 4
Text 9
⋮
Class3
Neural Network

机器翻译


Google
翻译
关闭即时翻译
英语 中文 德语 检测语言
中文(简体) 英语 日语
翻译
床前明月光,疑是地上霜
11/5000
Bed moonlight, suspected to be frost on the ground
Chuáng qián míng yuèguāng, yí shì dìshàng shuāng
点击图标下载 App
Android
iOS

创作诗歌

可以写诗，下面几首诗，大家来猜猜，哪些是机器写的，哪些是人写的?

千秋明月照幽窗，一夜西风满院凉。山寺钟鸣惊宿鸟，水边芳草自生香。
一枕相思夜未休，春山秋雨惹离愁。凭栏望断江南月，花落无声水自流。
春到江南草更青，胭脂粉黛玉为屏。无端一夜西窗雨，吹落梨花满地庭。
百万兵戈战阵前，楚歌声里起狼烟。旌旗蔽日烽连塞，鼓角惊城血染关。
一夜秋风扫叶开，云边雁阵向南来。清霜渐染梧桐树，满地黄花坡上栽。
梨花落尽柳絮飞，雨打芭蕉入翠微。夜静更深人不寐，江头月下泪沾衣。
雨打芭蕉滴泪痕，残灯孤影对黄昏。夜来无寐听窗外，数声鸡鸣过晓村。
孤舟一叶泊江头，雁去无声送客愁。莫道春来芳草绿，人间万里尽风流。
客梦初醒惊夜雨，西窗帘外月如钩。梧桐落叶知秋意，一任相思到白头。
秋深更觉少人行，雁去无声月满庭。兄弟别离肠断处，江南烟雨总关情。
明月当窗照夜空，桂花香透小楼东。金风玉露三更后，雪落梅梢一点红。
琴静云水清，　夕阳照天明。　一曲相思调，　肠断心不宁。
楼头一夜风，　烟雨锁朦胧。　江上千帆过，　枝头黄叶红。

图像生成：make.girls.moe

Generate

+1 -1

Share on Twitter

Options

Hair Color: Black

Hair Style: Long Hair

Eye Color: Random

Blush: Off | Random | On

Smile: Off | Random | On

Open Mouth: Off | Random | On

Hat: Off | Random | On

Ribbon: Off | Random | On

Glasses: Off | Random | On

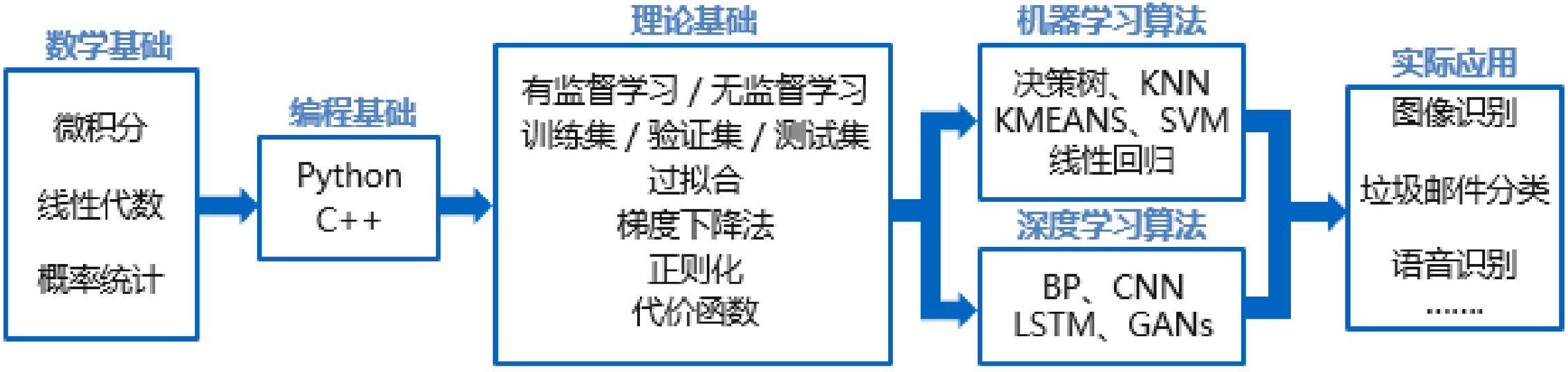
Noise: Random | Fixed

Current Noise

Noise Import/Export: Import | Export

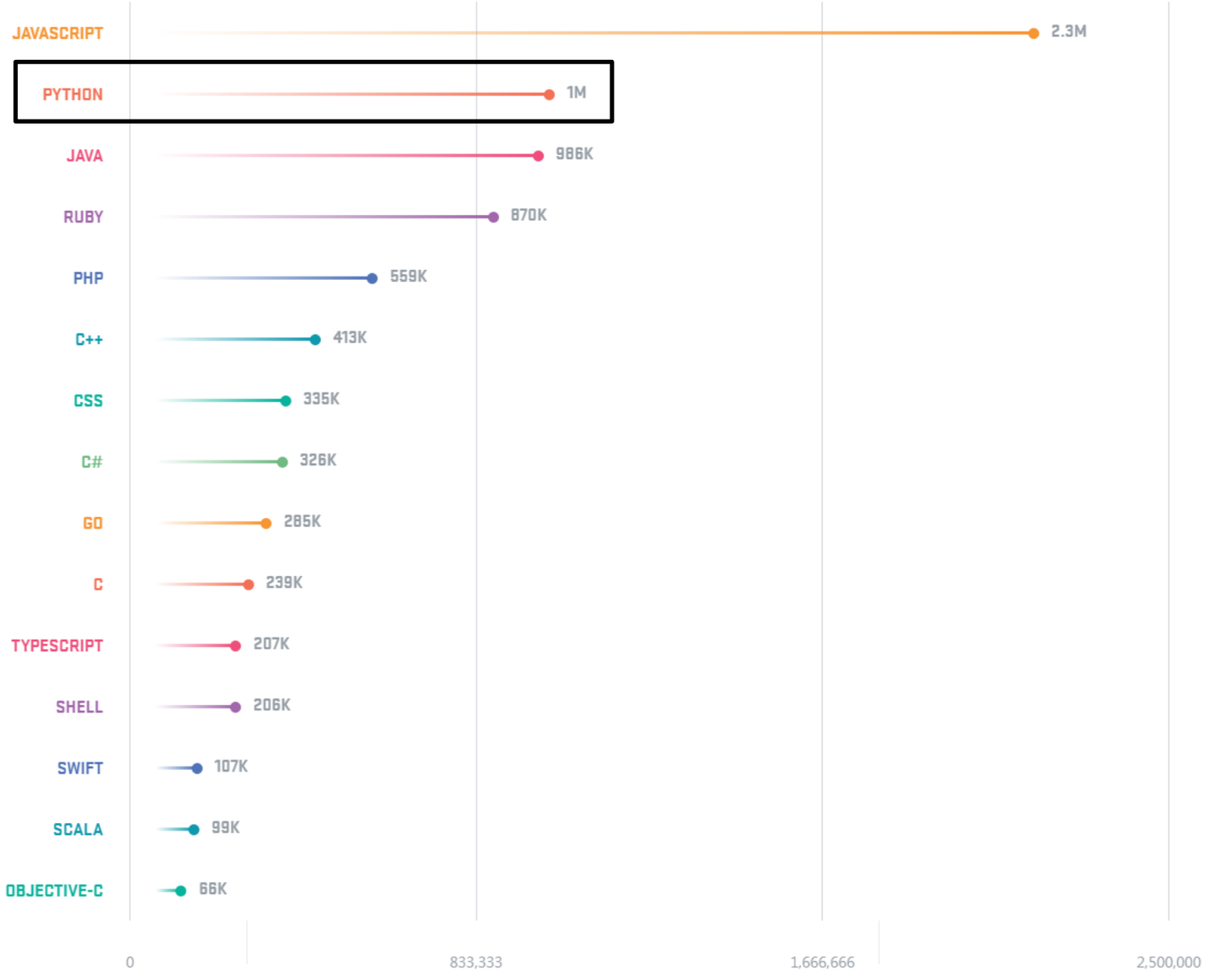
如何学习Deep Learning

知识框架


数学基础
微积分
线性代数
概率统计
编程基础
Python
C++
理论基础
有监督学习 / 无监督学习
训练集 / 验证集 / 测试集
过拟合
梯度下降法
正则化
代价函数
机器学习算法
决策树、KNN
KMEANS、SVM
线性回归
深度学习算法
BP、CNN
LSTM、GANs
实际应用
图像识别
垃圾邮件分类
语音识别
.....

Python/Numpy/Matplotlib

2017GitHub上最受欢迎的前15 门语言



优点用途

优点：功能强大，开发效率高，应用广泛

用途：数据分析
科学计算
机器学习
深度学习
可视化界面
网页开发
网络爬虫
脚本

Anaconda安装

https://www.continuum.io/downloads

如果没有jupyter_notebook_config.py文件
打开命令提示符执行:jupyter notebook --generate-config

程序

Python

Numpy

Matplotlib

机器学习

从数据到智慧

" Information is not knowledge.

Knowledge is not wisdom.

Wisdom is not truth.

Truth is not beauty.

Beauty is not love.

Love is not music.

Music is THE BEST."

——Frank Vincent Zappa (1940 –1993) was an American composer, electric guitarist, record producer, and film director.

什么是数据挖掘？

1.周杰伦是男歌手吗？

2.吸烟是不是肺癌发病的主要诱因？

训练数据/验证数据/测试数据

- 建模之前，我们可以把数据分成三部分。

  训练集(Training data)

  验证集(Validation data)

  测试集(Test data)

- 训练集还是用来训练，构建模型。
- 验证集是用来在模型训练阶段测试模型的好坏。
- 等模型训练好之后，再用测试集来评估模型的好坏。

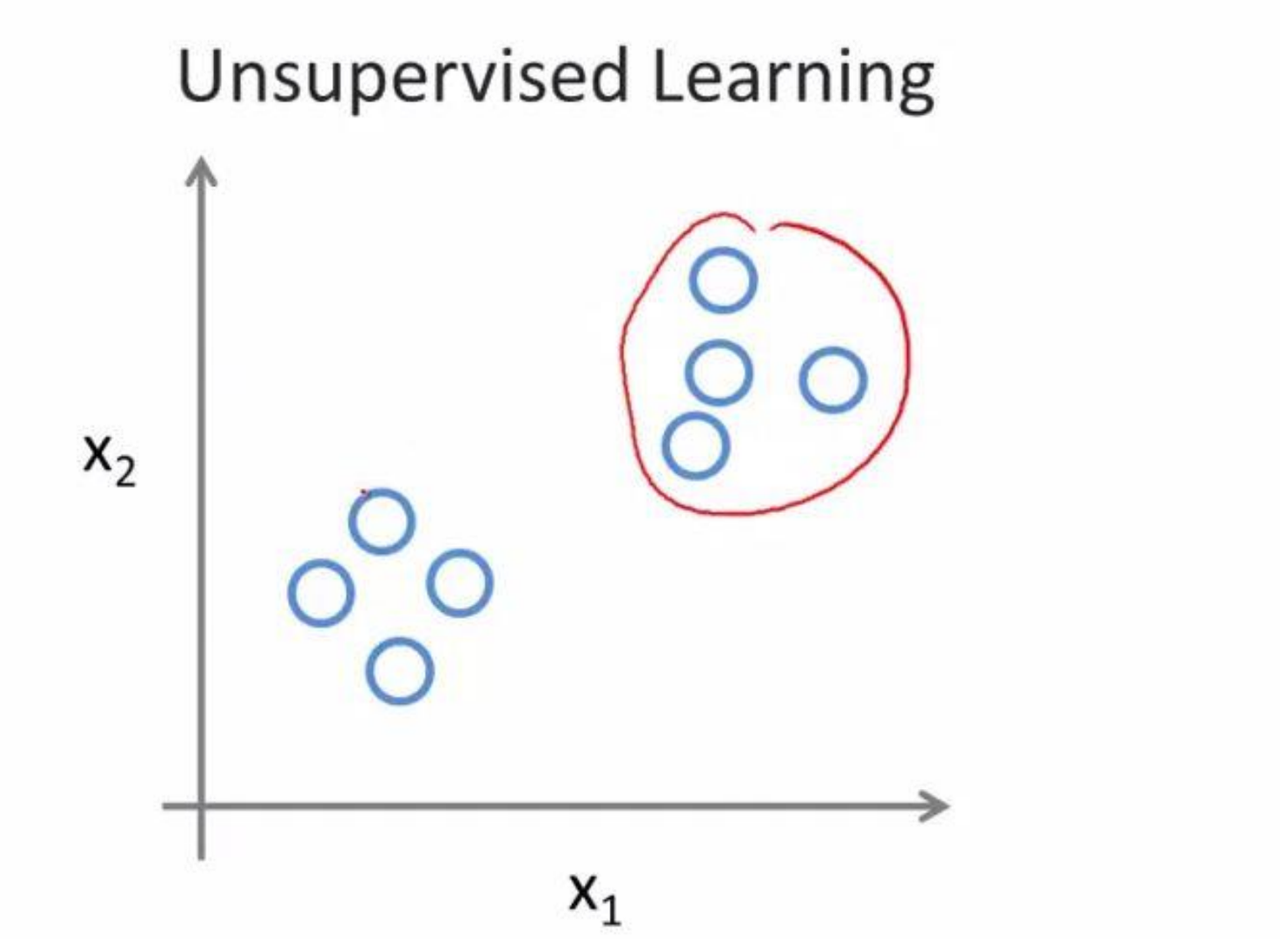
学习方式

监督学习

3

dog

无监督学习

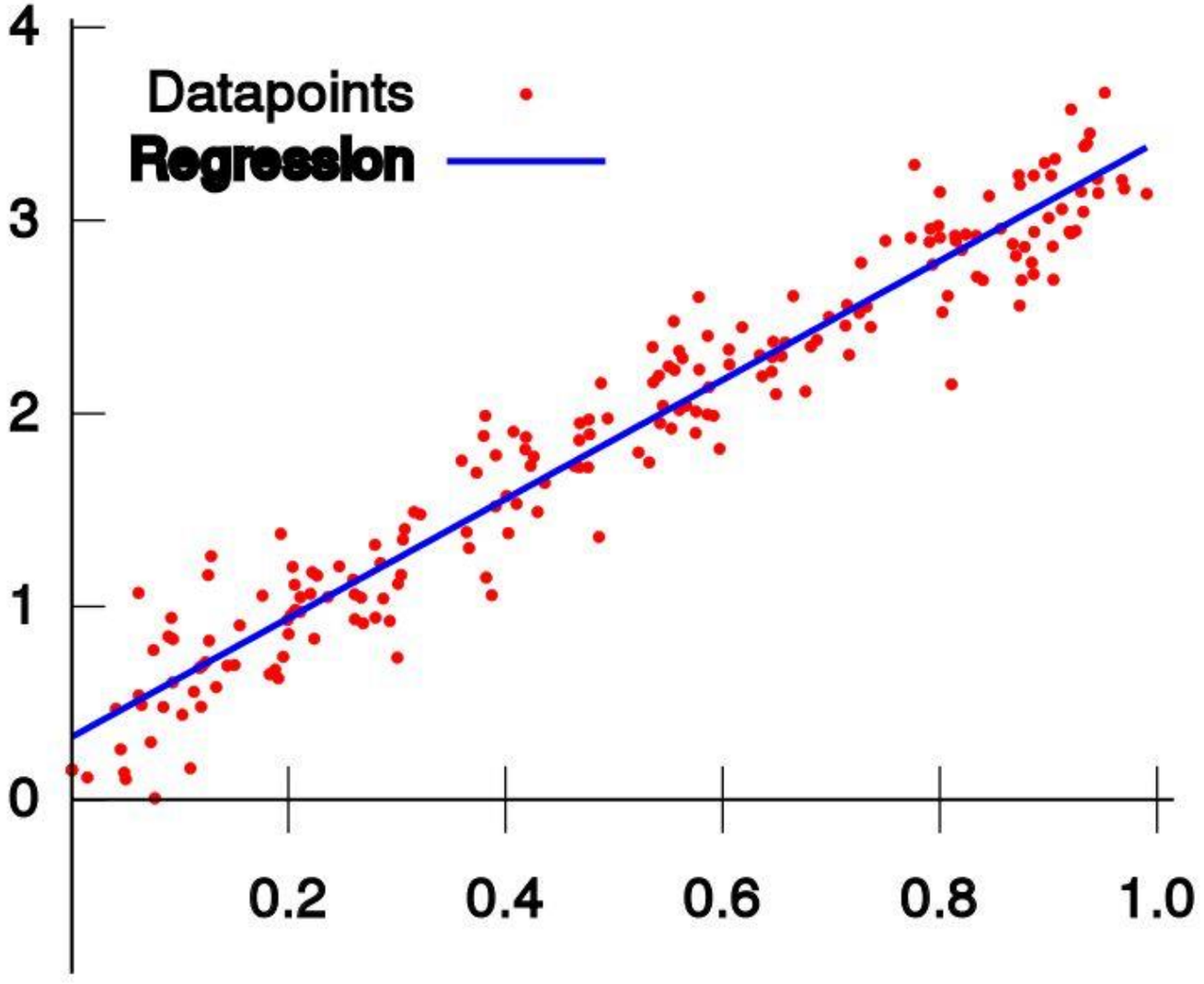


半监督学习

半监督学习是监督学习和无监督学习相结合的一种学习方式。主要是用来解决使用少量带标签的数据和大量没有标签的数据进行训练和分类的问题。

常见应用

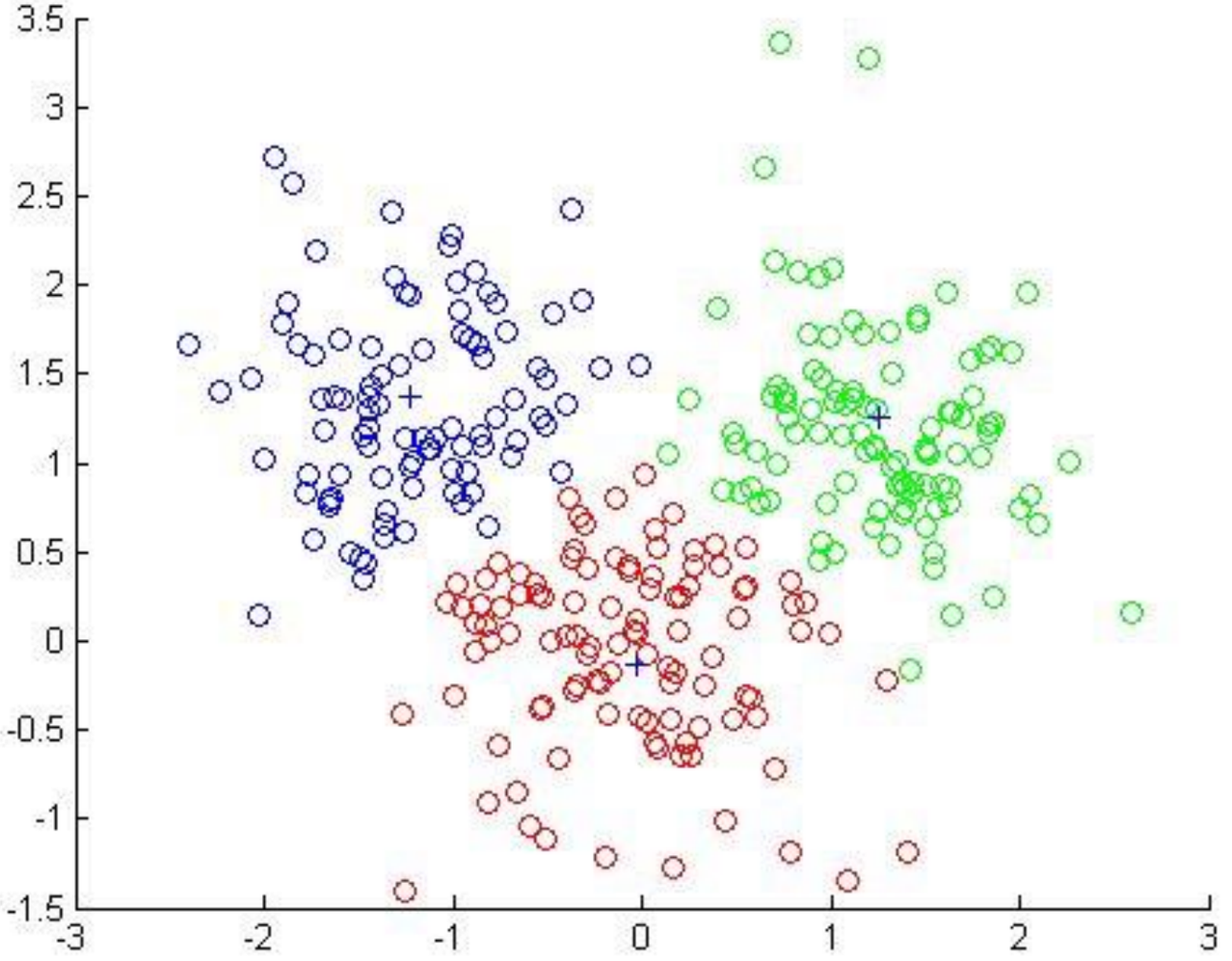
回归



分类

- 图像识别
- 垃圾邮件分类
- 文本分类
- ……

聚类



简单回归例子

样本	面积(平方米)	学区	房价(万)
1	100	8	100
2	120	9	130
3	60	6	80
4	95	5	85

拿到新的房子面积和学区编号，预测房价

简单分类例子

样本	天气	温度	湿度	风力	周末	是否运动
1	晴	暖	普通	强	是	是
2	晴	暖	高	弱	否	是
3	雨	冷	高	强	是	否
4	晴	暖	高	弱	否	是

天气：晴，阴，雨
温度：暖，冷
湿度：普通，大
风力：强，弱
周末：是，否
预测是否运动：是，否

简单聚类例子

样本	购买次数	购买总金额(万)	浏览次数
1	10	5	50
2	1	0.5	5
3	0	0	15
4	1	0.1	2

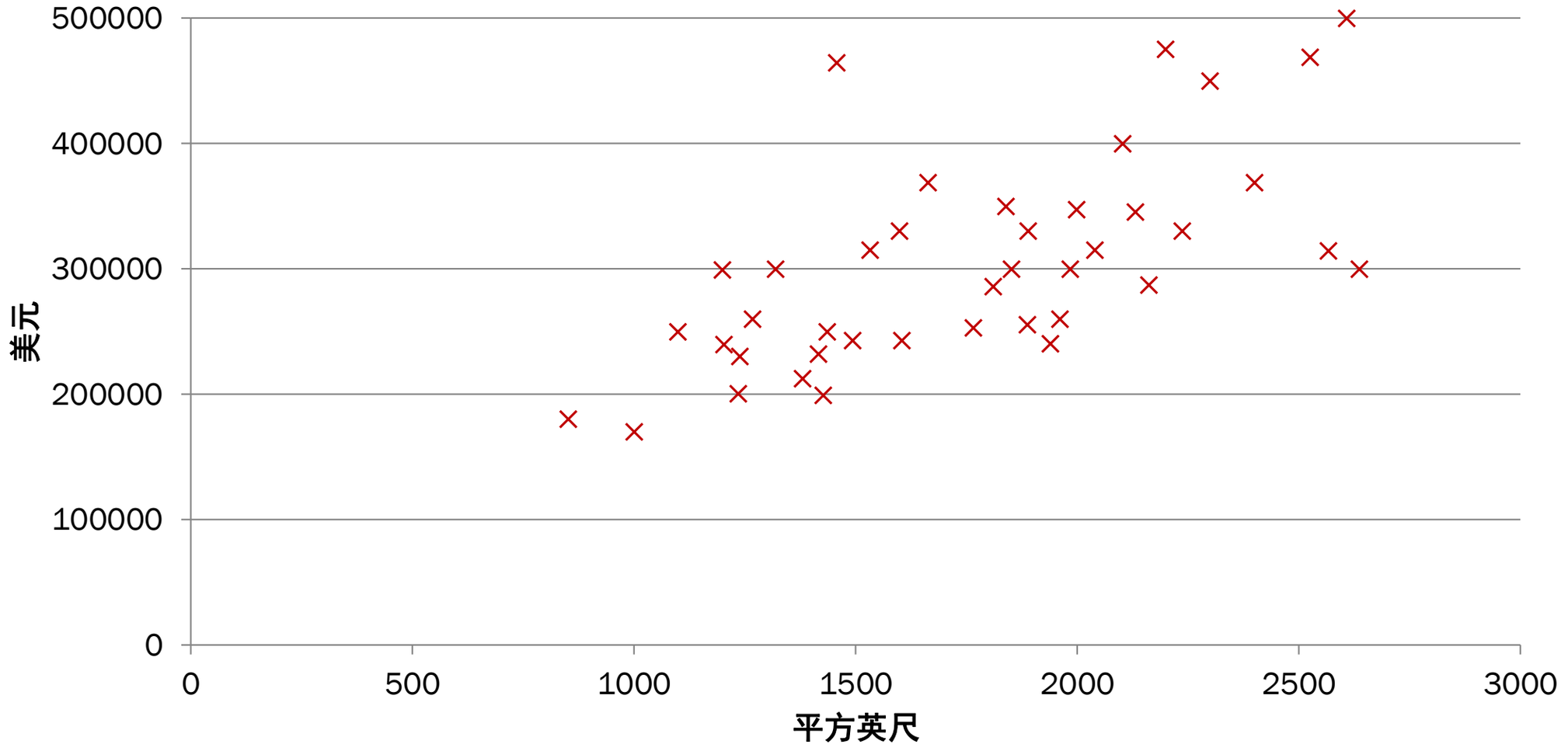
根据用户数据给用户分类，分类数量可以视情况而定

回归/分类/聚类

回归：预测数据为连续型数值。
分类：预测数据为类别型数据，并且类别已知。
聚类：预测数据为类别型数据，但是类别未知。

回归分析

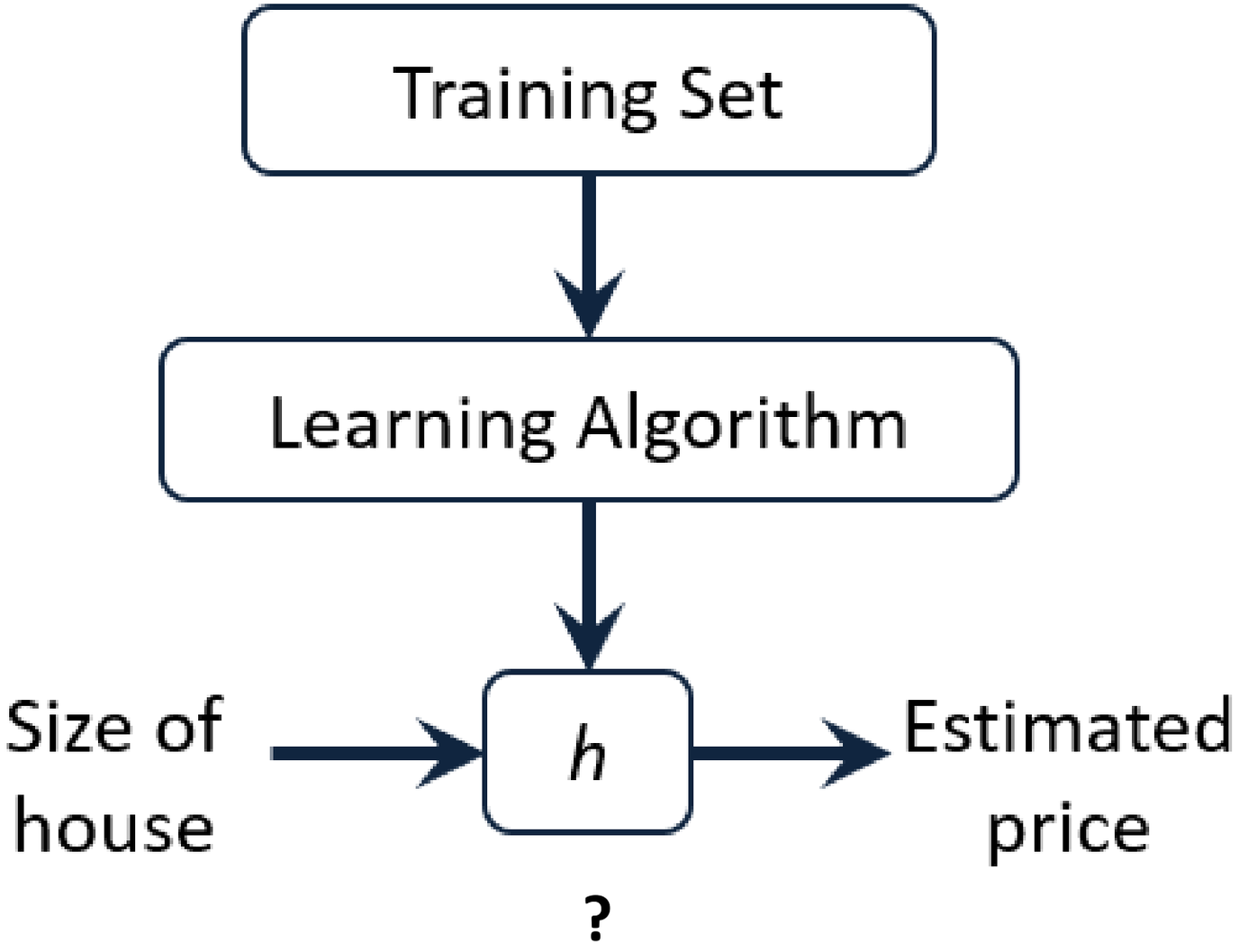
房价预测


500000
400000
300000
200000
100000
0
美元
0
500
1000
1500
2000
2500
3000
平方英尺

房价预测

Size in feet^2 (x)	Price ($) in 1000's (y)
2104	460
1416	232
1534	315
852	178
...	...
特征值(feature)	标签/结果(target)

房价预测


Training Set
Learning Algorithm
Size of house
h
Estimated price
?

一元线性回归

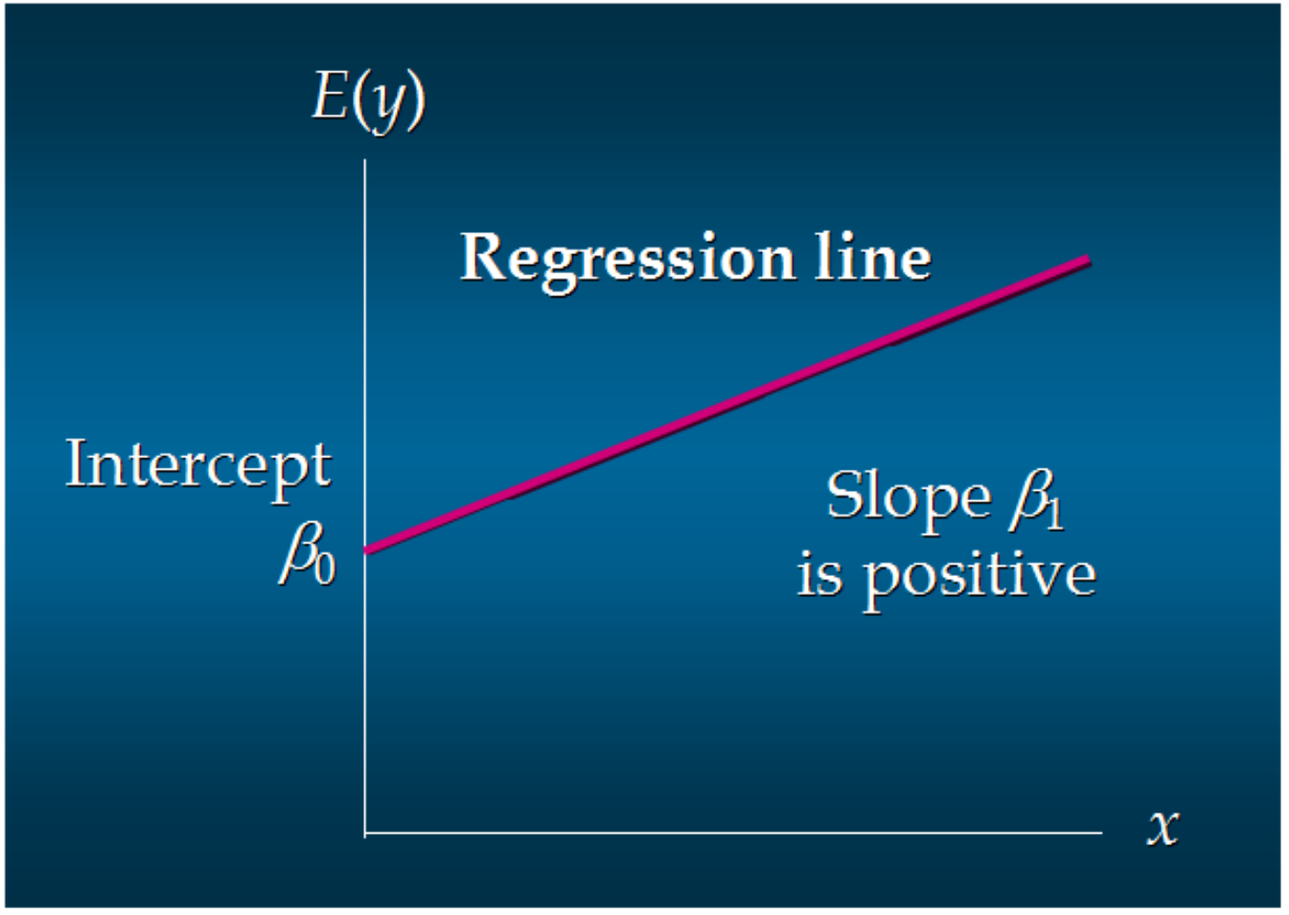
- 回归分析(regression analysis)用来建立方程模拟两个或者多个变量之间如何关联
- 被预测的变量叫做：因变量(dependent variable), 输出(output)
- 被用来进行预测的变量叫做： 自变量(independent variable), 输入(input)
- 一元线性回归包含一个自变量和一个因变量
- 以上两个变量的关系用一条直线来模拟
- 如果包含两个以上的自变量，则称作多元回归分析(multiple regression)

一元线性回归

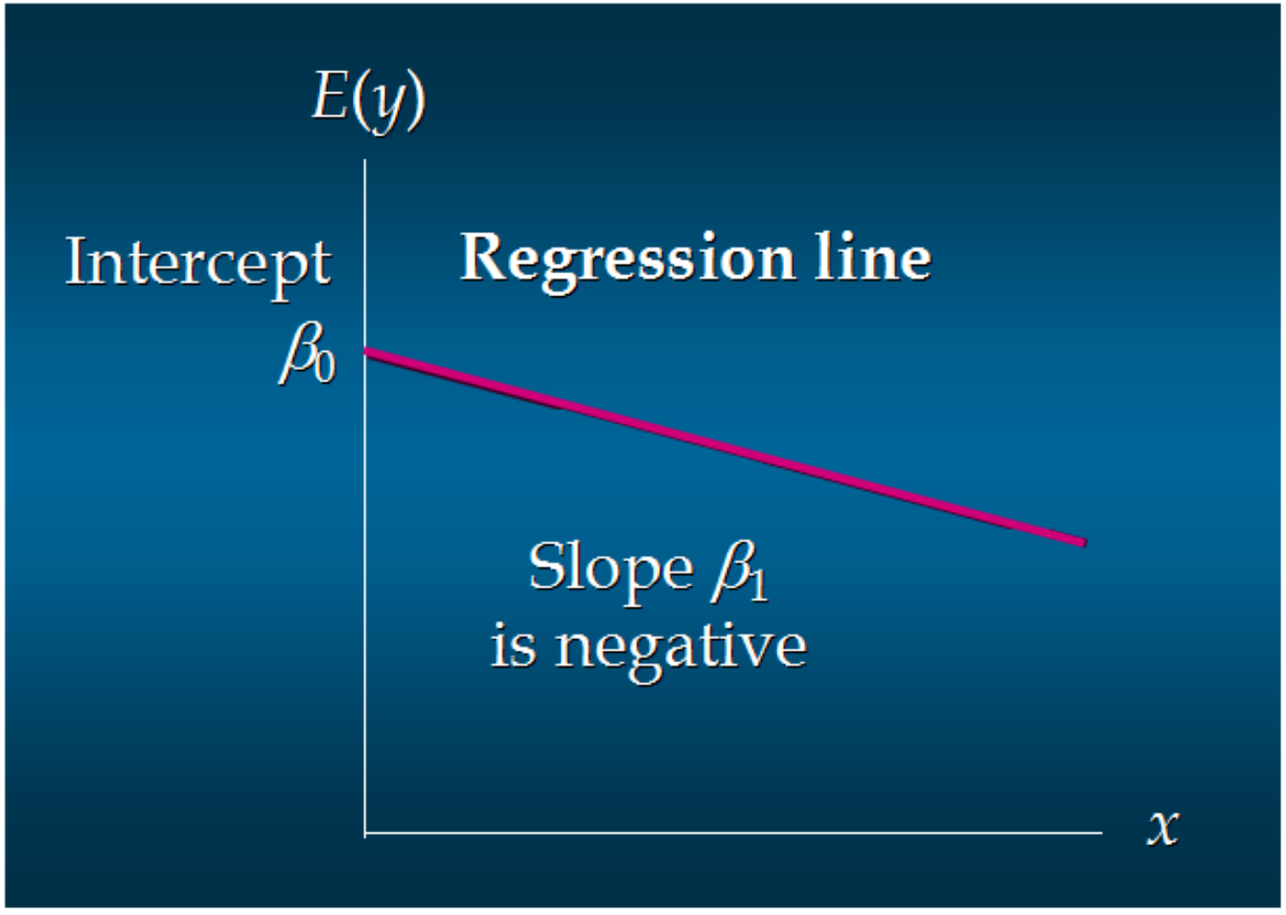
$$h_\theta(x) = \theta_0 + \theta_1 x$$

这个方程对应的图像是一条直线，称作回归线。其中，θ_1为回归线的斜率， θ_0为回归线的截距。

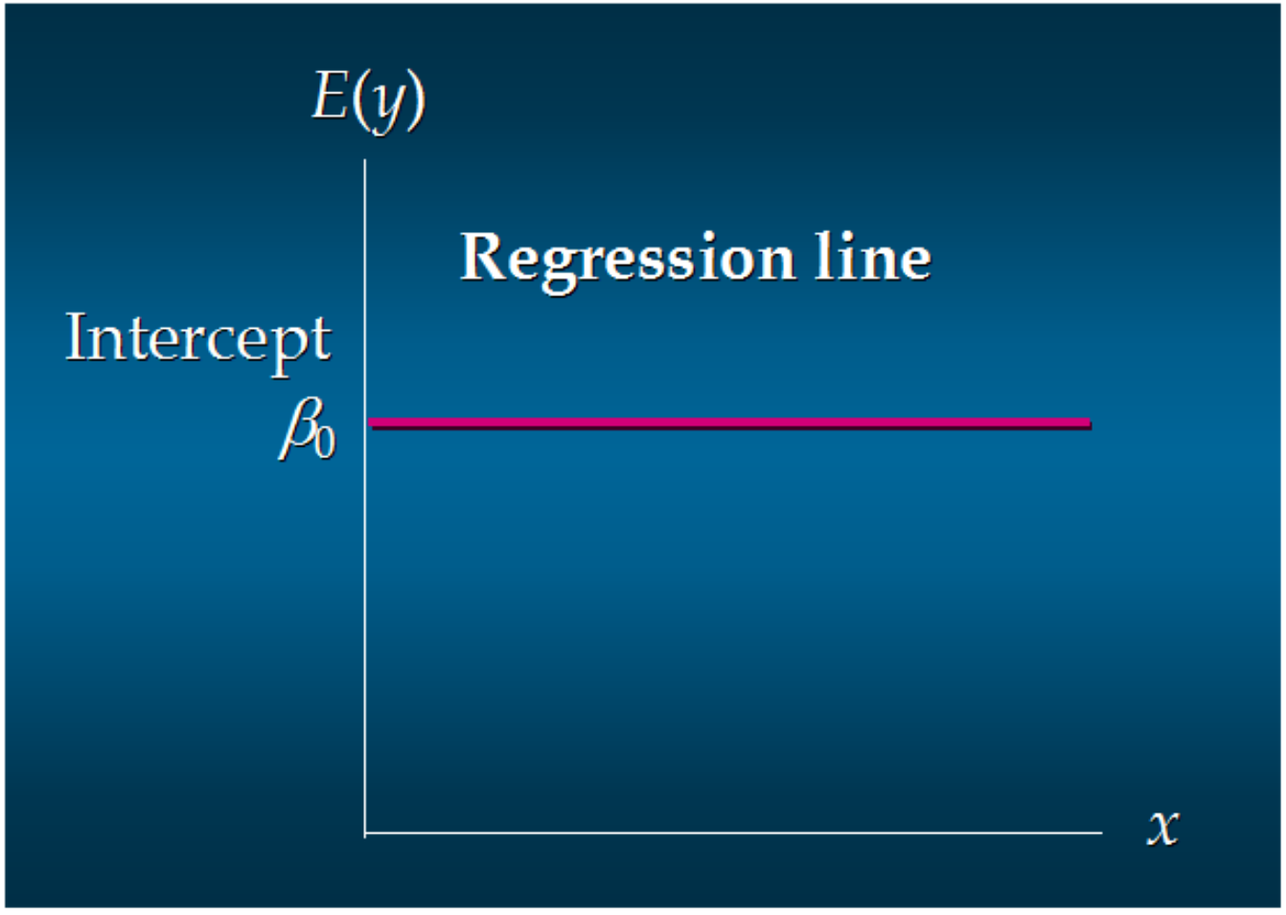
一元线性回归-正相关


E(y)
Regression line
Intercept
β_0
Slope β_1
is positive
x

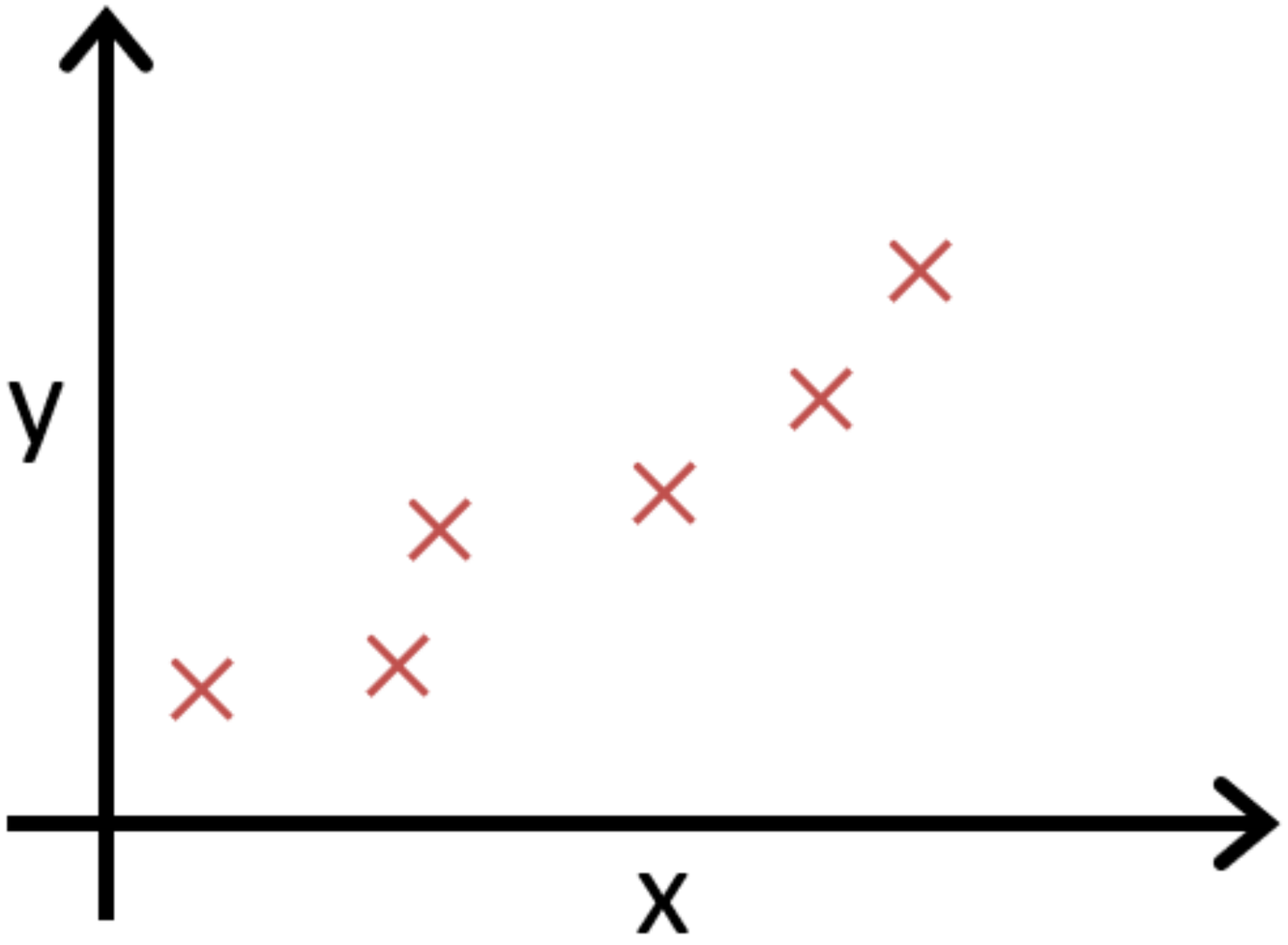
一元线性回归-负相关



一元线性回归-不相关



求解方程系数



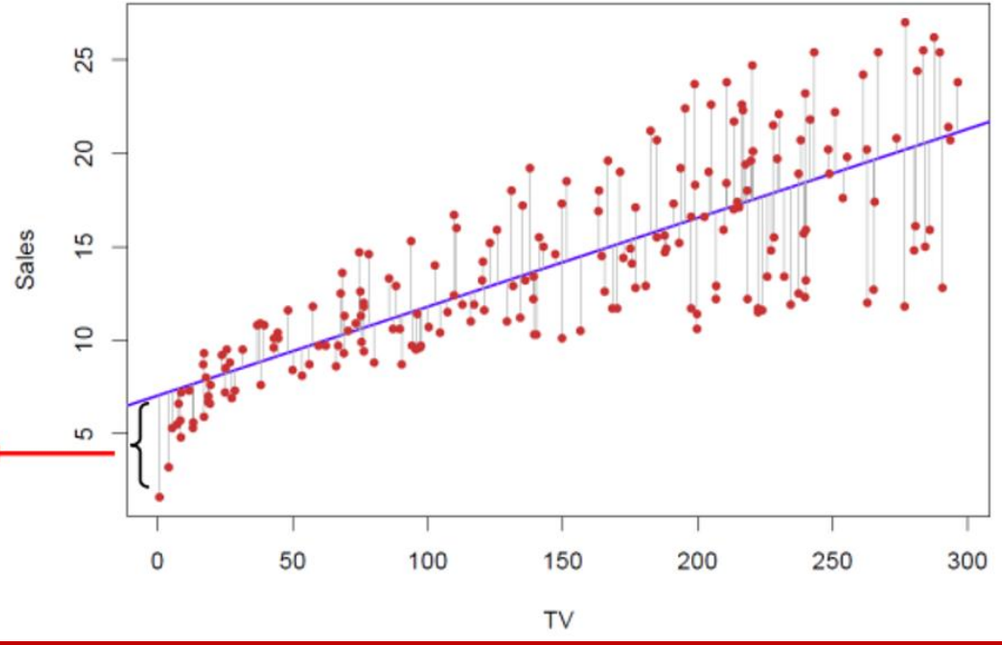
代价函数

代价函数(Cost Function)

- 最小二乘法
- 真实值y，预测值$h_\theta(x)$，则误差平方为$\left(\mathrm{y}-h_\theta(x)\right)^2$
- 找到合适的参数，使得误差平方和：

$$J(\theta_0,\theta_1)=\frac{1}{2m}\sum_{i=1}^{m}\left(y^i-h_\theta\left(x^i\right)\right)^2\text{最小}$$

$\mathrm{y}^{\mathrm{i}}-h_\theta\left(\mathrm{x}^{\mathrm{i}}\right)$



梯度下降法

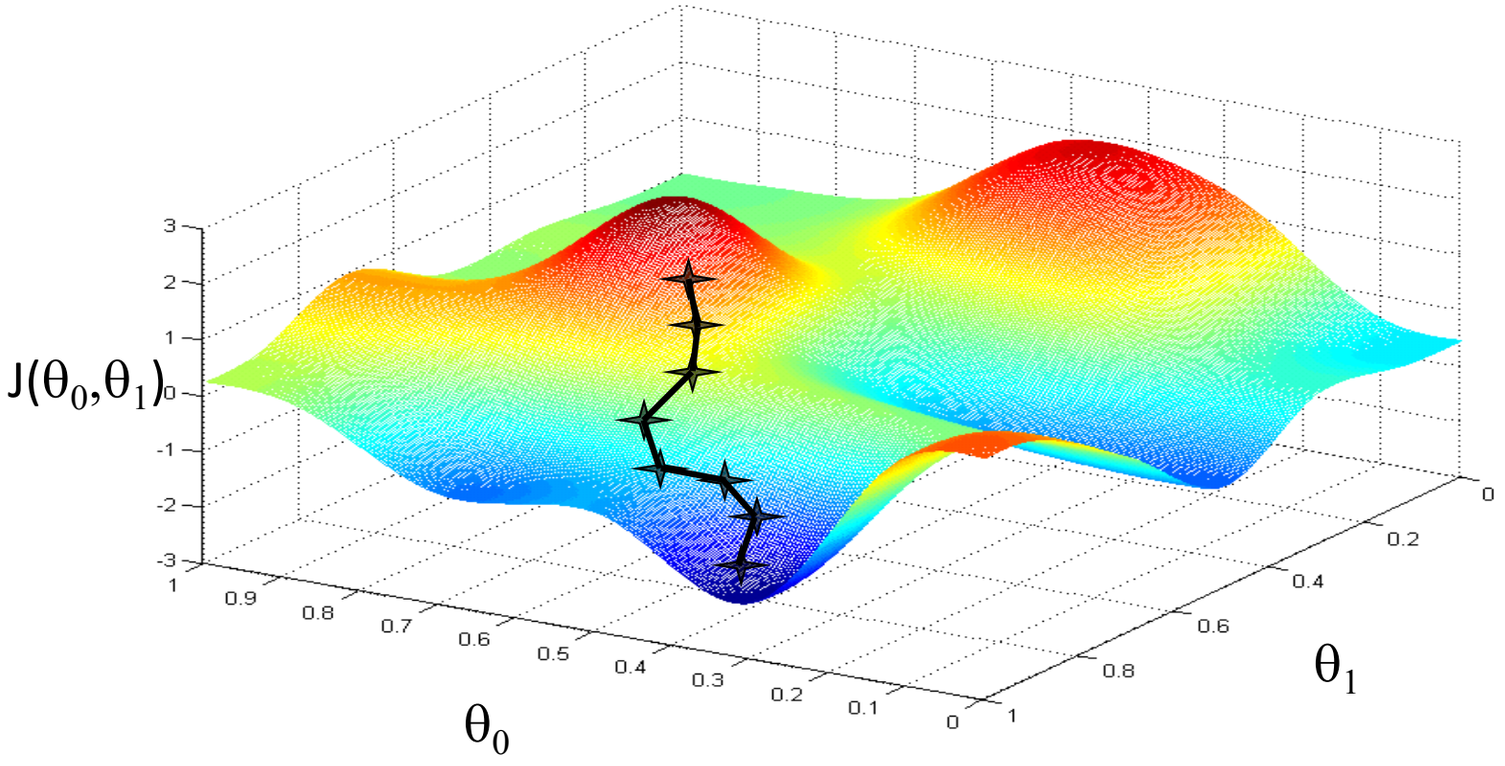
梯度下降法

Have some function $J(\theta_0, \theta_1)$

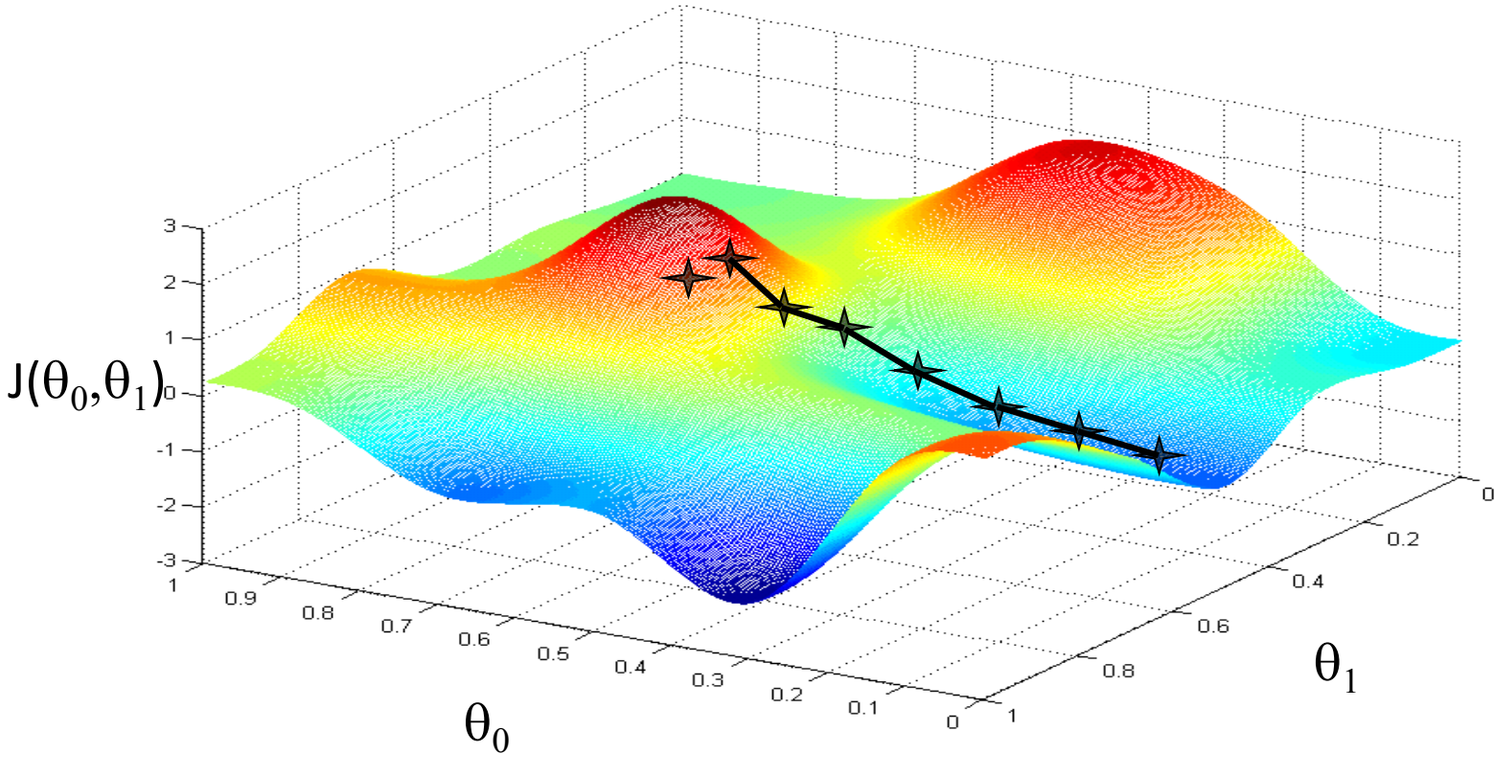
Want $\min_{\theta_0, \theta_1} J(\theta_0, \theta_1)$

- 初始化θ_0, θ_1
- 不断改变θ_0, θ_1，直到$J(\theta_0, \theta_1)$到达一个全局最小值，或局部极小值。

梯度下降法



梯度下降法


$J(\theta_0,\theta_1)$
3
2
1
0
-1
-2
-3
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
θ_0
0
0.2
0.4
0.6
0.8
1
θ_1

梯度下降法

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

学习率

正确做法：同步更新

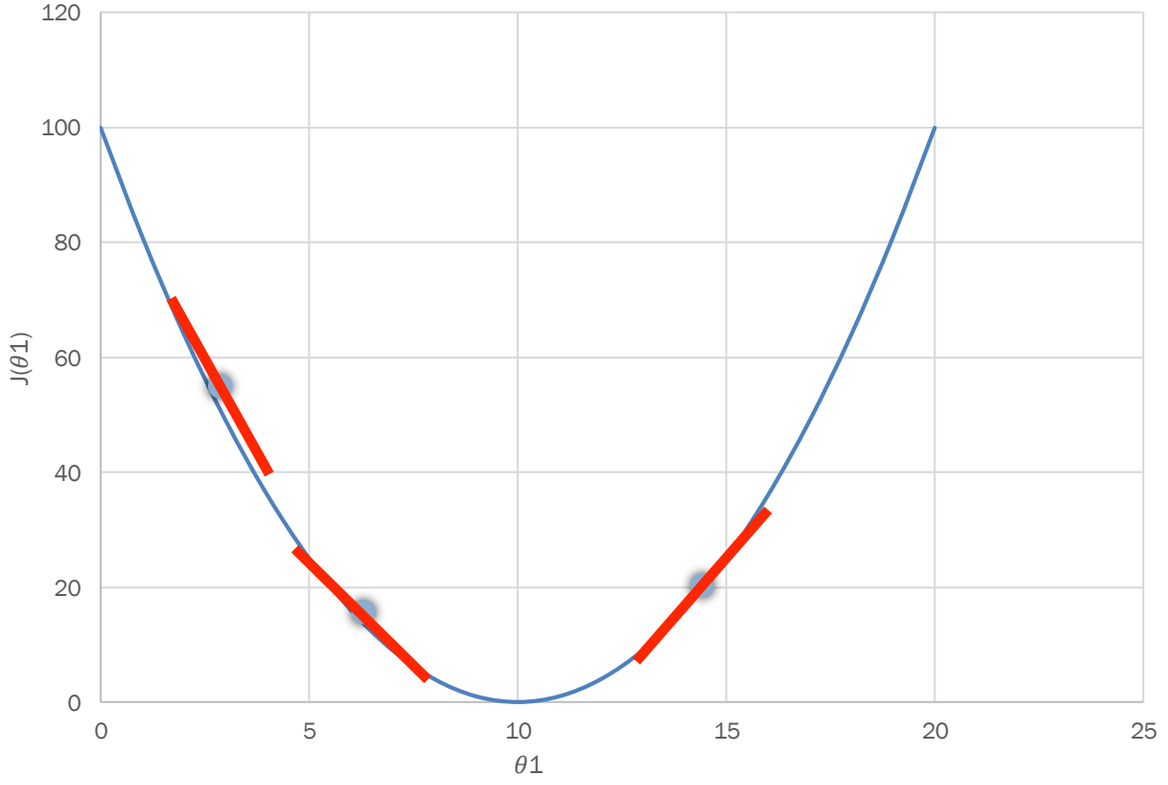
$$\text{temp0} := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$
$$\text{temp1} := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$
$$\theta_0 := \text{temp0}$$
$$\theta_1 := \text{temp1}$$

不正确做法

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

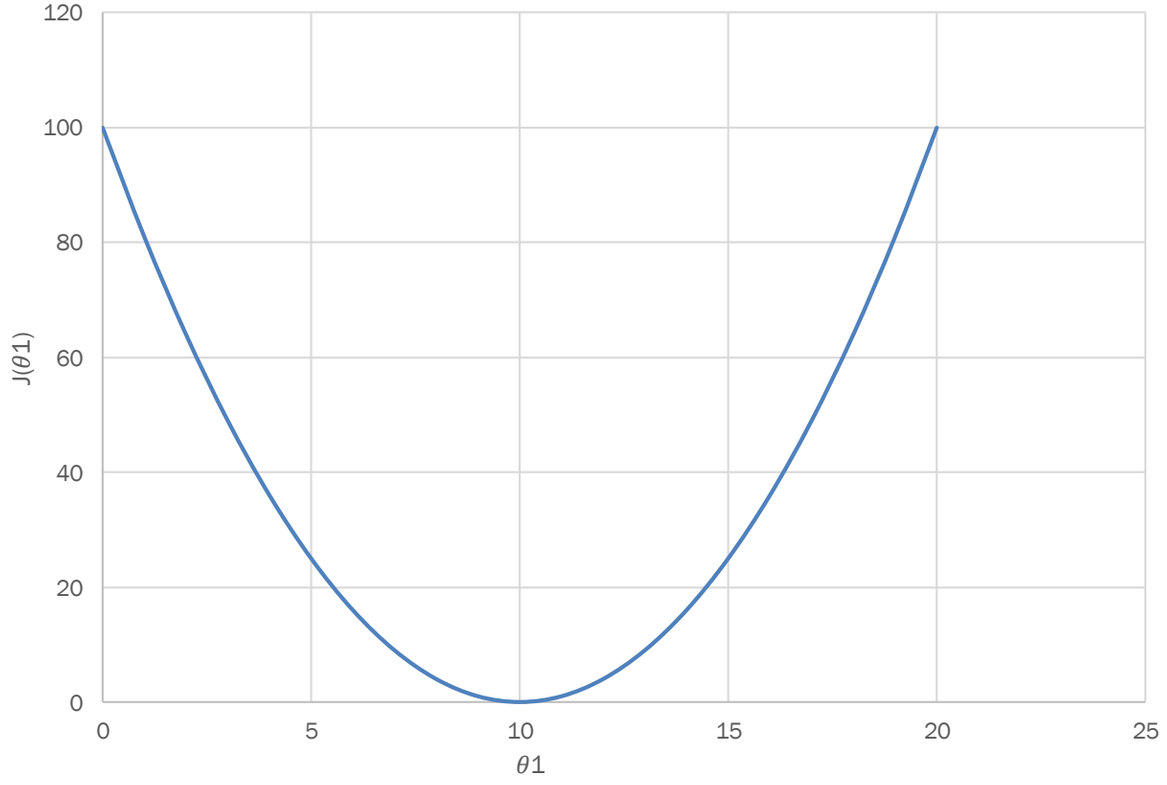
梯度下降法

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



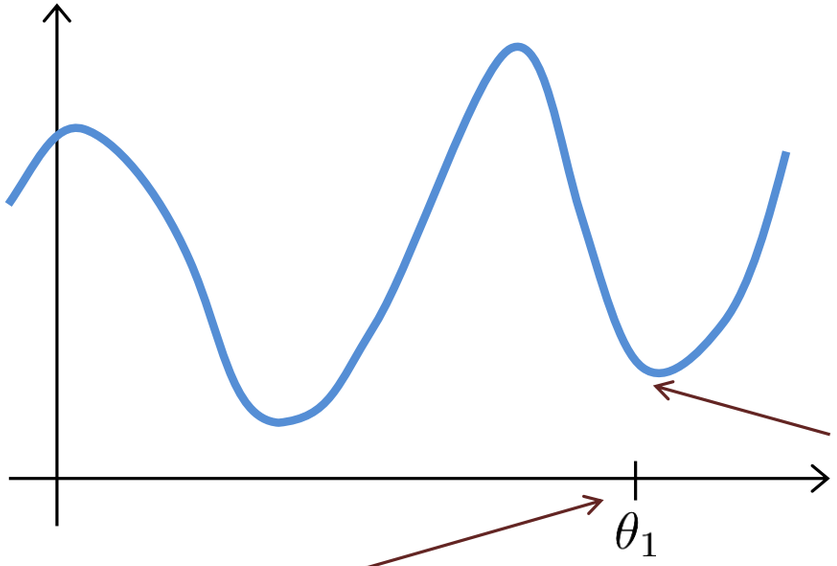
梯度下降法

学习率不能太小，也不能太大，可以多尝试一些值
0.1,0.03,0.01,0.003,0.001,0.0003,0.0001...



梯度下降法

有可能会陷入局部极小值



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

用梯度下降法来求解线性回归

梯度下降法

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

线性回归的模型和代价函数

$$h_\theta(x) = \theta_0 + \theta_1 x$$

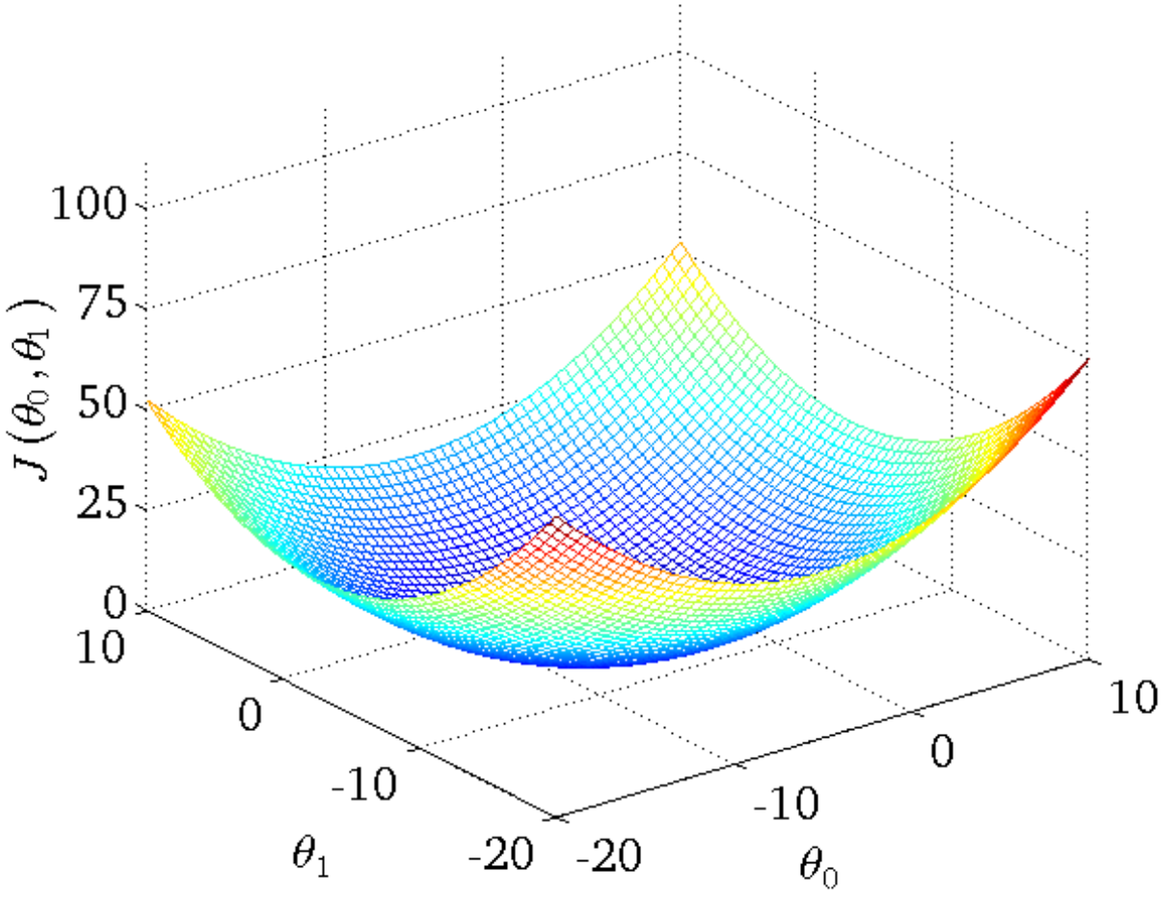
$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$$

用梯度下降法来求解线性回归

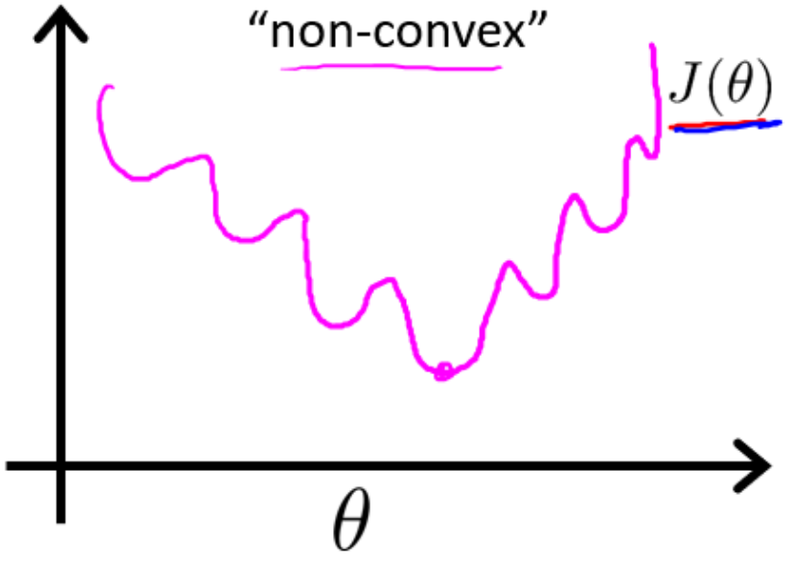
$$\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) = \begin{cases} j = 0 : \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right) \\ j = 1 : \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right) \cdot x^{(i)} \end{cases}$$

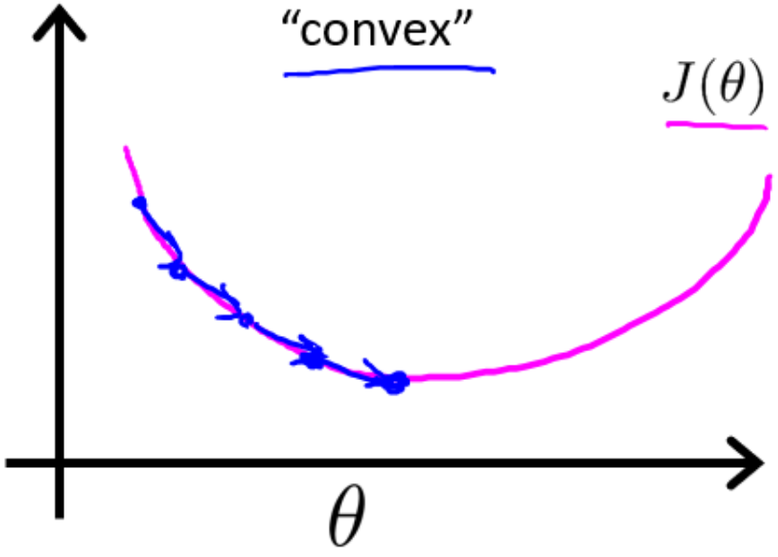
$$\begin{aligned} &\text{repeat until convergence } \{ \\ &\quad \theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right) \\ &\quad \theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right) \cdot x^{(i)} \\ &\} \end{aligned}$$

线性回归的代价函数是凸函数



非凸函数和凸函数

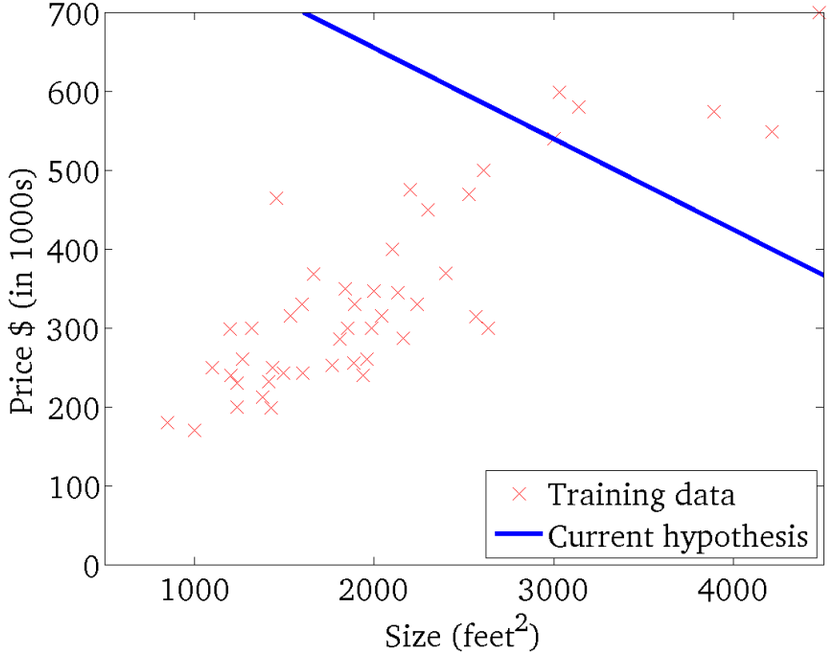




梯度下降法优化过程

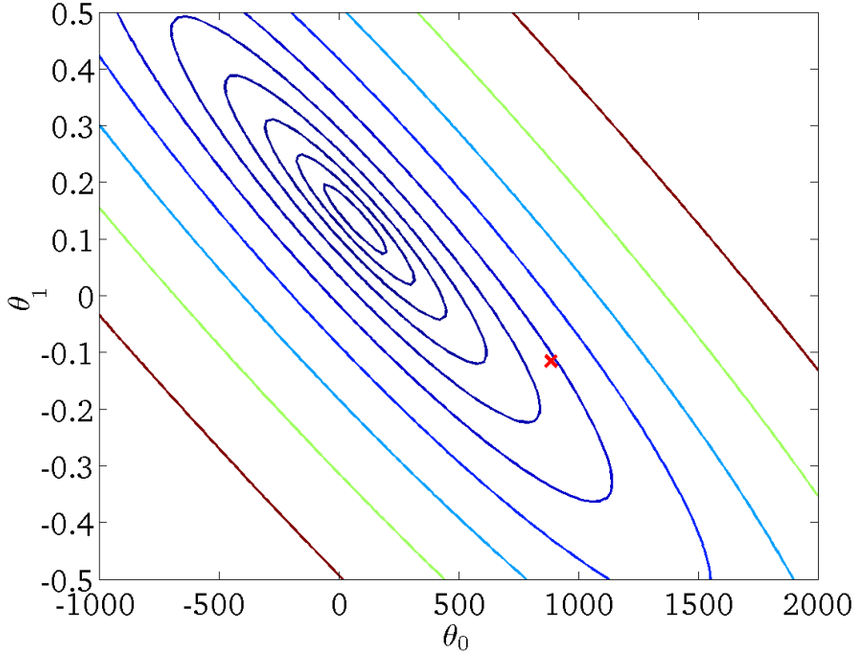
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

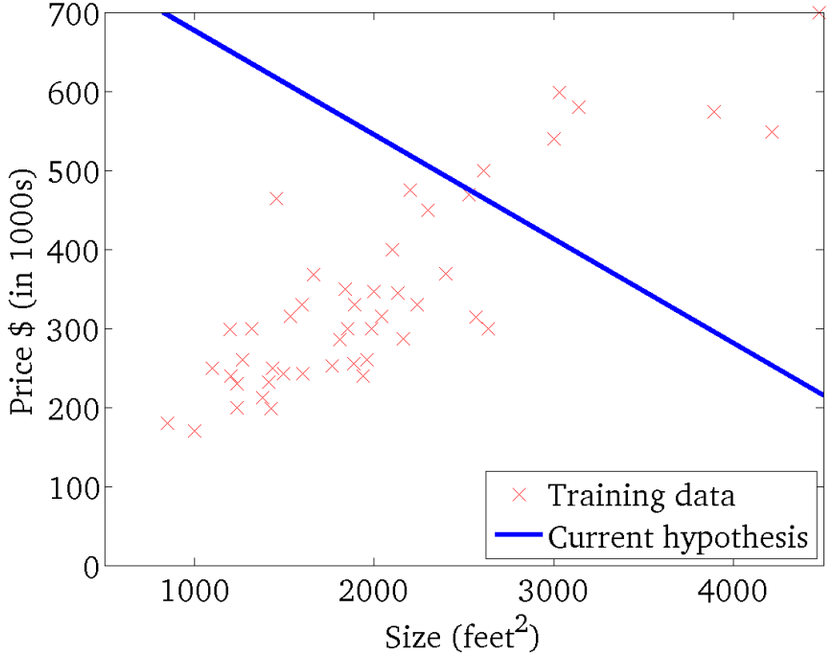
(function of the parameters θ_0, θ_1)



梯度下降法优化过程

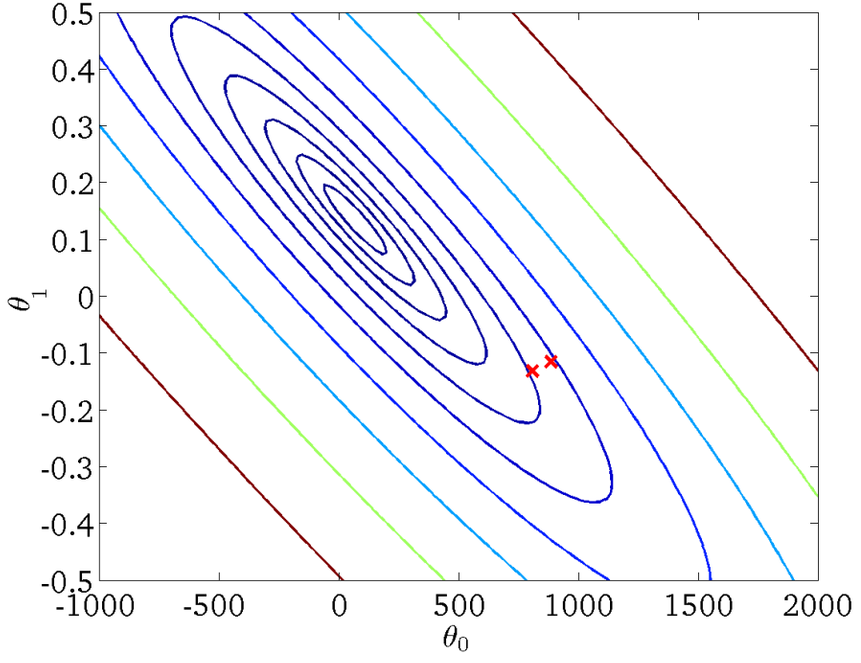
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

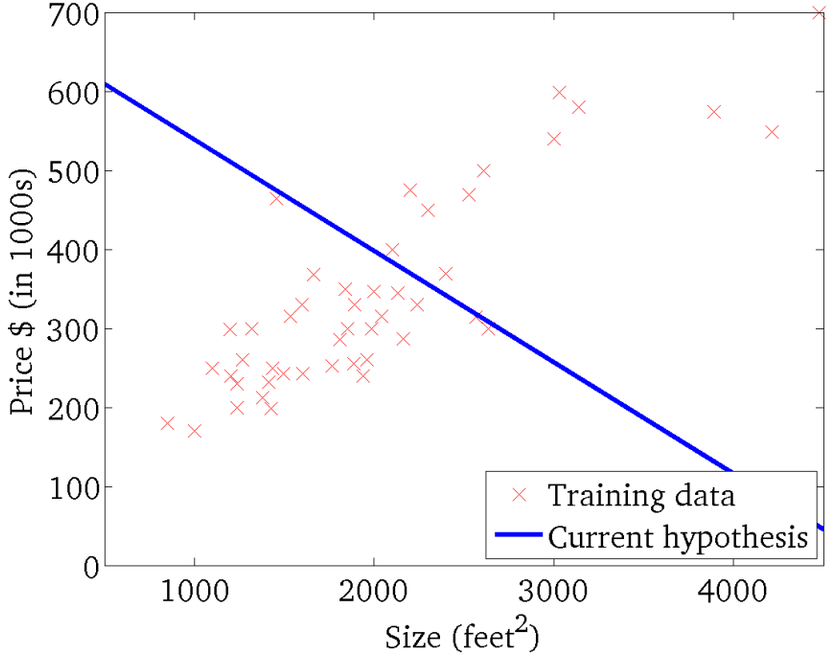
(function of the parameters θ_0, θ_1)



梯度下降法优化过程

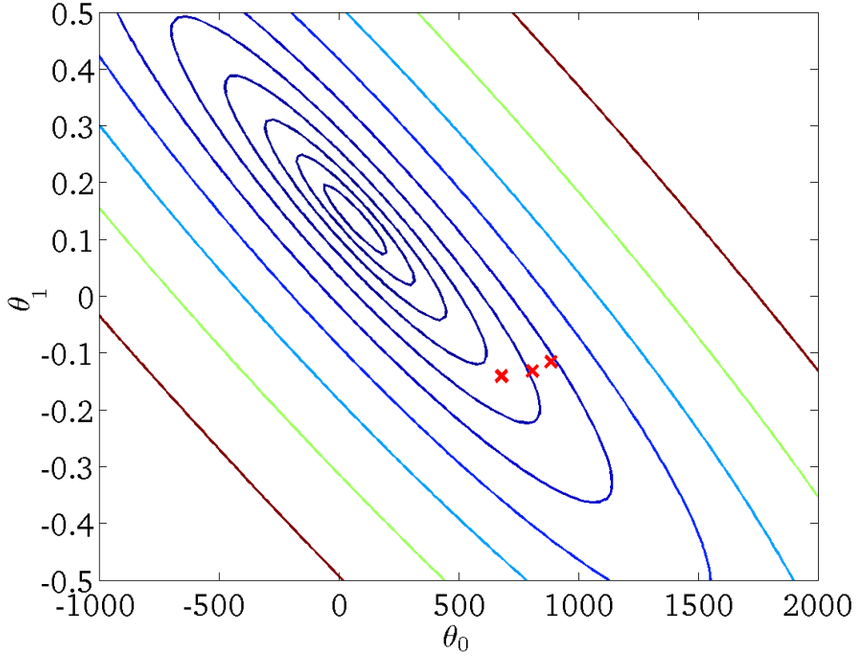
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

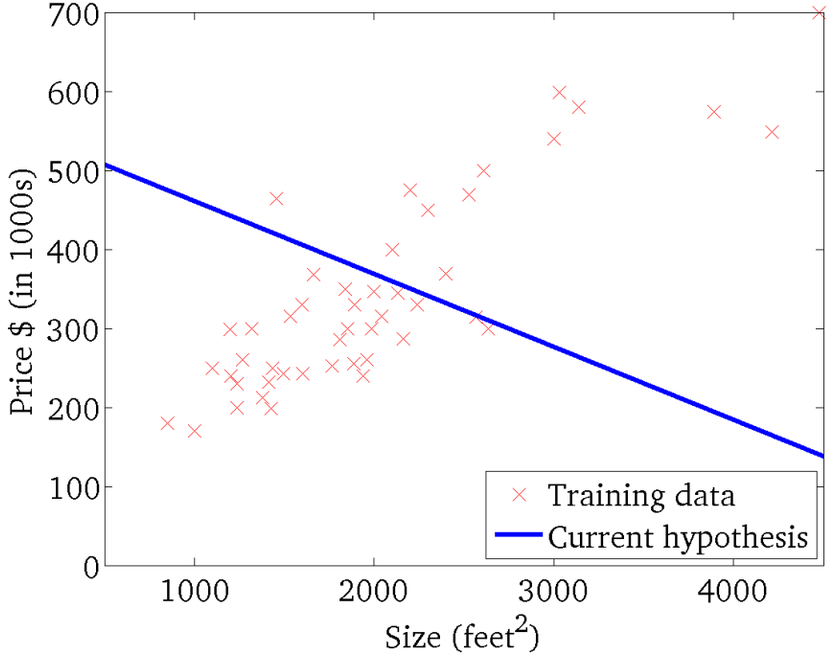
(function of the parameters θ_0, θ_1)



梯度下降法优化过程

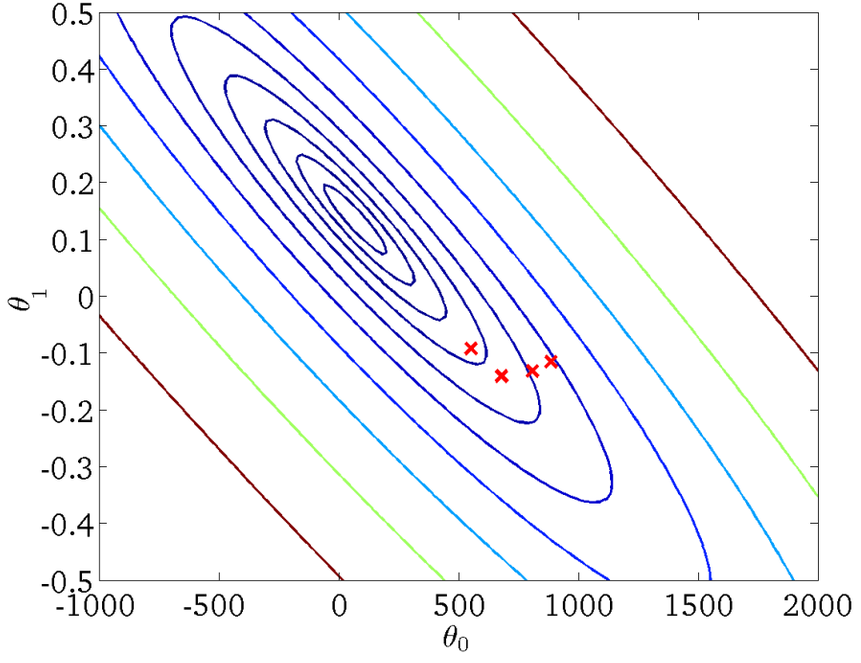
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

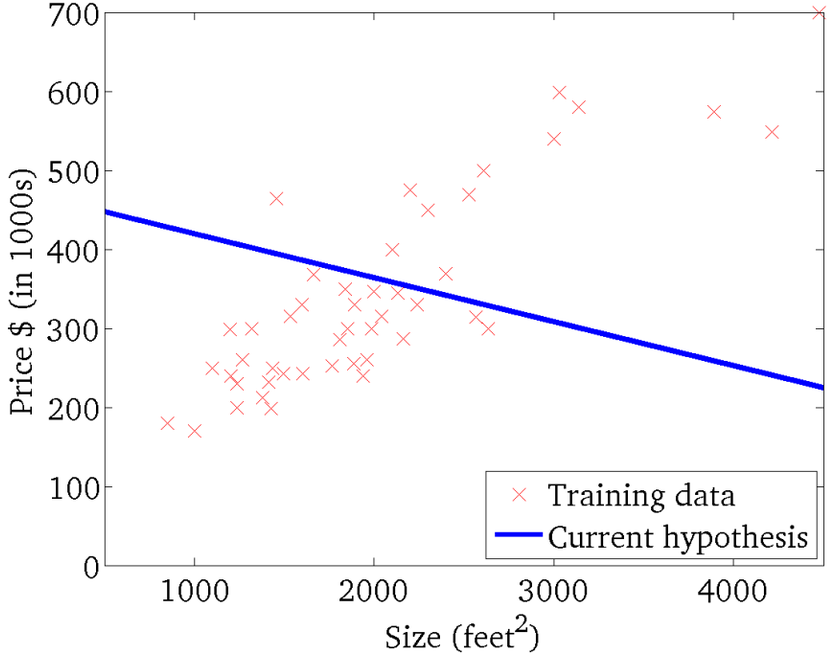
(function of the parameters θ_0, θ_1)



梯度下降法优化过程

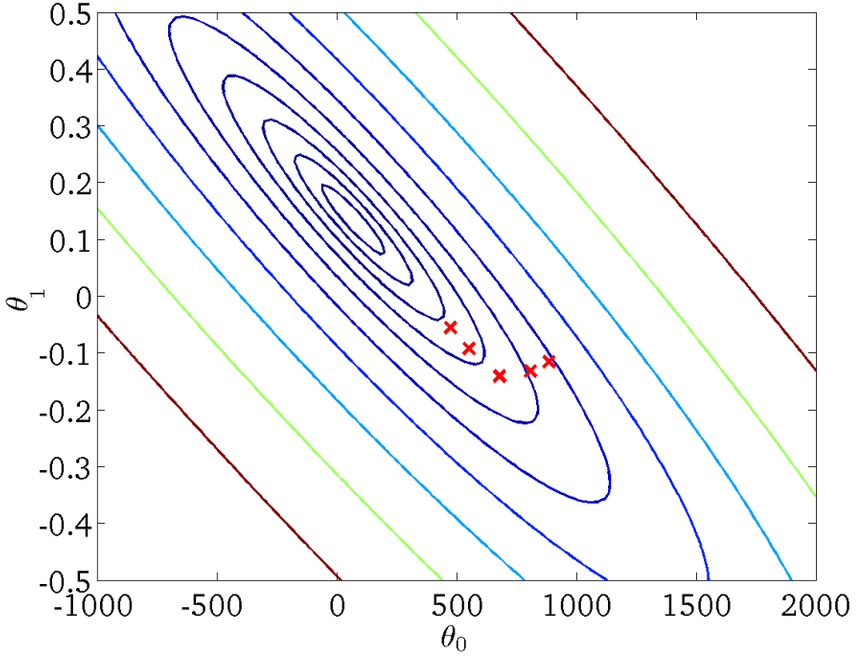
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

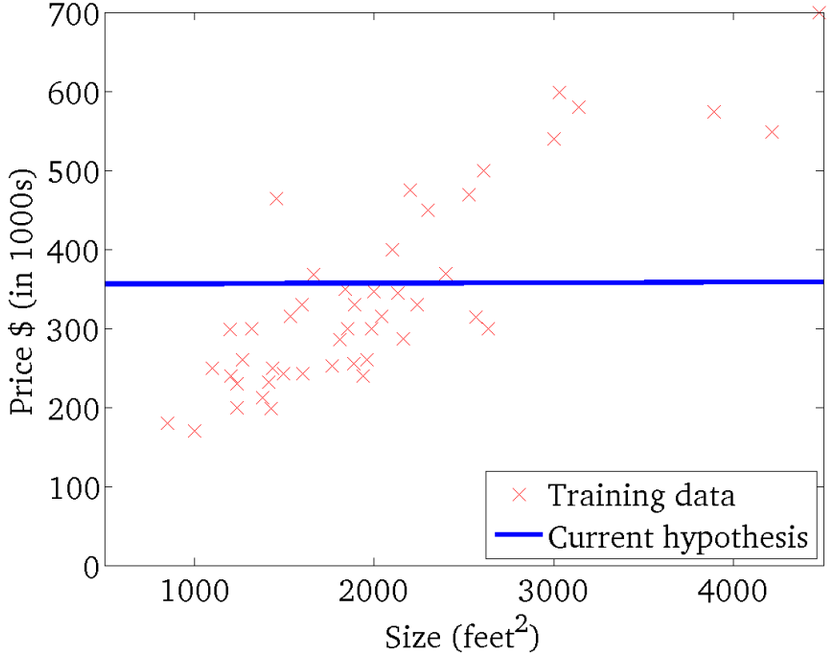
(function of the parameters θ_0, θ_1)



梯度下降法优化过程

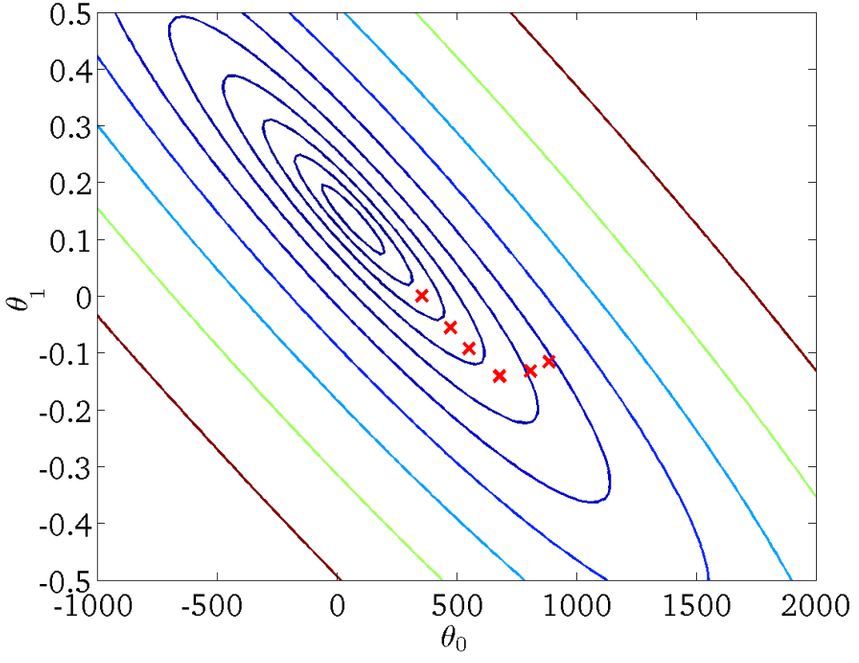
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

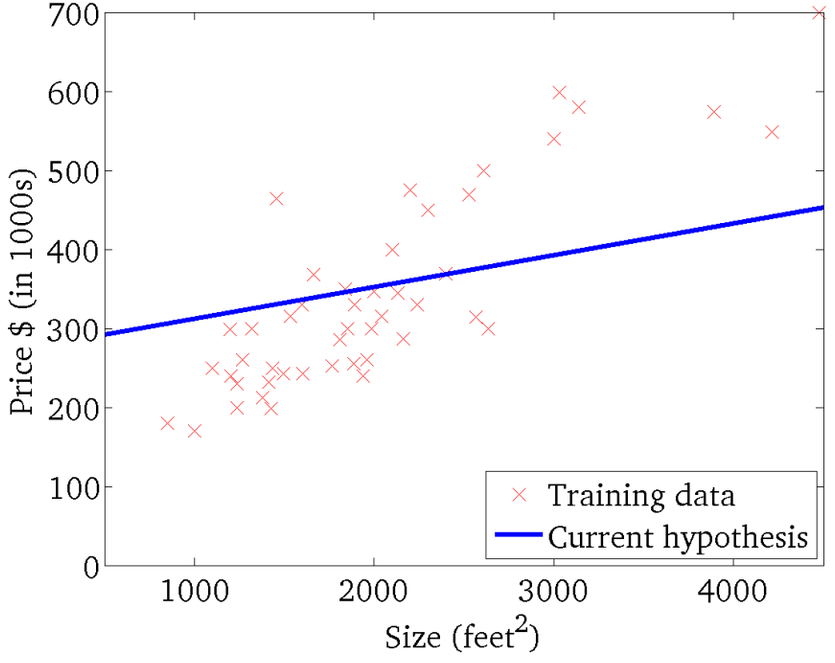
(function of the parameters θ_0, θ_1)



梯度下降法优化过程

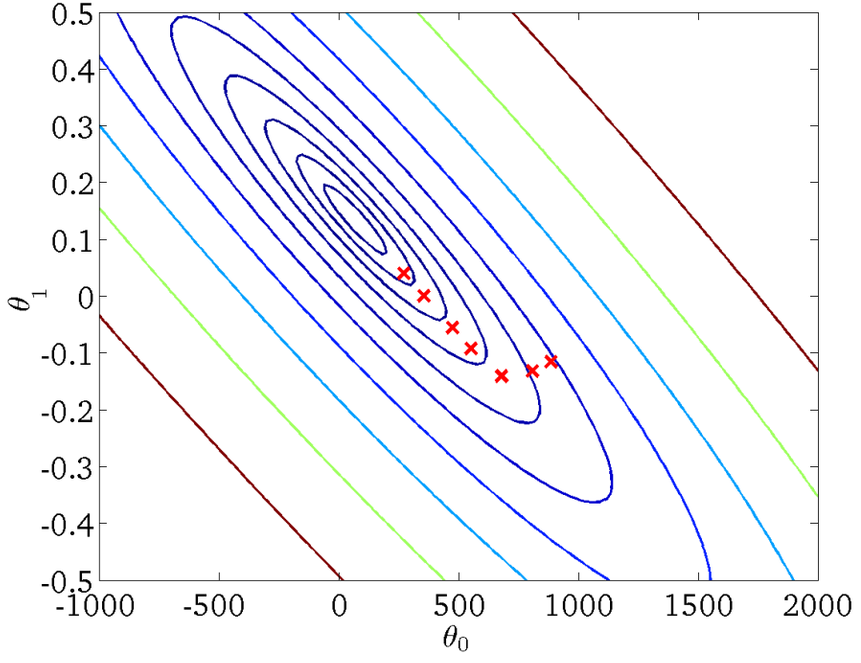
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

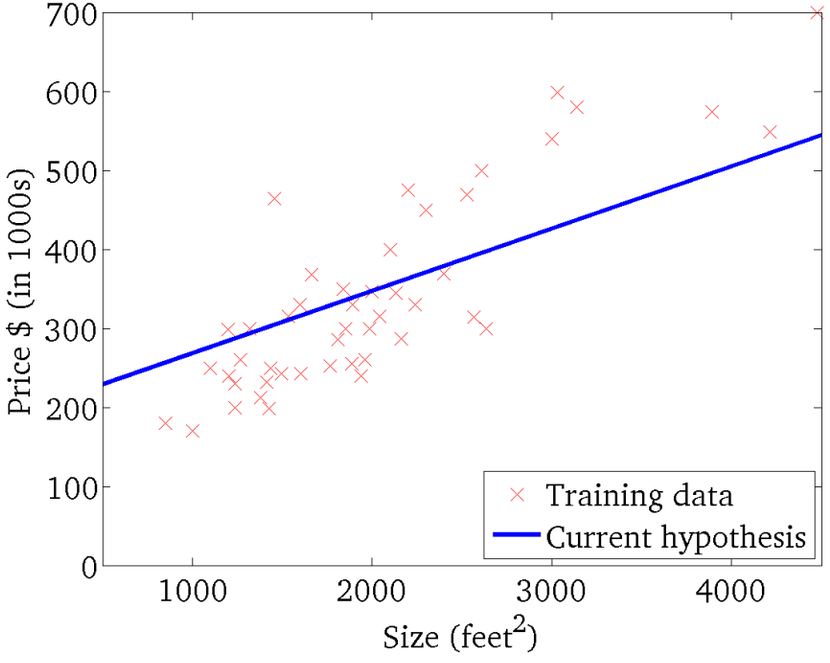
(function of the parameters θ_0, θ_1)



梯度下降法优化过程

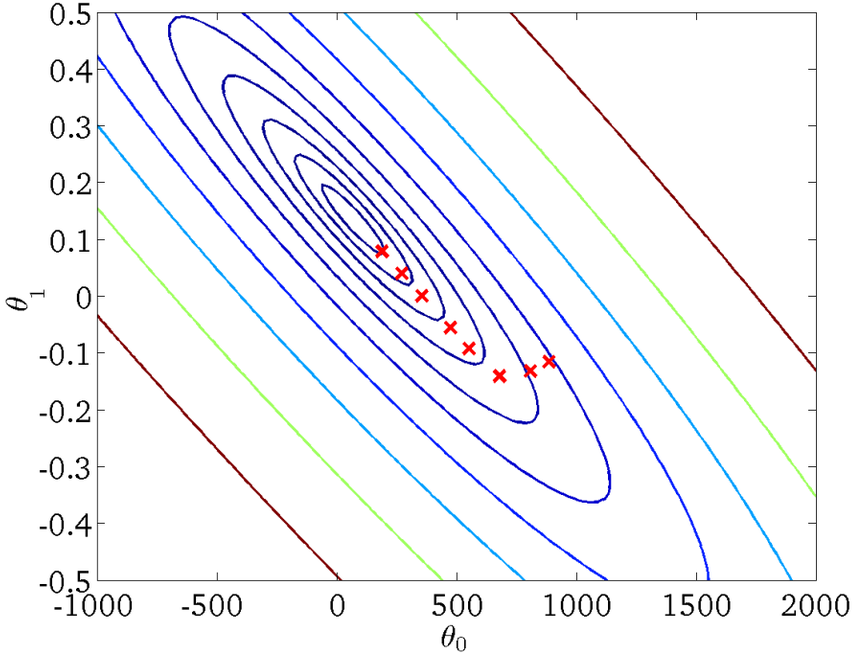
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

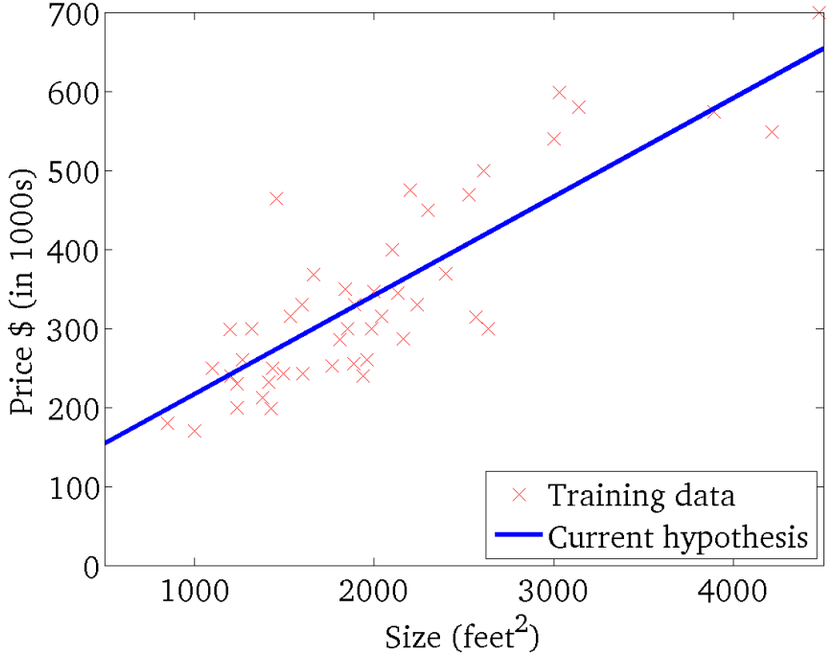
(function of the parameters θ_0, θ_1)



梯度下降法优化过程

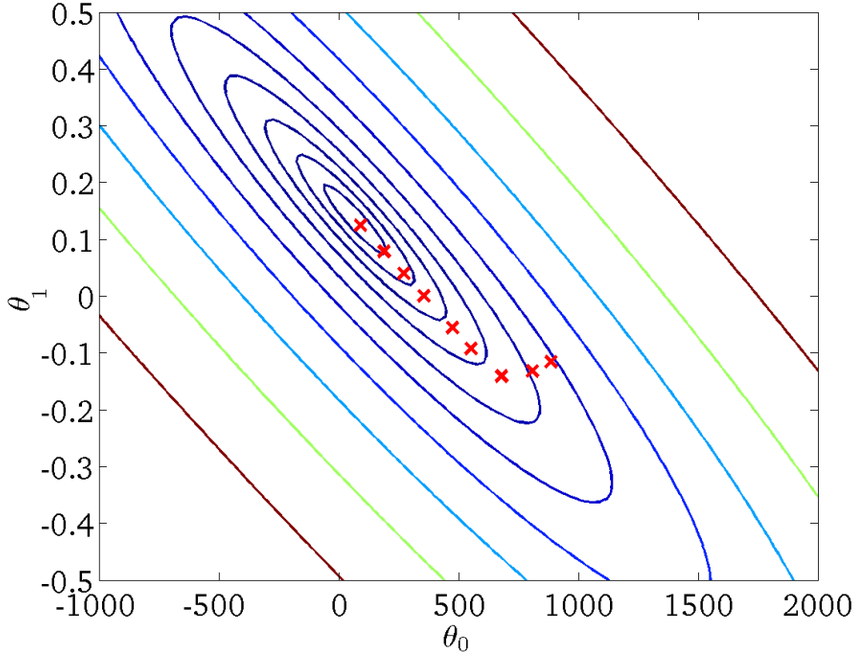
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



程序

sklearn-一元线性回归

多元线性回归

单特征

Size (feet²)	Price ($1000)
2104	460
1416	232
1534	315
852	178
…	…

$$h_\theta(x) = \theta_0 + \theta_1 x$$

多特征

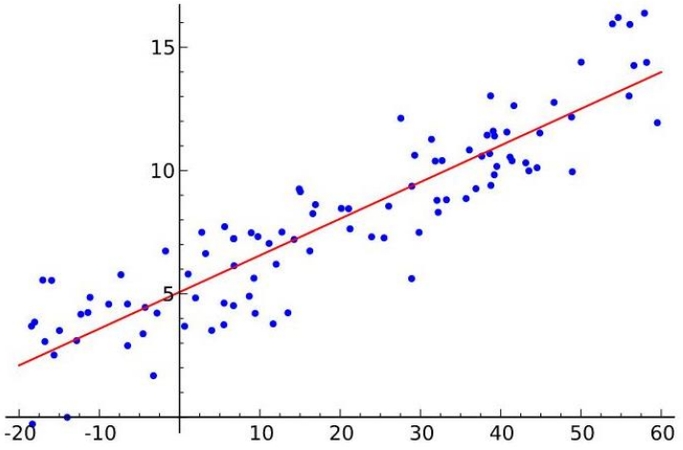
Size (feet²)	Number of bedrooms	Number of floors	Age of home (years)	Price ($1000)
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178
…	…	…	…	…

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

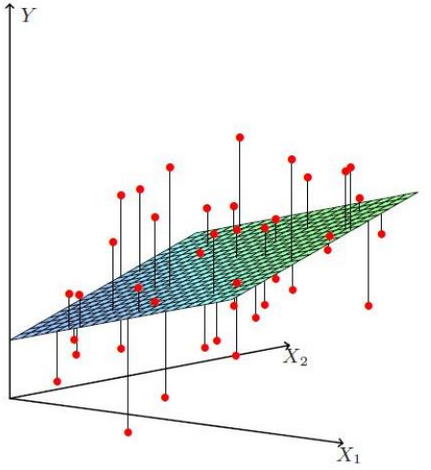
多元线性回归模型

当Y值的影响因素不是唯一时，采用多元线性回归模型

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$



一元线性回归



二元线性回归

多元线性回归

Hypothesis: $h_\theta(x) = \theta^T x = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$

Parameters: $\theta_0, \theta_1, \ldots, \theta_n$

Cost function:

$$J(\theta_0, \theta_1, \ldots, \theta_n) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

Gradient descent:

Repeat $\{$

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \ldots, \theta_n)$$

$\}$ (simultaneously update for every $j = 0, \ldots, n$)

多元线性回归

Gradient Descent

Previously (n=1):

Repeat $\{$

$$\theta_0 := \theta_0 - \alpha \underbrace{\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})}_{\frac{\partial}{\partial\theta_0}J(\theta)}$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x^{(i)}$$

(simultaneously update θ_0, θ_1)

$\}$

New algorithm $(n \geq 1)$

Repeat $\{$

$$\theta_j := \theta_j - \alpha \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$$

(simultaneously update θ_j for $j = 0, \ldots, n$)

$\}$

$$\theta_0 := \theta_0 - \alpha \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_0^{(i)}$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_1^{(i)}$$

$$\theta_2 := \theta_2 - \alpha \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_2^{(i)}$$

例子

一家快递公司送货：X1： 运输里程 X2：运输次数　Y：总运输时间

Driving Assignment	X1=Miles Traveled	X2=Number of Deliveries	Y= Travel Time (Hours)
1	100	4	9.3
2	50	3	4.8
3	100	4	8.9
4	100	2	6.5
5	50	2	4.2
6	80	2	6.2
7	75	3	7.4
8	65	4	6.0
9	90	3	7.6
10	90	2	6.1

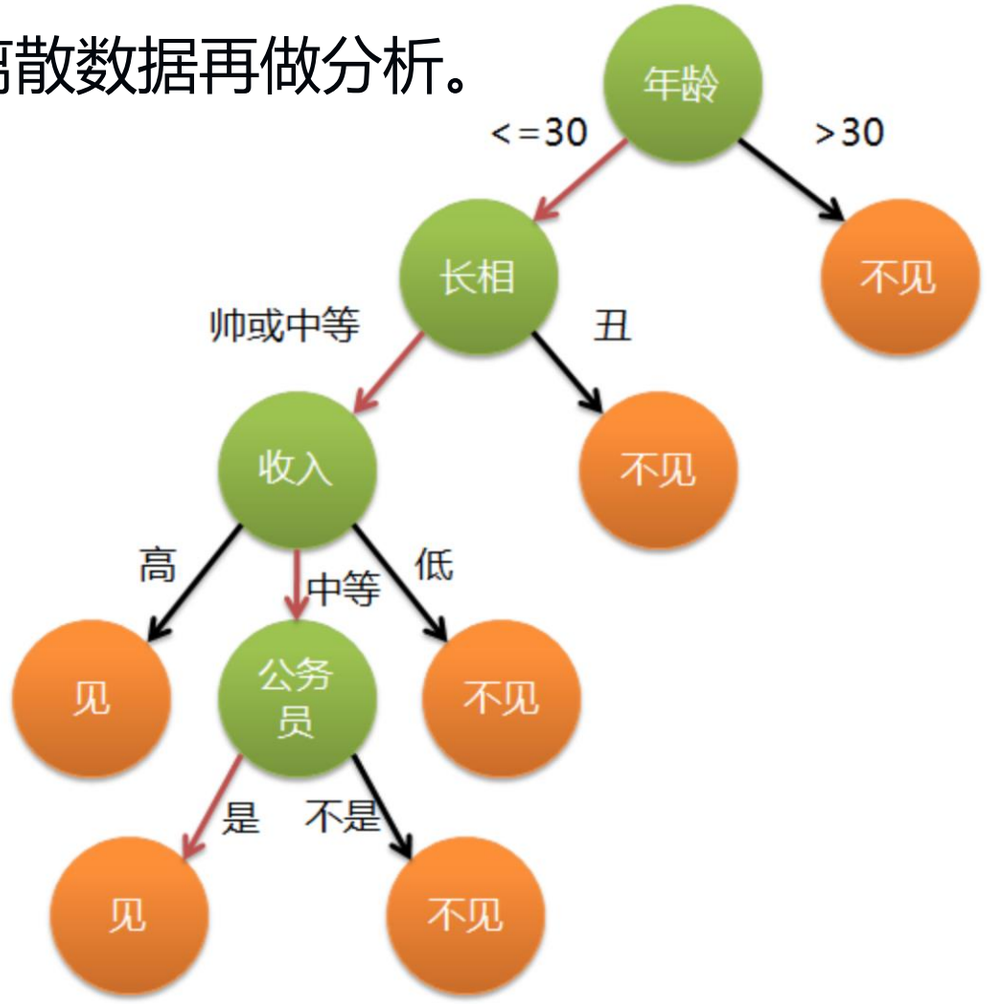
程序

sklearn-多元线性回归

决策树

决策树

比较适合分析离散数据
如果是连续数据要先转成离散数据再做分析。



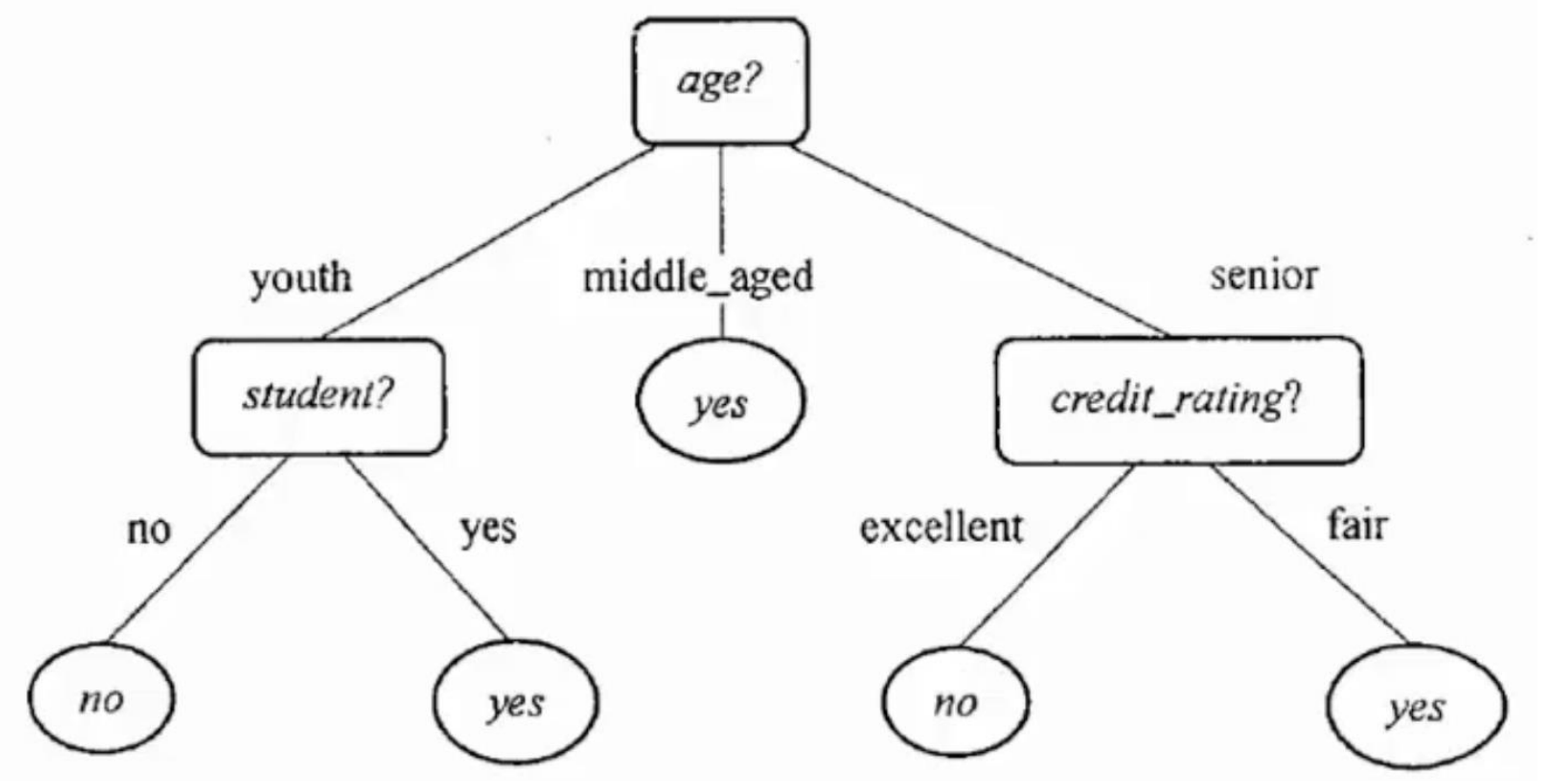
决策树算法

- 70年代后期至80年代，Quinlan开发了ID3算法。
- Quinlan改进了ID3算法，称为C4.5算法。
- 1984年，多位统计学家提出了CART算法。

例子：训练数据

RID	age	income	student	credit_rating	Class: buys_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

例子：期待输出的结果


age?
youth
middle_aged
senior
student?
yes
credit_rating?
no
yes
excellent
fair
no
yes
no
yes

熵(entropy)概念：

- 1948年，香农提出了“信息熵”的概念
- 一条信息的信息量大小和它的不确定性有直接的关系，要搞清楚一件非常非常不确定的事情，或者是我们一无所知的事情，需要了解大量信息->信息量的度量就等于不确定性的多少。

信息熵计算

信息熵公式： $H[x] = -\sum_{x} p(x) log_2 p(x)$

假如有一个普通骰子A，仍出1-6的概率都是1/6

有一个骰子B，扔出6的概率是50%，扔出1-5的概率都是10%

有一个骰子C，扔出6的概率是100%。

信息熵计算

骰子A：$-\left(\frac{1}{6}\times\log_2\frac{1}{6}\right)\times 6\approx 2.585$

骰子B：$-\left(\frac{1}{10}\times\log_2\frac{1}{10}\right)\times 5-\frac{1}{2}\times\log_2\frac{1}{2}\approx 2.161$

骰子C：$-(1\times\log_2 1)=0$

ID3算法

信息增益计算：

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j)$$

$$Gain(A) = Info(D) - Info_A(D)$$

选择根节点-ID3算法

信息增益(Information Gain)：Gain(A) = Info(D) - Infor_A(D)

RID	age	income	student	credit_rating	Class: buys_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

$$Info(D) = -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0.940 \text{ bits.}$$

$$\begin{aligned} Info_{age}(D) &= \frac{5}{14} \times \left(-\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5}\right) \\ &+ \frac{4}{14} \times \left(-\frac{4}{4}\log_2\frac{4}{4} - \frac{0}{4}\log_2\frac{0}{4}\right) \\ &+ \frac{5}{14} \times \left(-\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}\right) \\ &= 0.694 \text{ bits.} \end{aligned}$$

$$Gain(age) = Info(D) - Info_{age}(D) = 0.940 - 0.694 = 0.246 \text{ bits.}$$

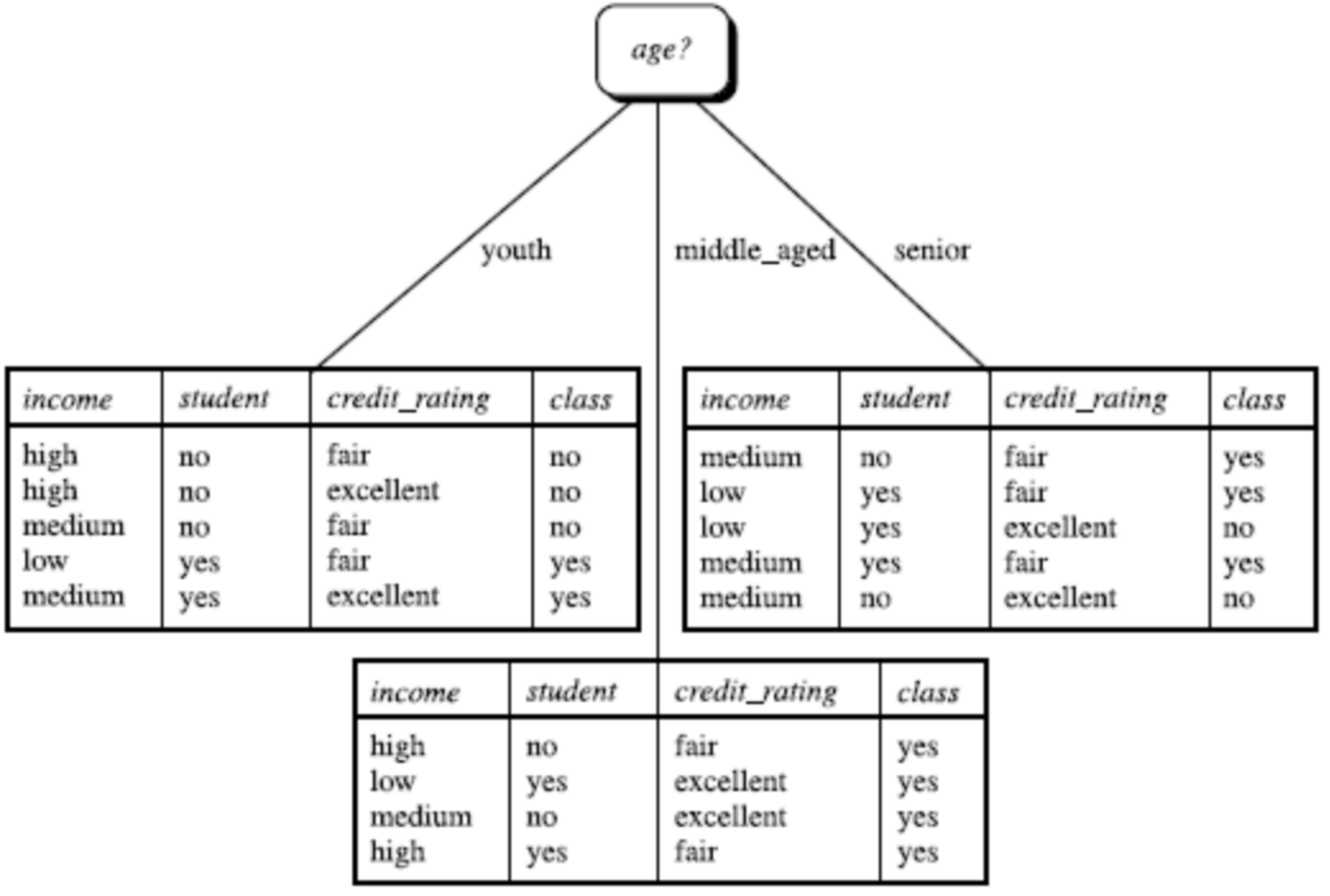
类似：

Gain(income) = 0.029

Gain(student) = 0.151

Gain(credit_rating)=0.048

选择根节点-ID3算法



income	*student*	*credit_rating*	*class*
high	no	fair	no
high	no	excellent	no
medium	no	fair	no
low	yes	fair	yes
medium	yes	excellent	yes

income	*student*	*credit_rating*	*class*
medium	no	fair	yes
low	yes	fair	yes
low	yes	excellent	no
medium	yes	fair	yes
medium	no	excellent	no

income	*student*	*credit_rating*	*class*
high	no	fair	yes
low	yes	excellent	yes
medium	no	excellent	yes
high	yes	fair	yes

连续变量处理

Age：

10

12

14

17

20

23

26

32

36

42

48

49

50

C4.5算法

信息增益的方法倾向于首先选择因子数较多的变量
信息增益的改进：增益率

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2\left(\frac{|D_j|}{|D|}\right)$$

$$GrianRate(A) = \frac{Grain(A)}{SplitInfo_A(D)}$$

CART算法

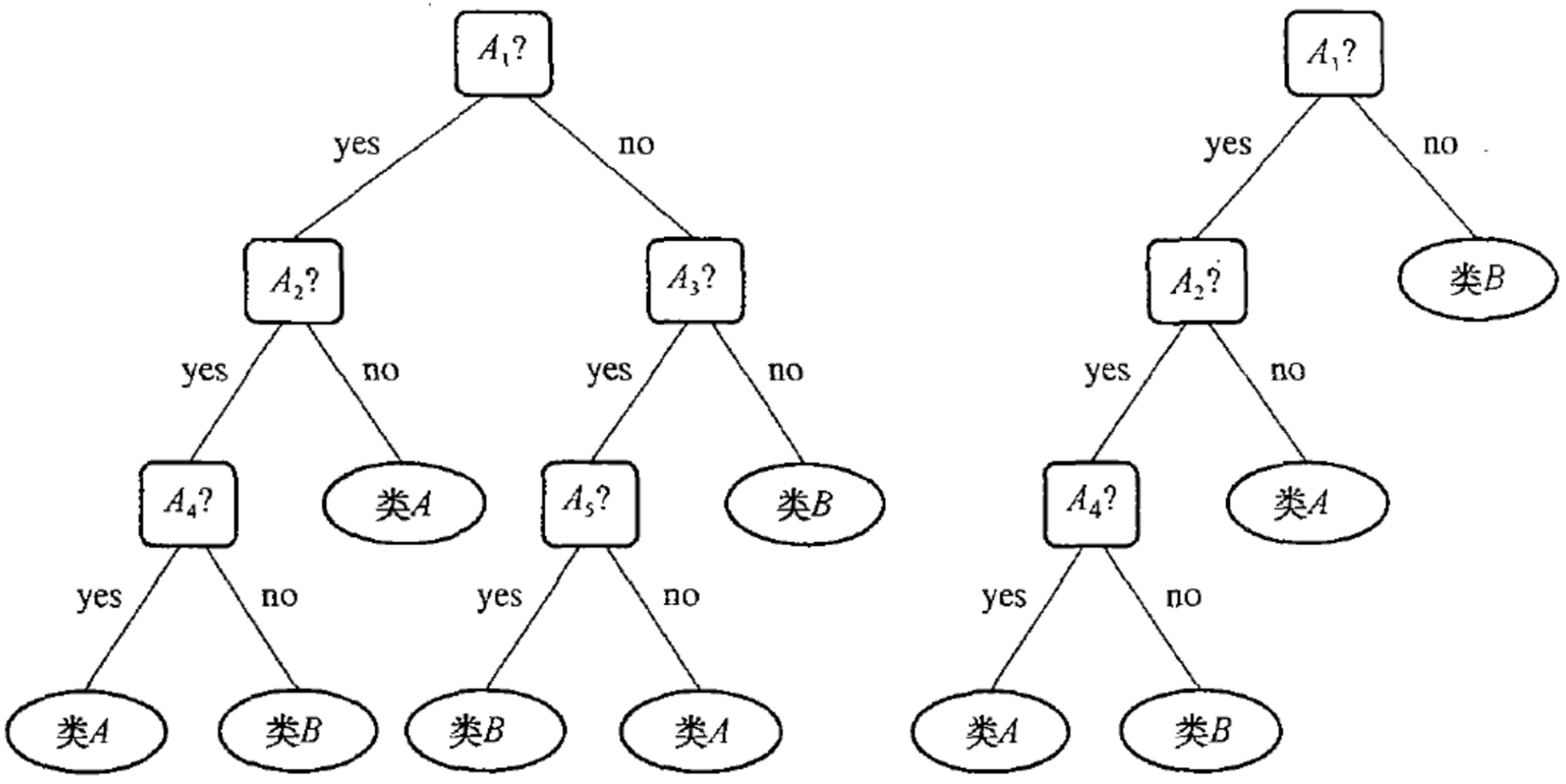
使用基尼指数选择变量：

$$Gini(D) = 1 - \sum_{i=1}^{m} p_i^2$$

$$Gini_A(D) = \frac{|D_1|}{|D|}Gini(D_1) + \frac{|D_2|}{|D|}Gini(D_2)$$

$$\Delta Gini(A) = Gini(D) - Gini_A(D)$$

剪枝


A_1?
yes
no
A_2?
A_3?
yes
no
yes
no
A_4?
类A
A_5?
类B
yes
no
yes
no
类A
类B
类B
类A
A_1?
yes
no
A_2?
类B
yes
no
A_4?
类A
yes
no
类A
类B

优缺点

优点：

- 小规模数据集有效

缺点：

- 处理连续变量不好
- 类别较多时，错误增加的比较快
- 不能处理大量数据

决策树程序

最邻近规则分类(KNN)

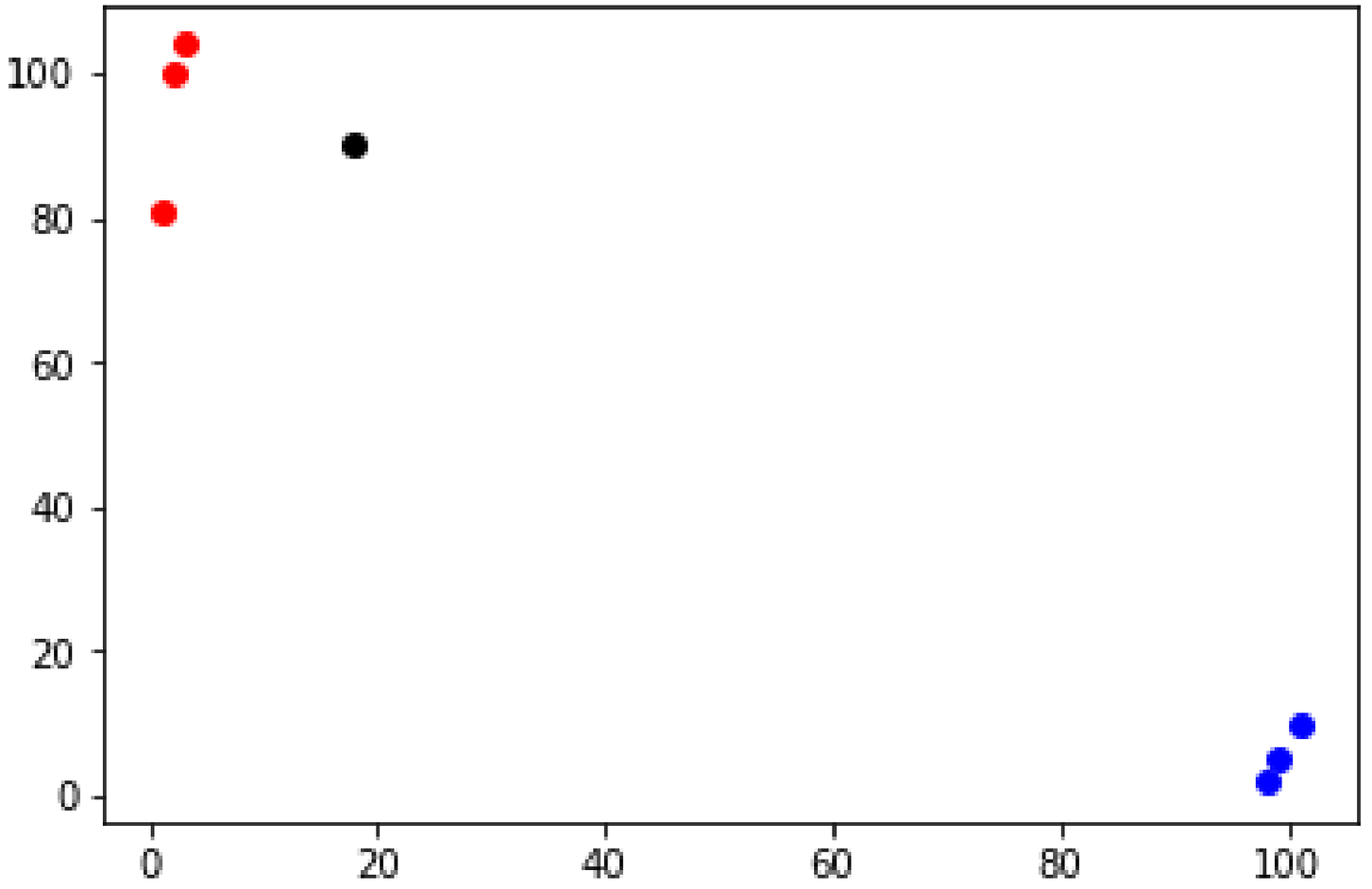
KNN例子

电影名称	打斗次数	接吻次数	电影类型
California Man	3	104	Romance
He’s Not Really into Dudes	2	100	Romance
Beautiful Woman	1	81	Romance
Kevin Longblade	101	10	Action
Robo Slayer 3000	99	5	Action
Amped II	98	2	Action
未知	18	90	Unknown

KNN例子

点	X坐标	Y坐标	点类型
A点	3	104	Romance
B点	2	100	Romance
C点	1	81	Romance
D点	101	10	Action
E点	99	5	Action
F点	98	2	Action
G点	18	90	Unknown

KNN例子

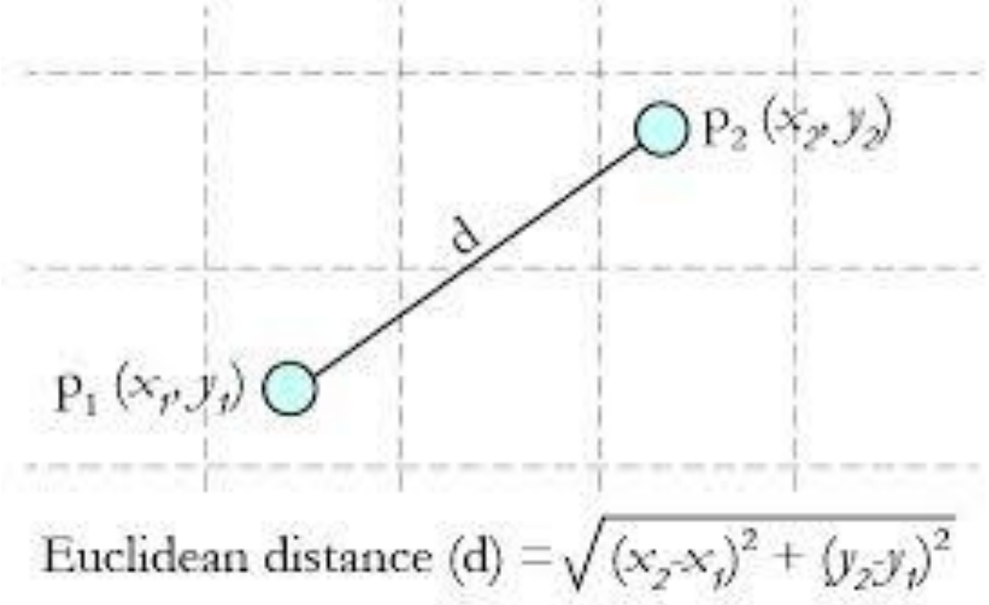


KNN算法

- 为了判断未知实例的类别，以所有已知类别的实例作为参照选择参数K
- 计算未知实例与所有已知实例的距离
- 选择最近K个已知实例
- 根据少数服从多数的投票法则(majority-voting)，让未知实例归类为K个最邻近样本中最多数的类别

欧氏距离

欧式距离也称为欧几里得距离



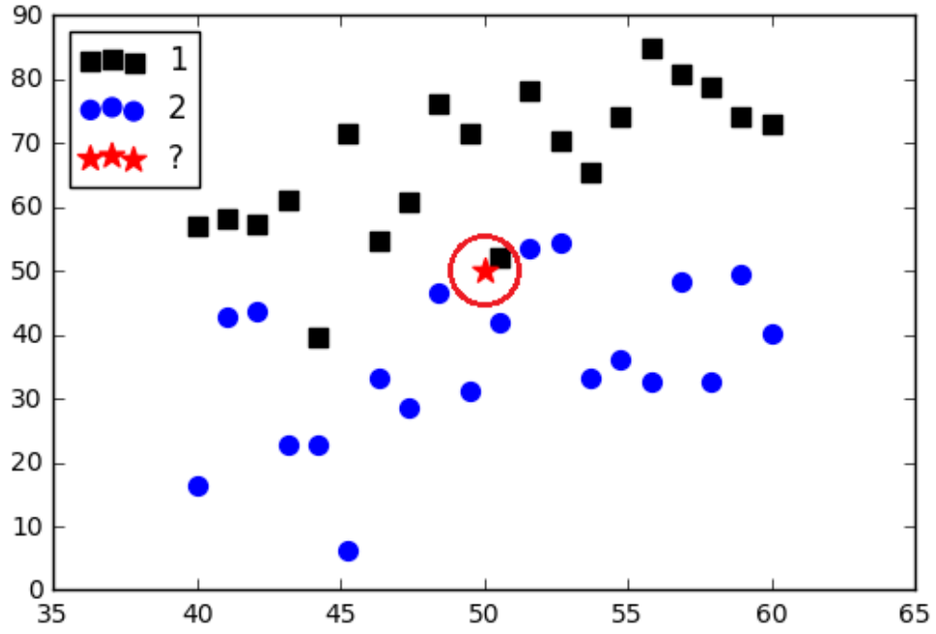
$$E(x,y) = \sqrt{\sum_{i=0}^{n}(x_i - y_i)^2}$$

其他距离衡量：余弦值距离（cos），相关度 （correlation），曼哈顿距离 （Manhattan distance）

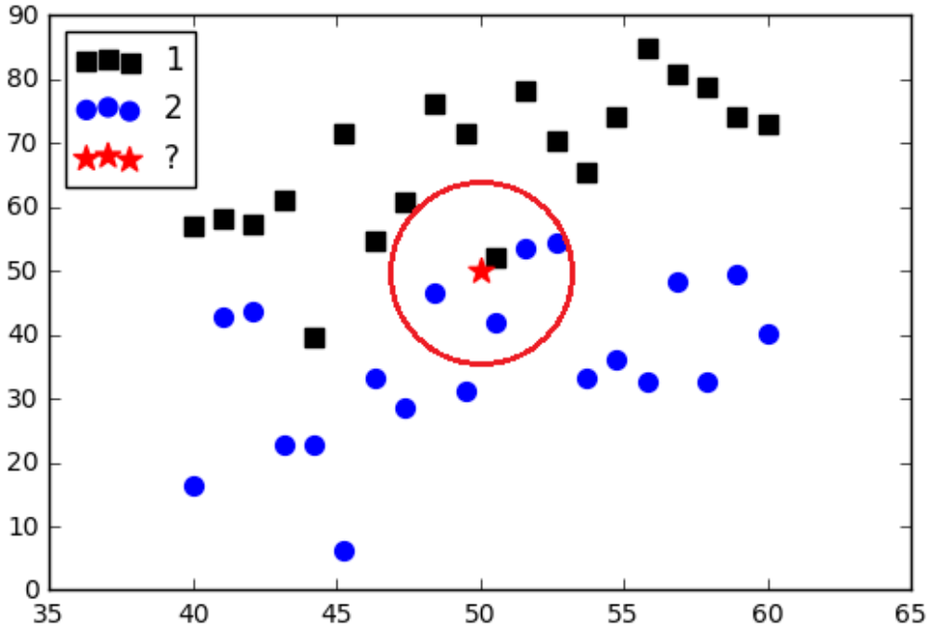
http://www.cnblogs.com/belfuture/p/5871452.html

K值选取

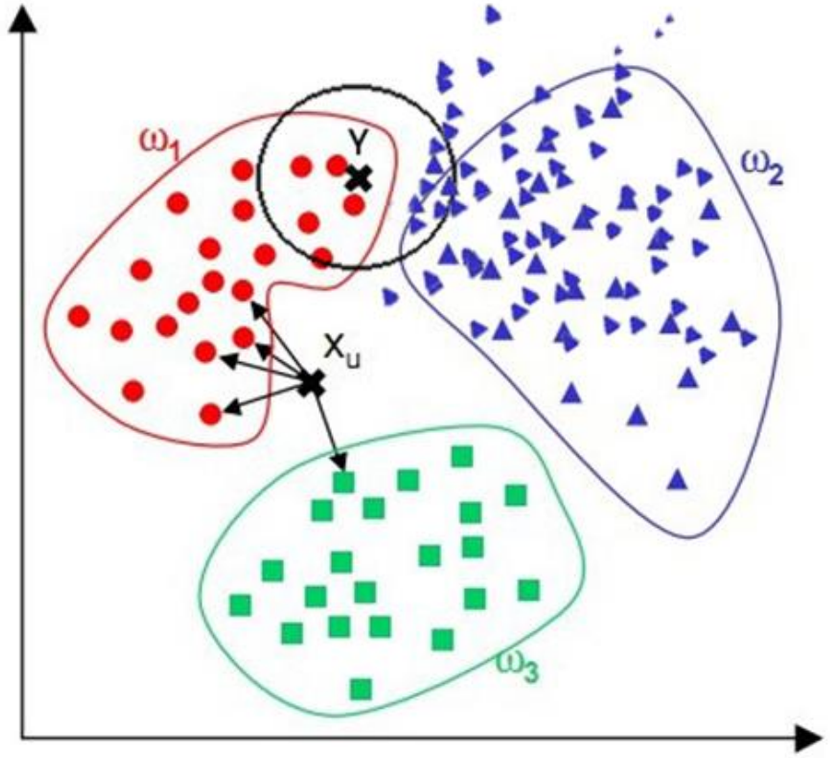
K=1


1
2
?
90
80
70
60
50
40
30
20
10
0
35
40
45
50
55
60
65

K=5


1
2
?
90
80
70
60
50
40
30
20
10
0
35
40
45
50
55
60
65

算法缺点



算法复杂度较高（需要比较所有已知实例与要分类的实例）
当其样本分布不平衡时，比如其中一类样本过大（实例数量过多）占主导的时候，新的未知实例容易被归类为这个主导样本，因为这类样本实例的数量过大，但这个新的未知实例实际并没有接近目标样本

Iris数据集

数据属性：萼片长度，萼片宽度，花瓣长度，花瓣宽度
(sepal length, sepal width, petal length and petal width)
类别：Iris setosa, Iris versicolor, Iris virginica

程序

鸢尾花分类程序

聚类算法(K-MEANS)

K-means算法

- Clustering 中的经典算法，数据挖掘十大经典算法之一
- 算法接受参数 k ；然后将事先输入的n个数据对象划分为 k个聚类以便使得所获得的聚类满足：同一聚类中的对象相似度较高；而不同聚类中的对象相似度较小。
- 算法思想：以空间中k个点为中心进行聚类，对最靠近他们的对象归类。通过迭代的方法，逐次更新各聚类中心的值，直至得到最好的聚类结果

K-means算法

1.先从没有标签的元素集合A中随机取k个元素，作为k个子集各自的重心。

2.分别计算剩下的元素到k个子集重心的距离（这里的距离也可以使用欧氏距离），根据距离将这些元素分别划归到最近的子集。

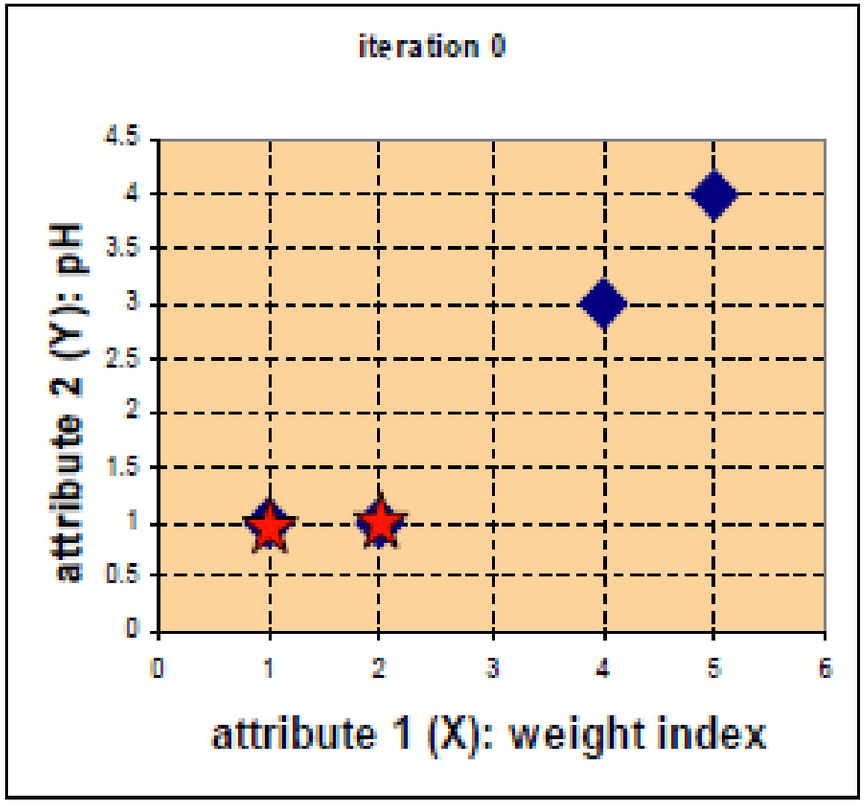
3.根据聚类结果，重新计算重心（重心的计算方法是计算子集中所有元素各个维度的算数平均数）。

4.将集合A中全部元素按照新的重心然后再重新聚类。

5.重复第4步，直到聚类结果不再发生变化。

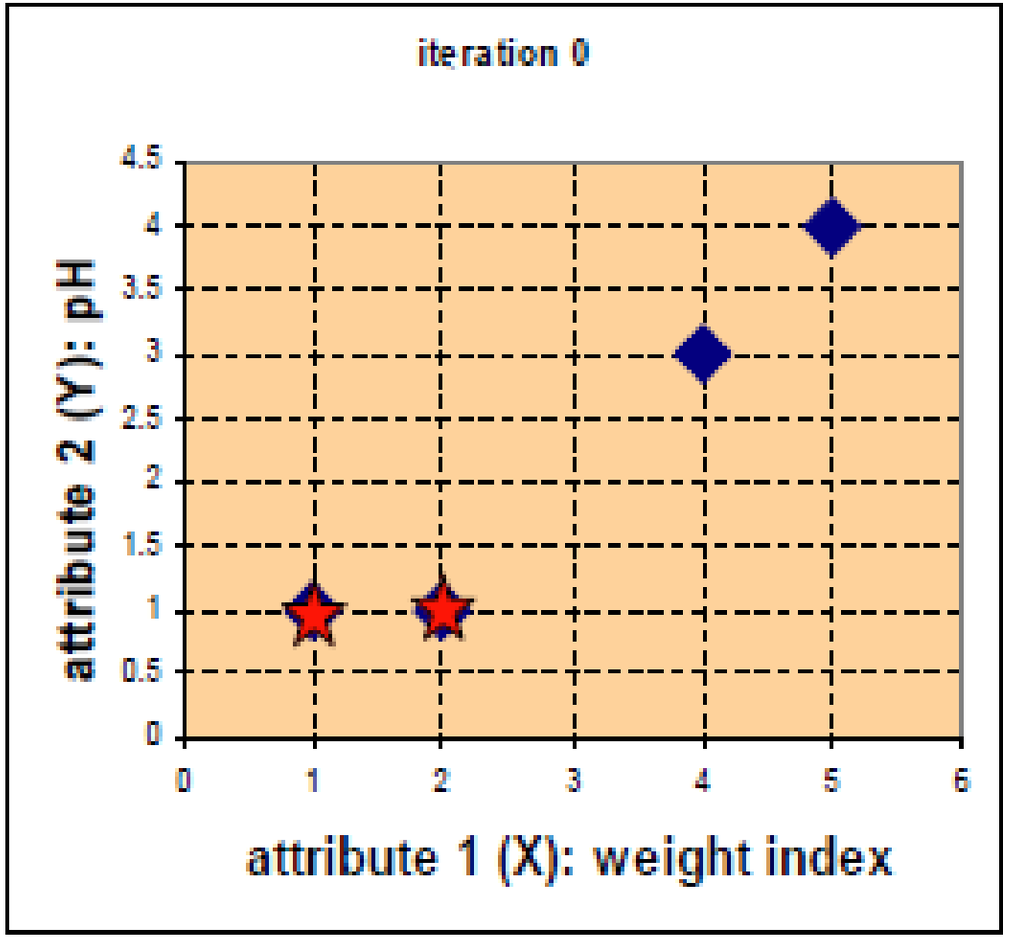
例子

Object	Feature 1 (X): weight index	Feature 2 (Y): pH
Medicine A	1	1
Medicine B	2	1
Medicine C	4	3
Medicine D	5	4



假设取(1,1)(2,1)为两个分类中心点

例子



$$D^0 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 1 & 0 & 2.83 & 4.24 \end{bmatrix} \quad \begin{matrix} c_1 = (1,1) & group-1 \\ c_2 = (2,1) & group-2 \end{matrix}$$

$$\begin{matrix} A & B & C & D \end{matrix}$$

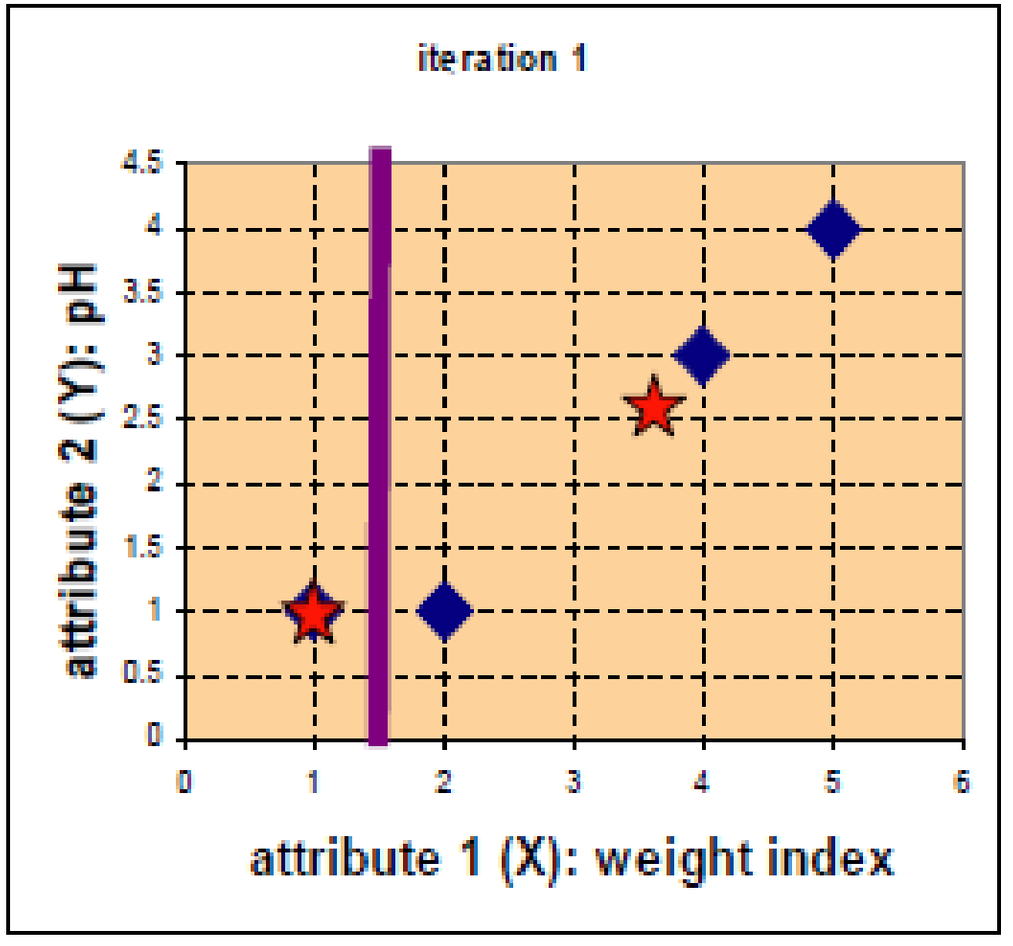
$$\begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix} \quad \begin{matrix} X \\ Y \end{matrix}$$

$$G^0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} \quad \begin{matrix} group-1 \\ group-2 \end{matrix}$$

$$\begin{matrix} A & B & C & D \end{matrix}$$

$$c_2 = \left(\frac{2+4+5}{3}, \frac{1+3+4}{3}\right) = \left(\frac{11}{3}, \frac{8}{3}\right)$$

例子



$$D^0 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 3.14 & 2.36 & 0.47 & 1.89 \end{bmatrix} \quad \begin{matrix} c_1 = (1,1) & group-1 \\ c_2 = (\frac{11}{3}, \frac{8}{3}) & group-2 \end{matrix}$$

$$\begin{matrix} A & B & C & D \end{matrix}$$

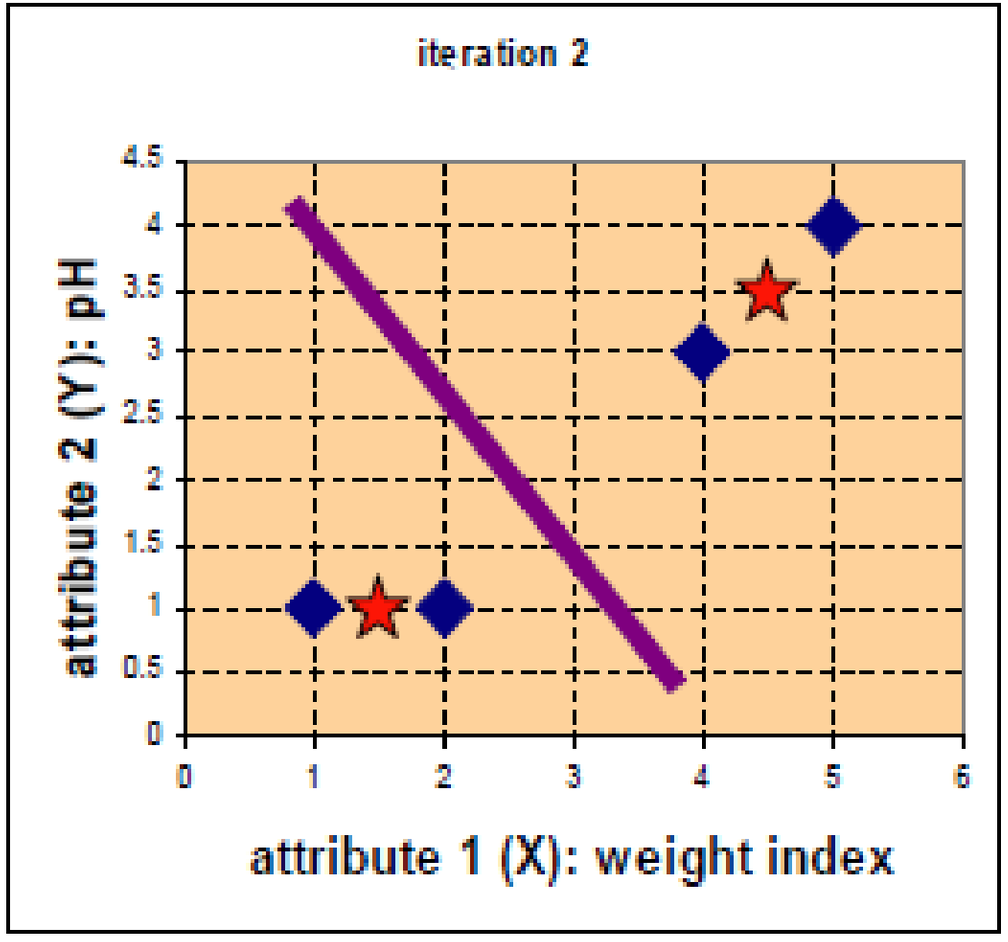
$$\begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix} \begin{matrix} X \\ Y \end{matrix}$$

$$G^1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{matrix} group-1 \\ group-2 \end{matrix}$$

$$\begin{matrix} A & B & C & D \end{matrix}$$

$$c_1 = (\frac{1+2}{2}, \frac{1+1}{2}) = (1\frac{1}{2}, 1) \text{ and } c_2 = (\frac{4+5}{2}, \frac{3+4}{2}) = (4\frac{1}{2}, 3\frac{1}{2})$$

例子



$$D^2 = \begin{bmatrix} 0.5 & 0.5 & 3.20 & 4.61 \\ 4.30 & 3.54 & 0.71 & 0.71 \end{bmatrix} \quad \begin{matrix} c_1 = (1\frac{1}{2}, 1) & group-1 \\ c_2 = (4\frac{1}{2}, 3\frac{1}{2}) & group-2 \end{matrix}$$

$$\begin{matrix} A & B & C & D \end{matrix}$$

$$\begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix} \quad \begin{matrix} X \\ Y \end{matrix}$$

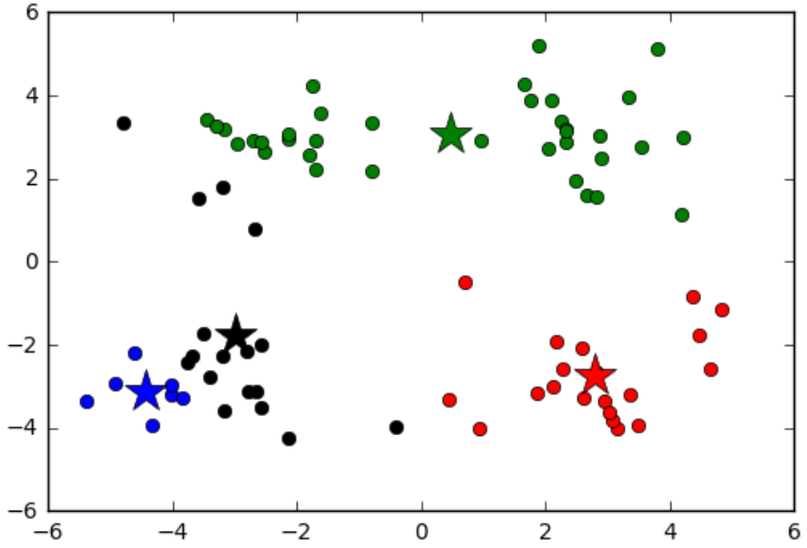
$$G^2 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad \begin{matrix} group-1 \\ group-2 \end{matrix}$$

$$\begin{matrix} A & B & C & D \end{matrix}$$

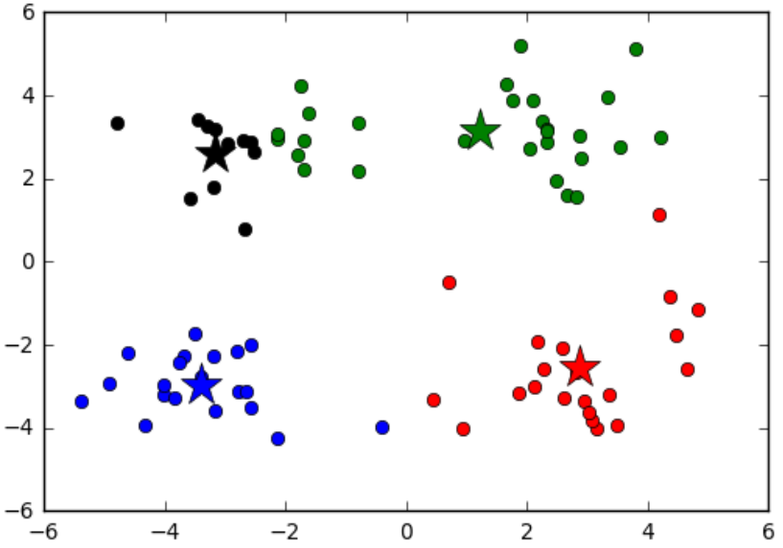
聚类不发生变化，算法迭代停止

K-means算法

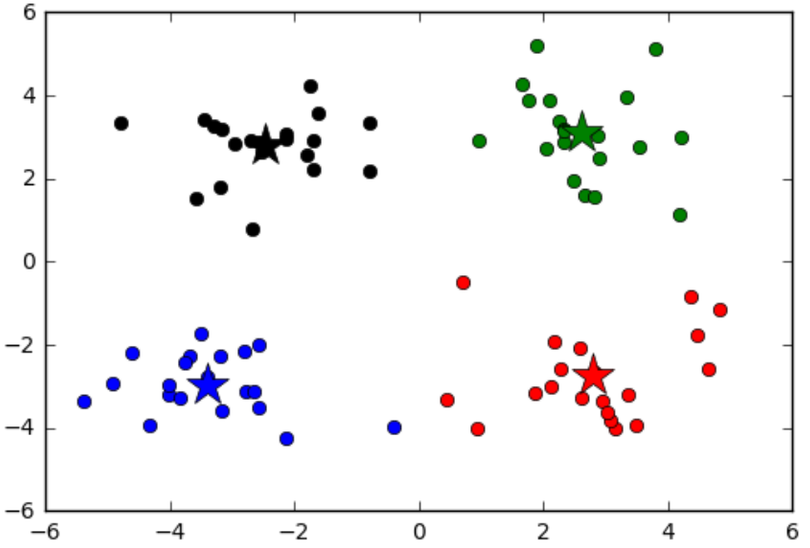
迭代了1次



迭代了5次



迭代了9次



程序

K-MEANS程序

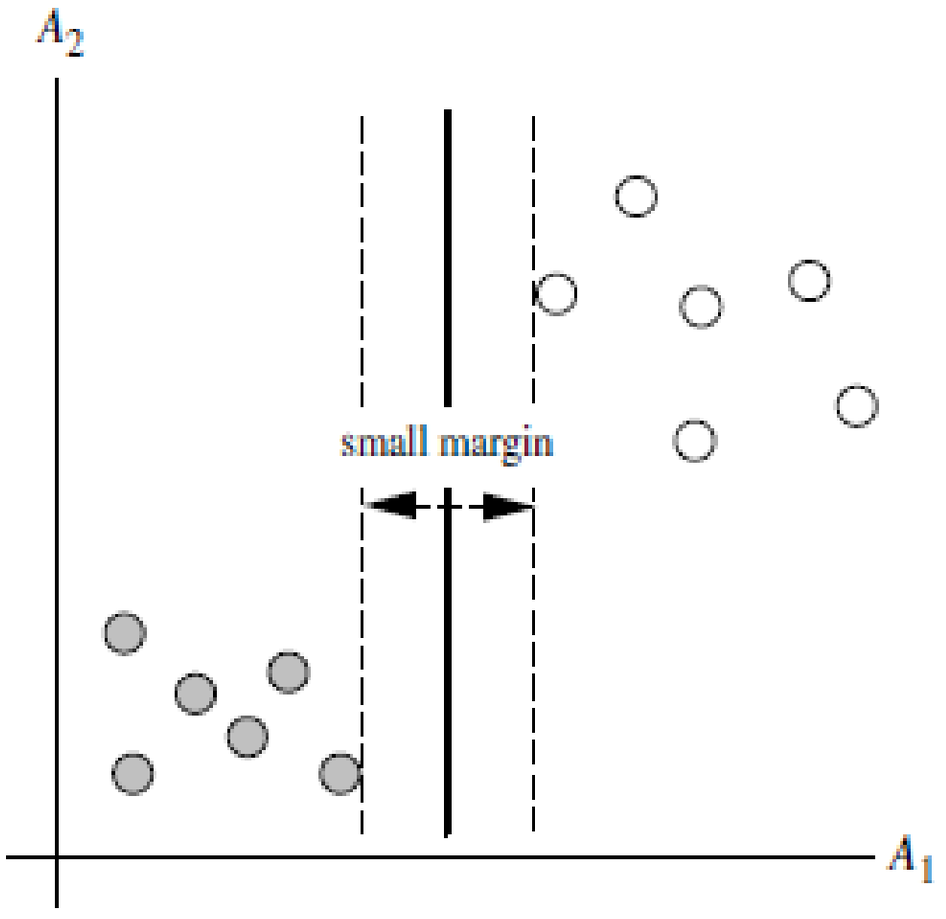
支持向量机(SVM)

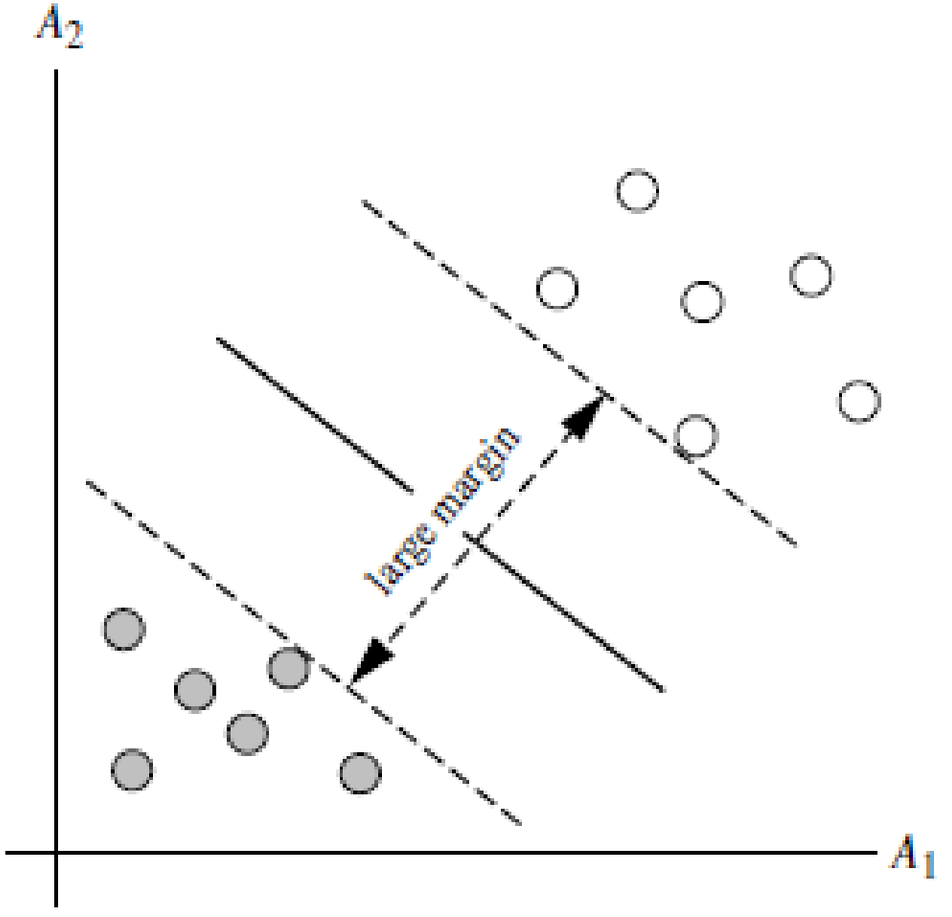
SVM

- 最早是由 Vladimir N. Vapnik 和 Alexey Ya. Chervonenkis 在1963年提出
- 目前的版本(soft margin)是由Corinna Cortes 和 Vapnik在1993年提出，并在1995年发表
- 深度学习（2012）出现之前，SVM被认为机器学习中近十几年来最成功，表现最好的算法

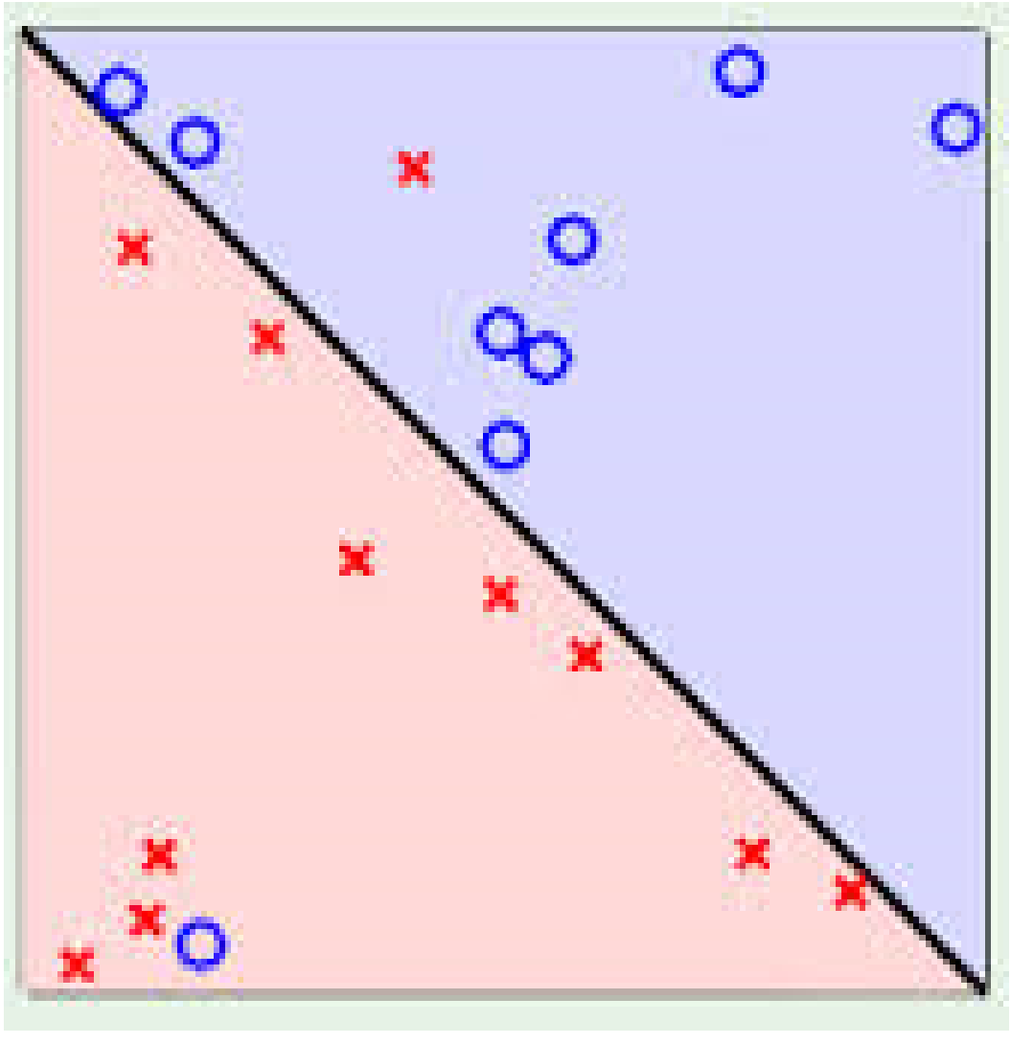
SVM

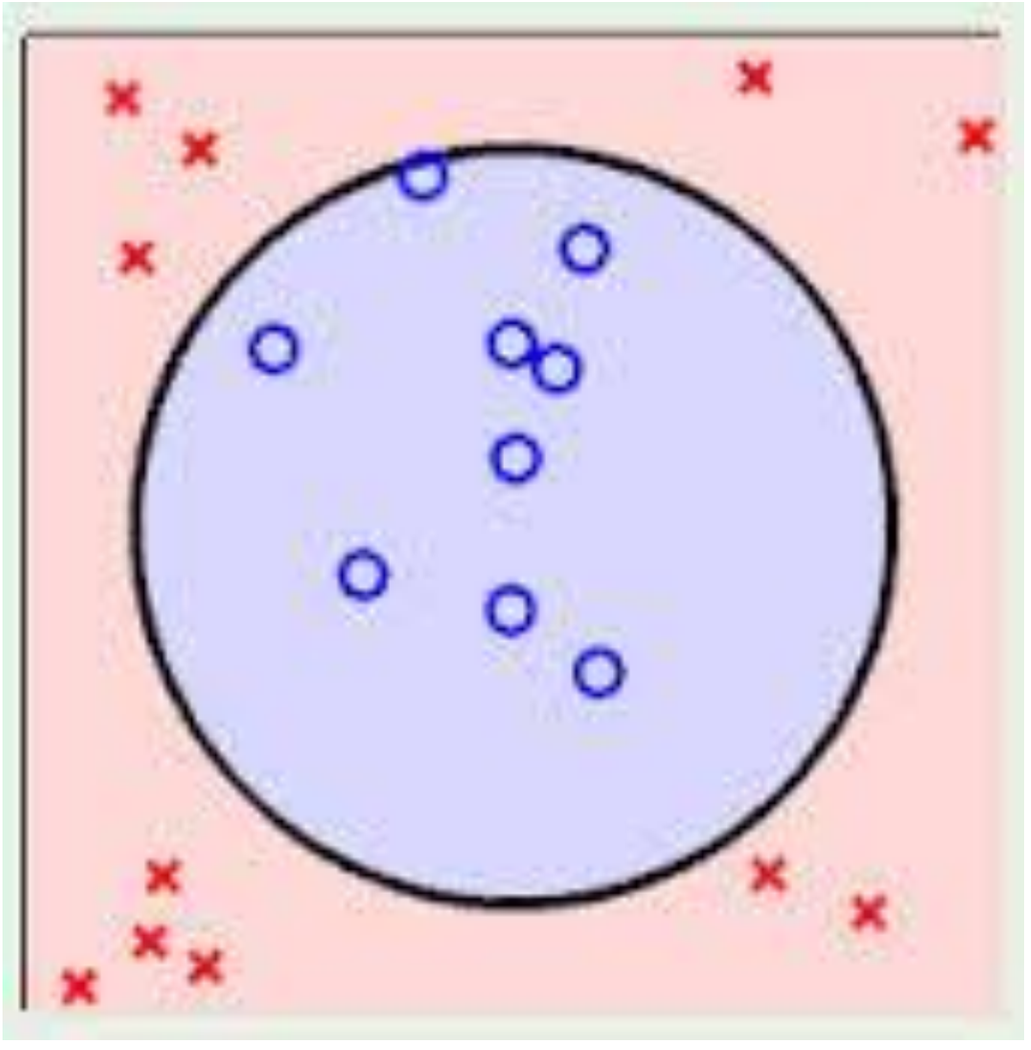
SVM寻找区分两类的超平面（ hyper plane), 使边际(margin)最大



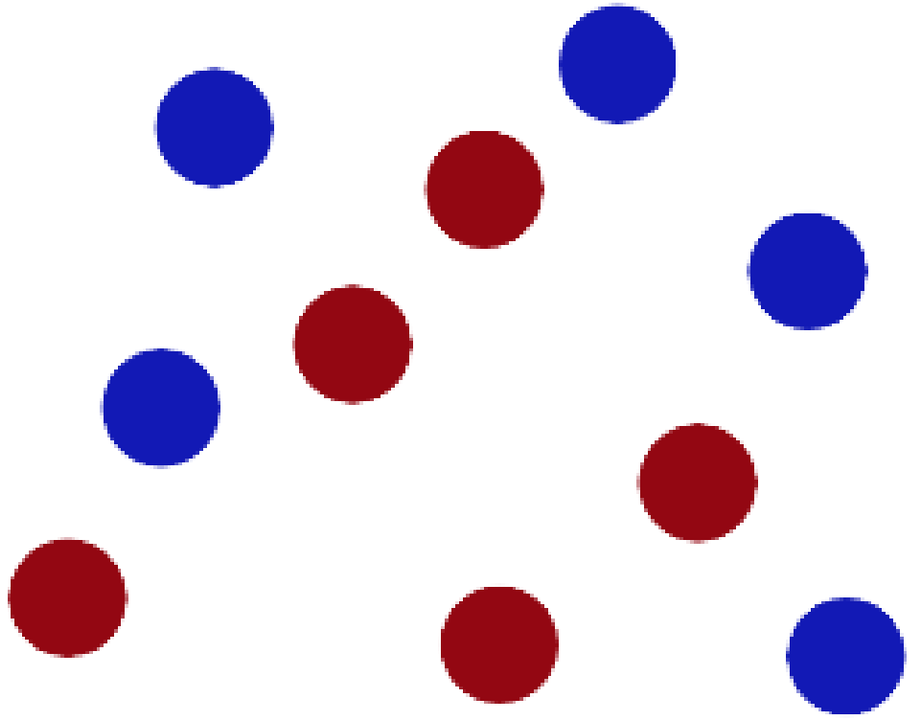


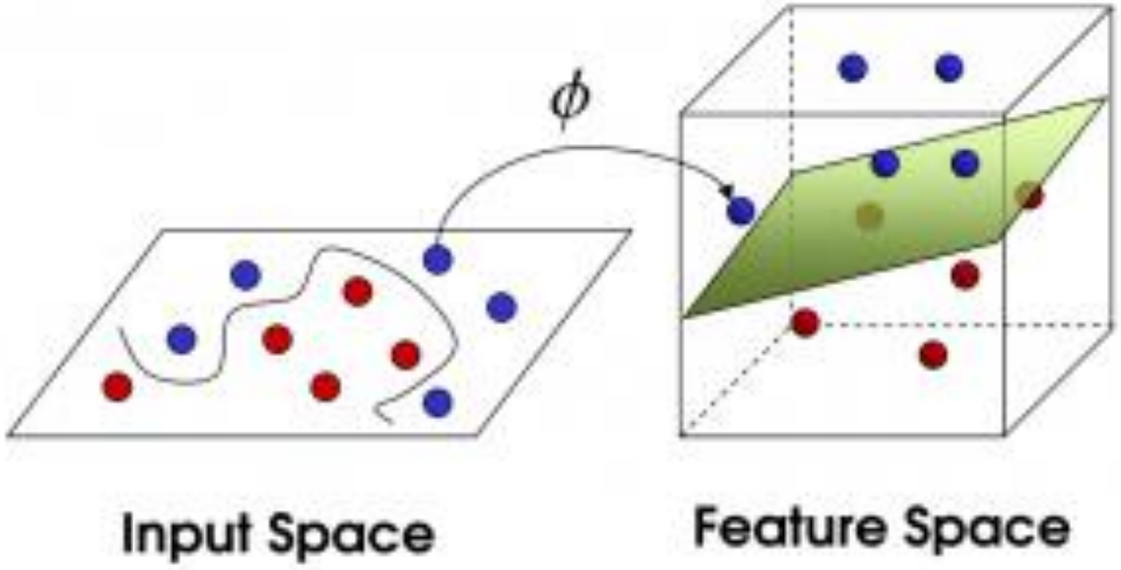
线性不可区分



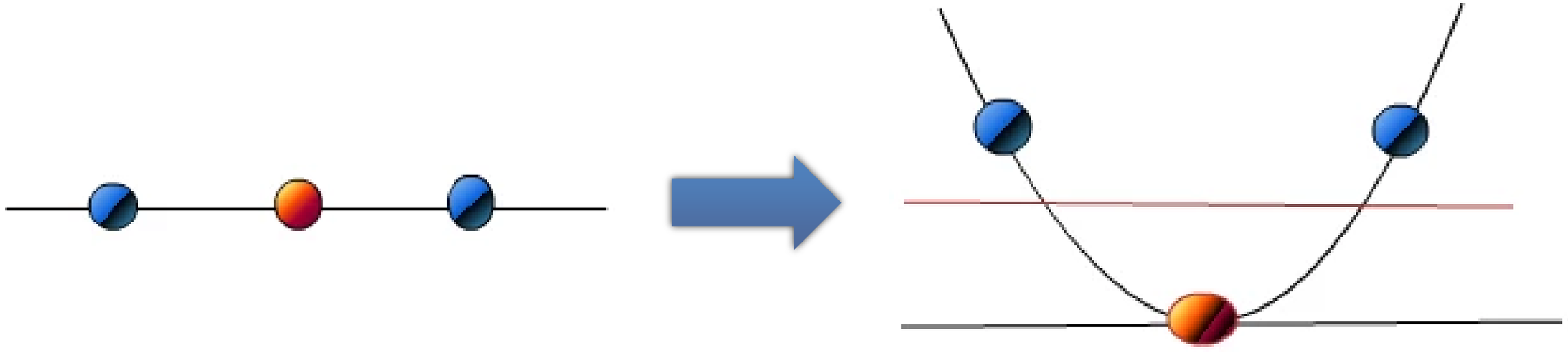


线性不可区分

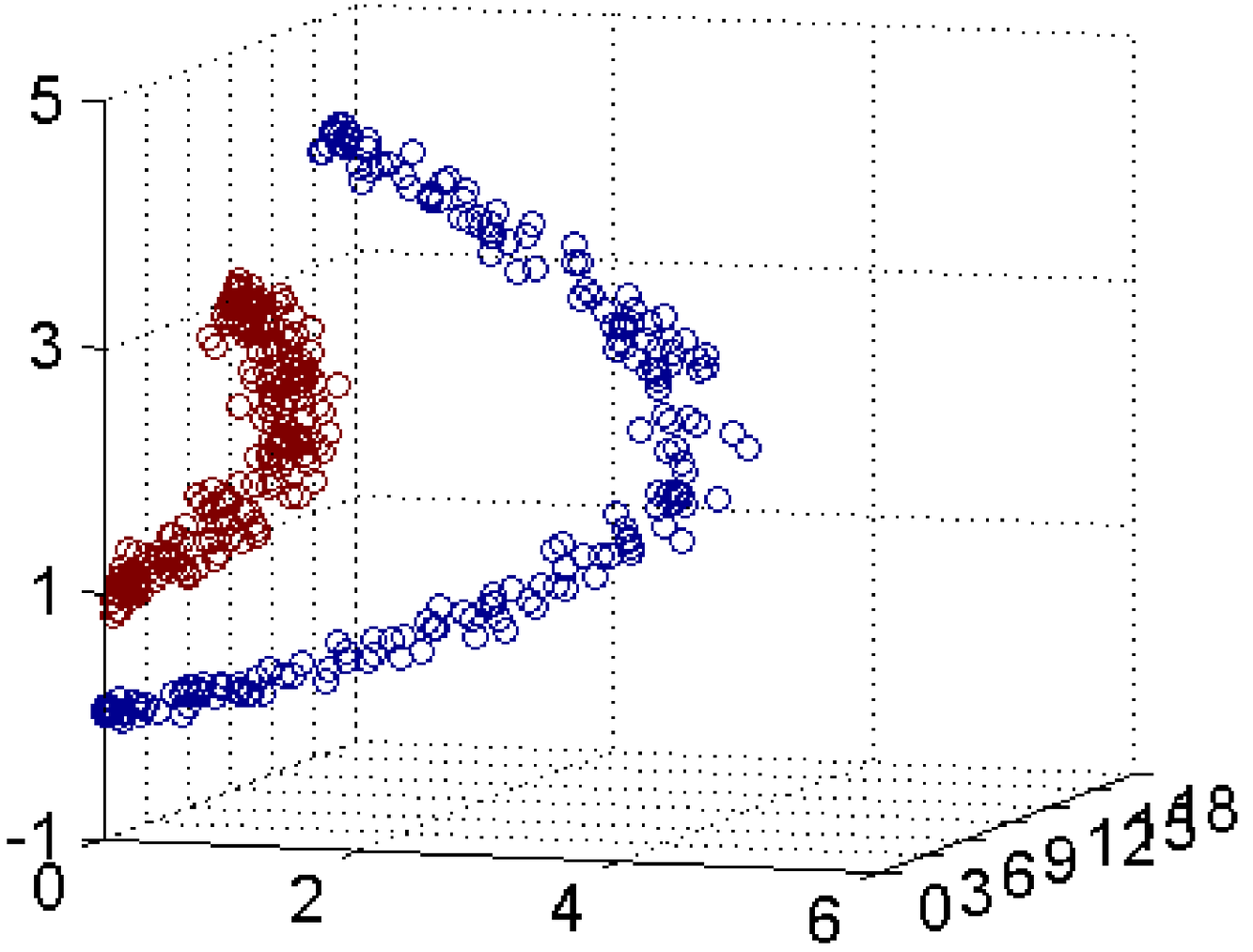




线性不可区分

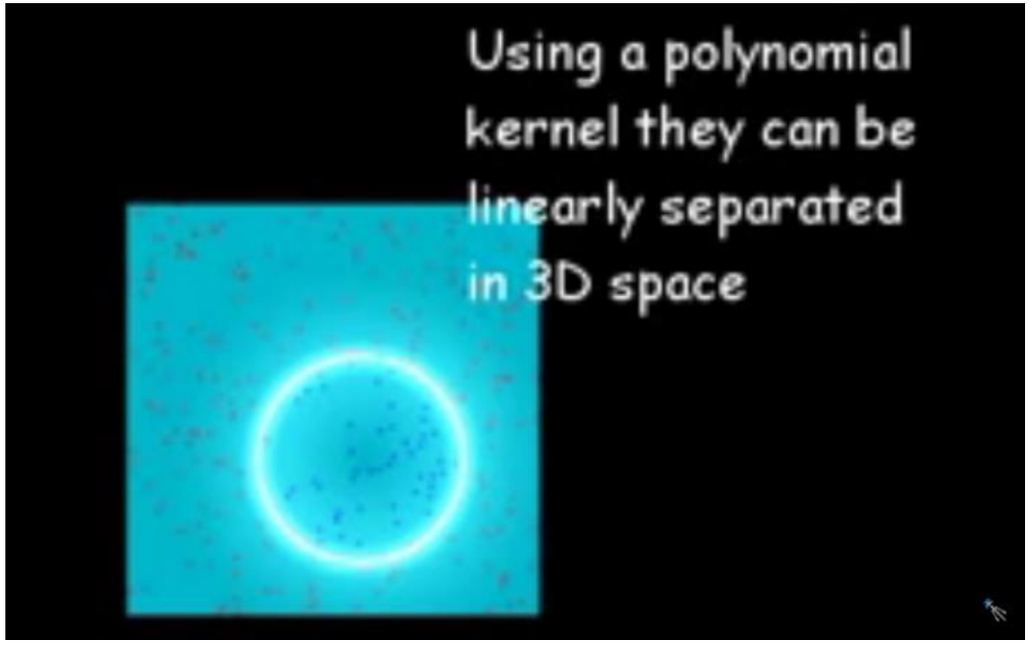


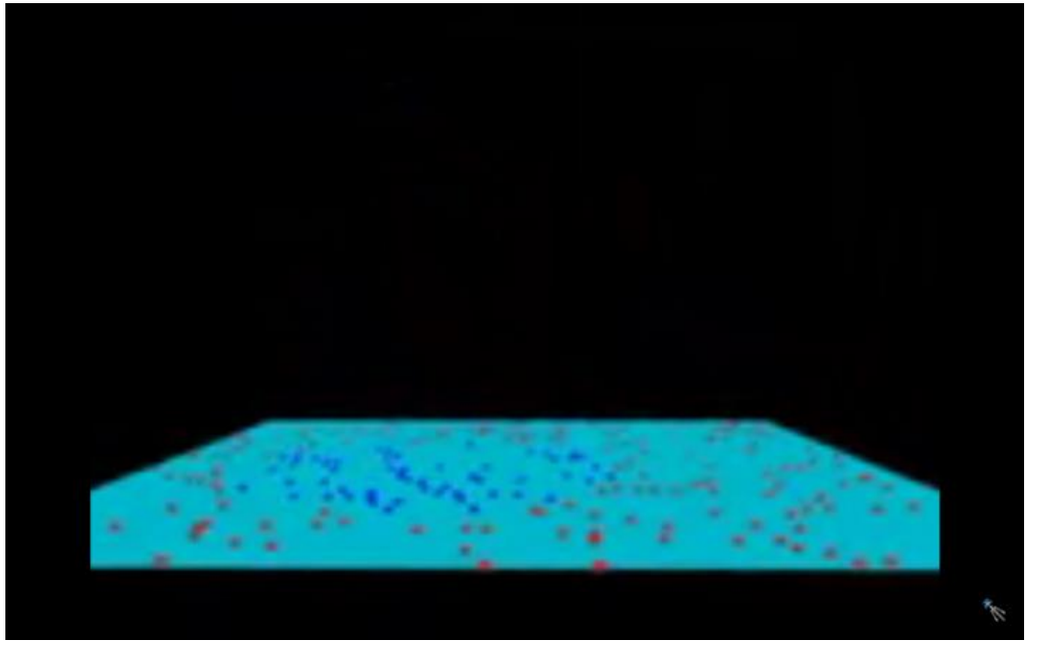
线性不可区分



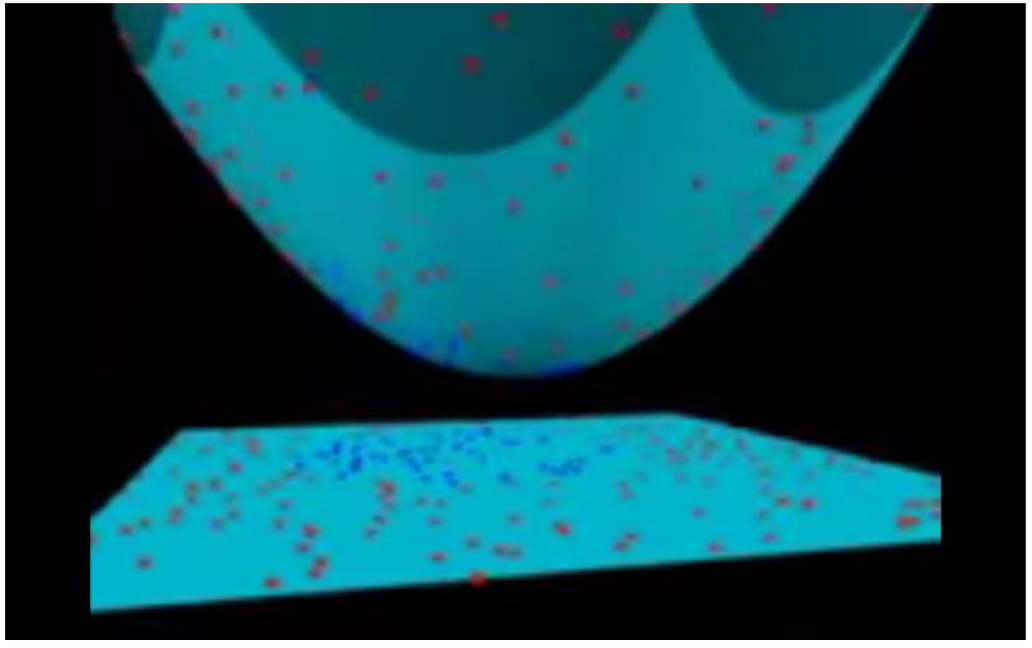
线性不可区分

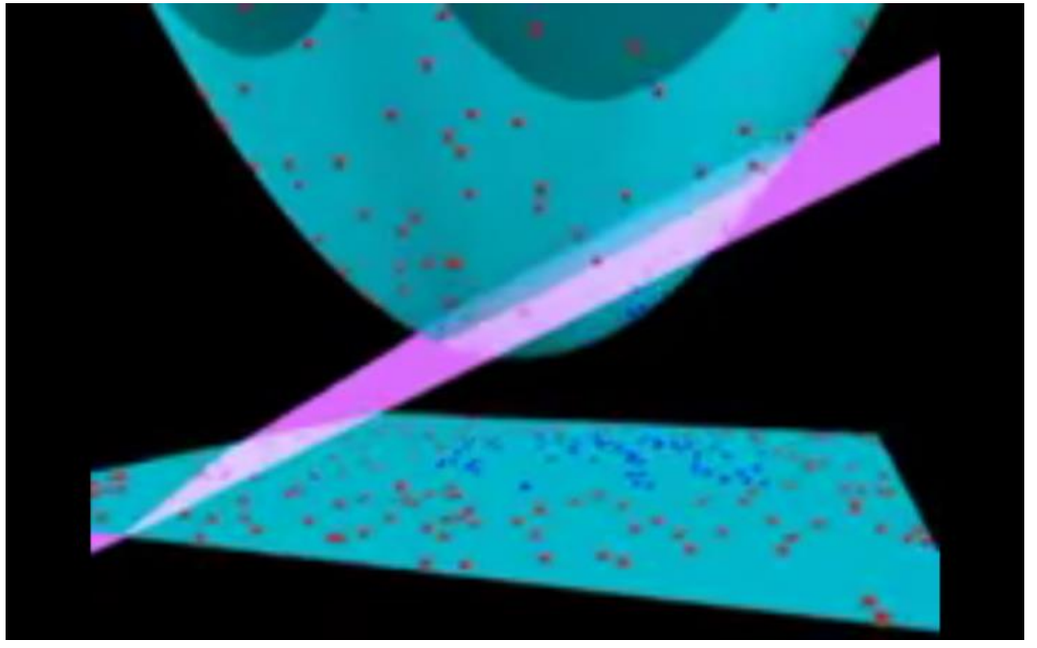
https://v.qq.com/x/page/k05170ntgzc.html



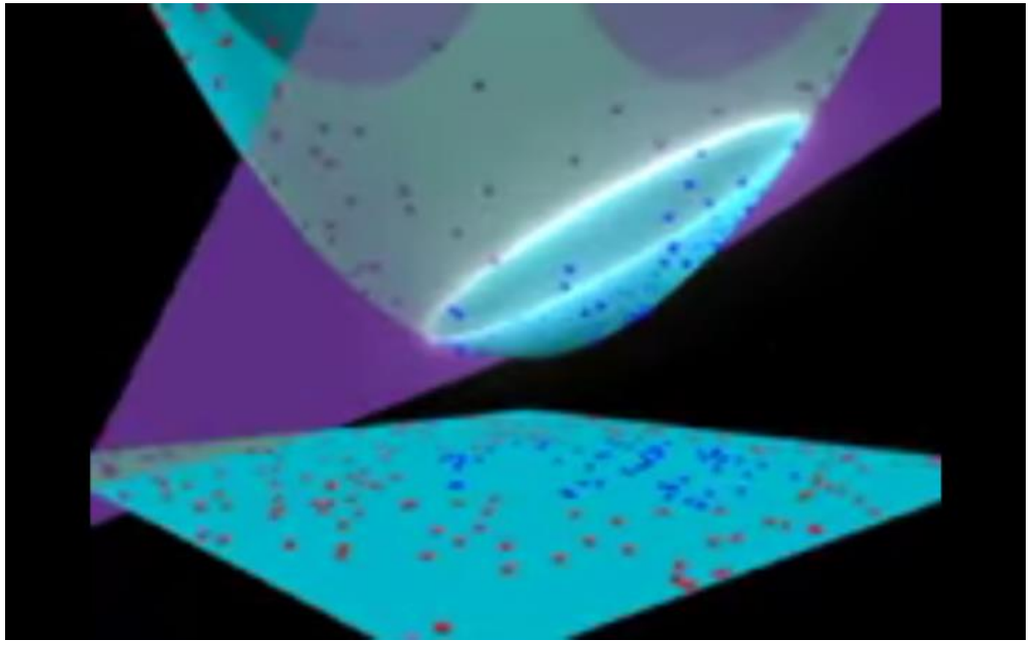


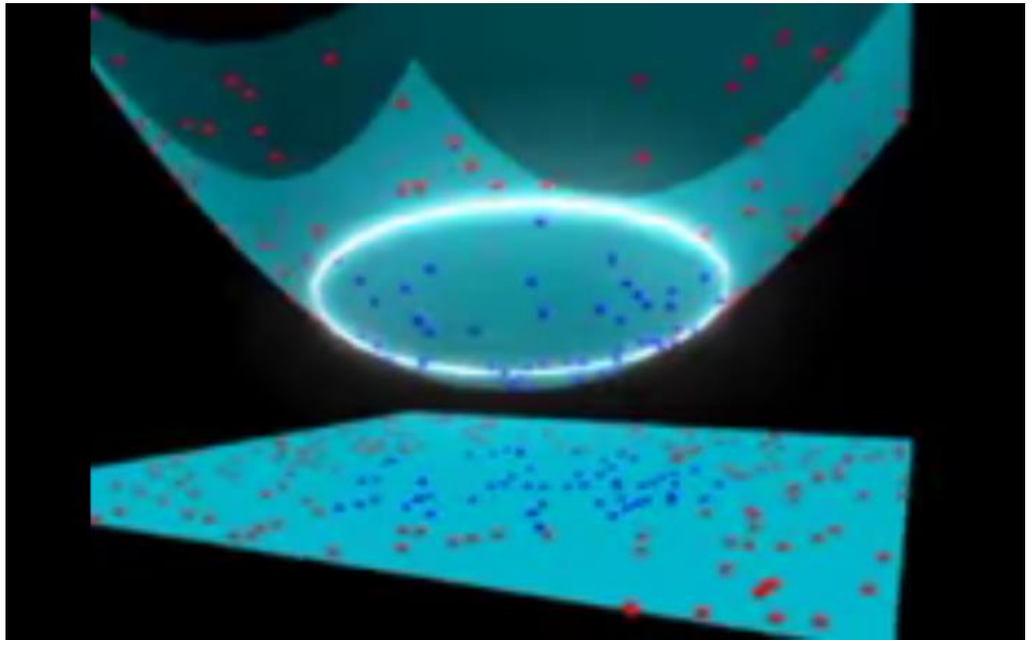
线性不可区分



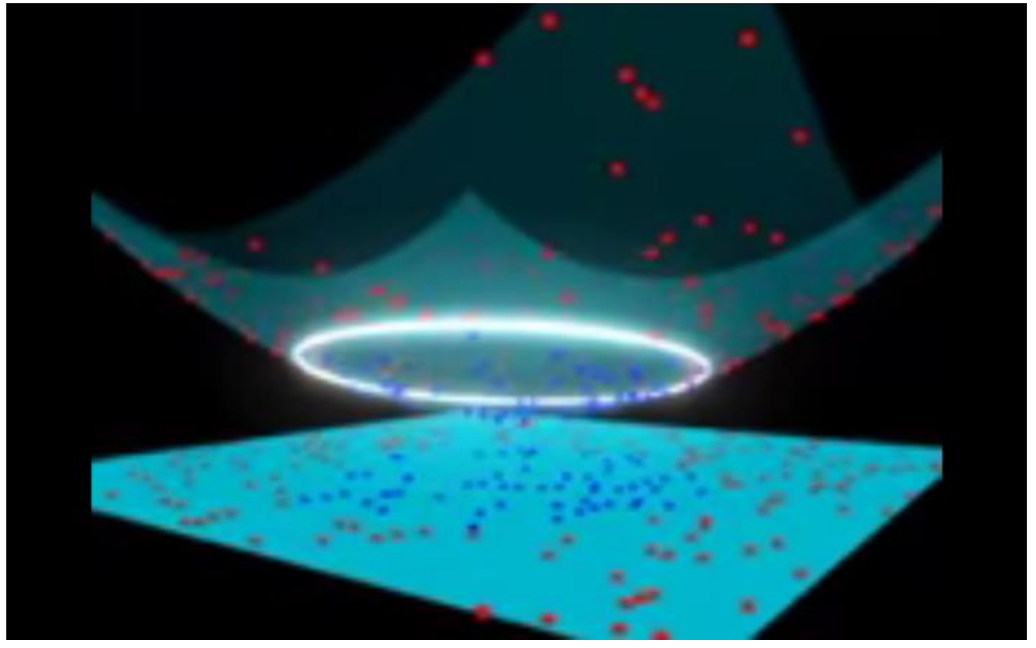


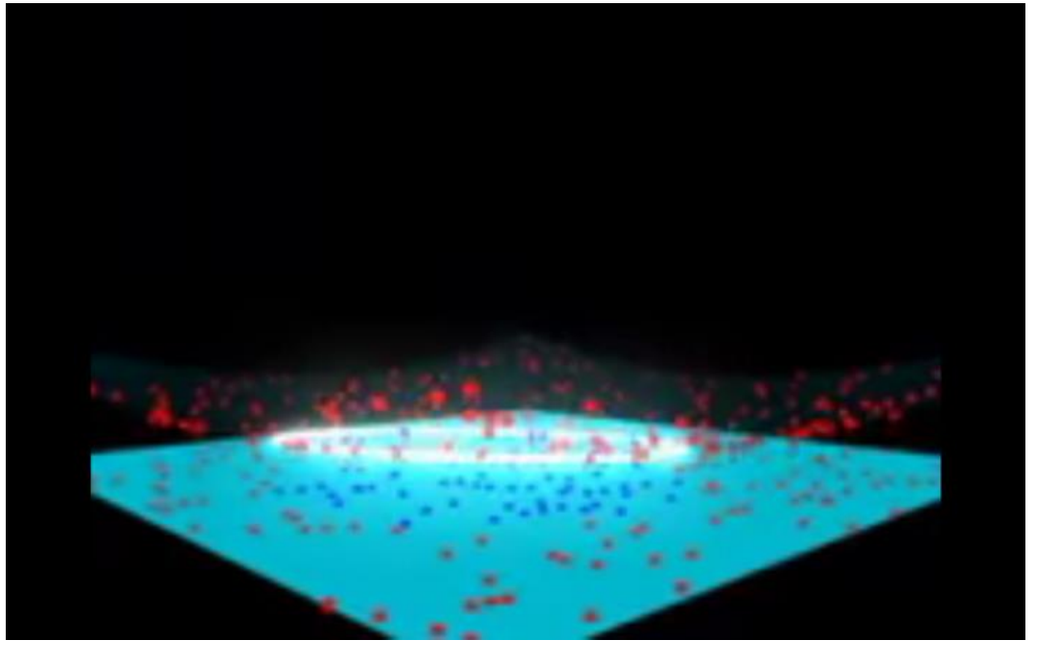
线性不可区分



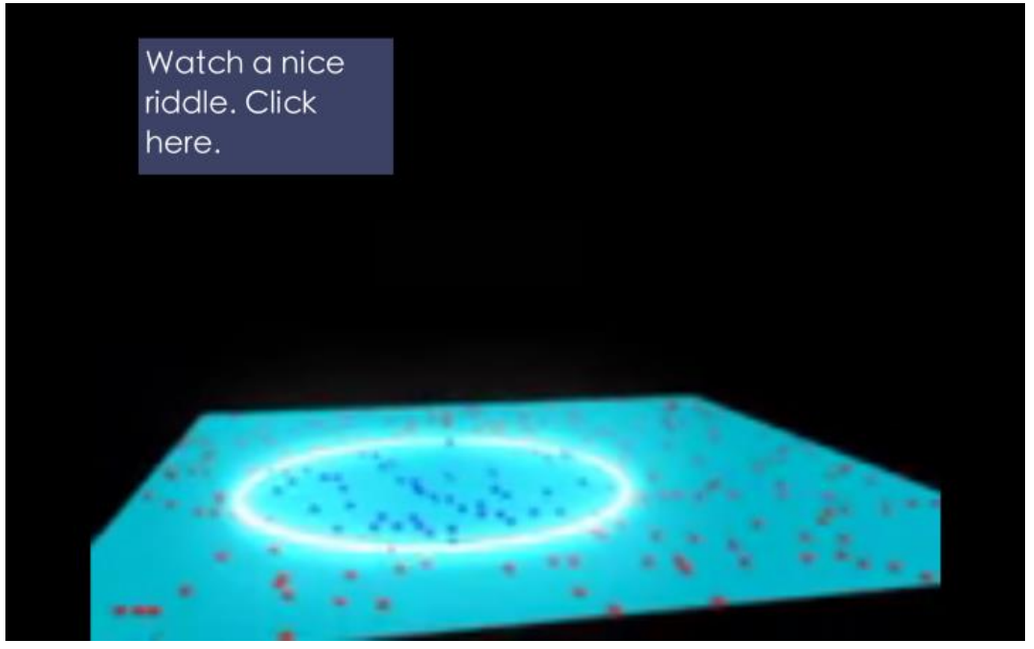


线性不可区分





线性不可区分



Thanks!