人工智能及其航空应用

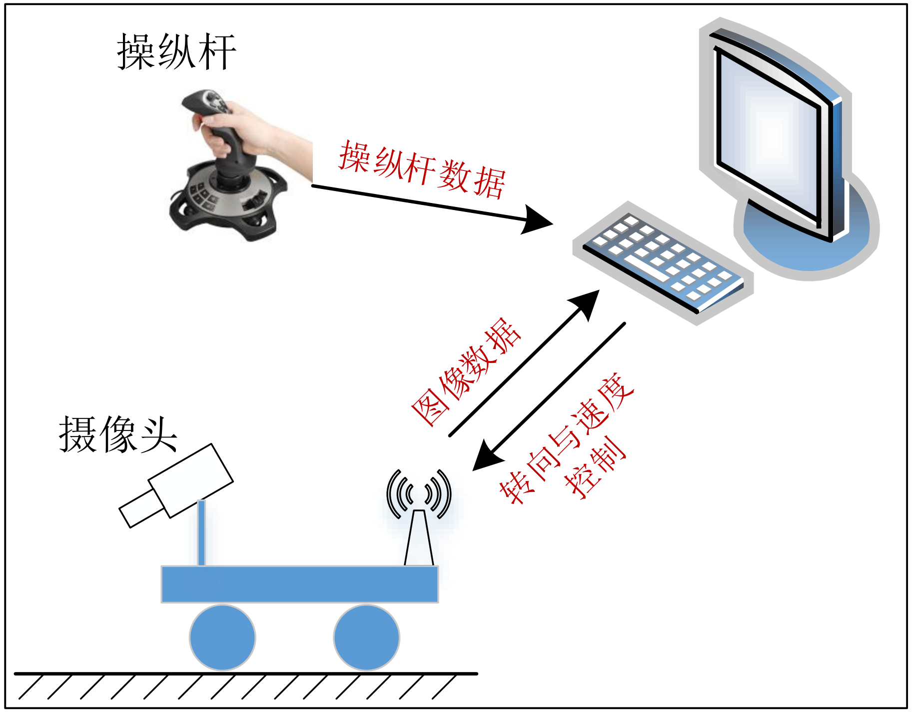
机器学习应用系统设计

项目大作业：飞机模拟滑行

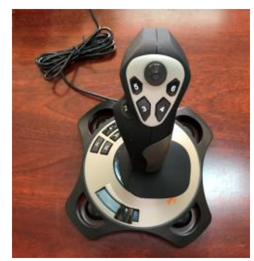
- **项目目标**
  - **获取图像信息，采用人工智能算法分析图像数据，控制转向和速度，使平稳、快速到达终点。**
- **基本要求**
  - **采用人工智能算法**
  - **利用Matlab等编程工具**

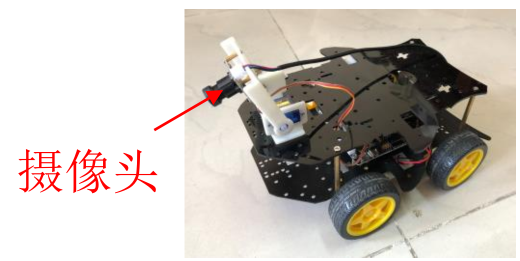
- **考核方式**
  - **根据是否能到达终点、运行速度等确定成绩**

项目大作业：飞机模拟滑行



- 摄像头、操纵杆、树莓派、无线控制
- 基于视觉图像实现车辆方向控制





Matlab相关操作

- 获取图像步骤:
  - ➢ 创建ipcam对象

    cam = ipcam('http://192.168.0.1:8080
  - ➢ 从ipcam对象获取一幅图像

    img=snapshot(cam)
  - ➢ 转换为灰度图像

    grayimg=rgb2gray(img)
  - ➢ 缩放图像

    resizeimg=imresize(grayimg,[20,20])

Matlab相关操作

- 获取操纵杆数据:
  - ➢ 创建操纵杆对象

    joy = vrjoystick(1)　　%相
  - ➢ 获取操纵杆某一轴的值 %

    axis(joy,1)
  - ➢ 获取操纵杆某一按键的值

    button(joy, 1)

Matlab相关操作

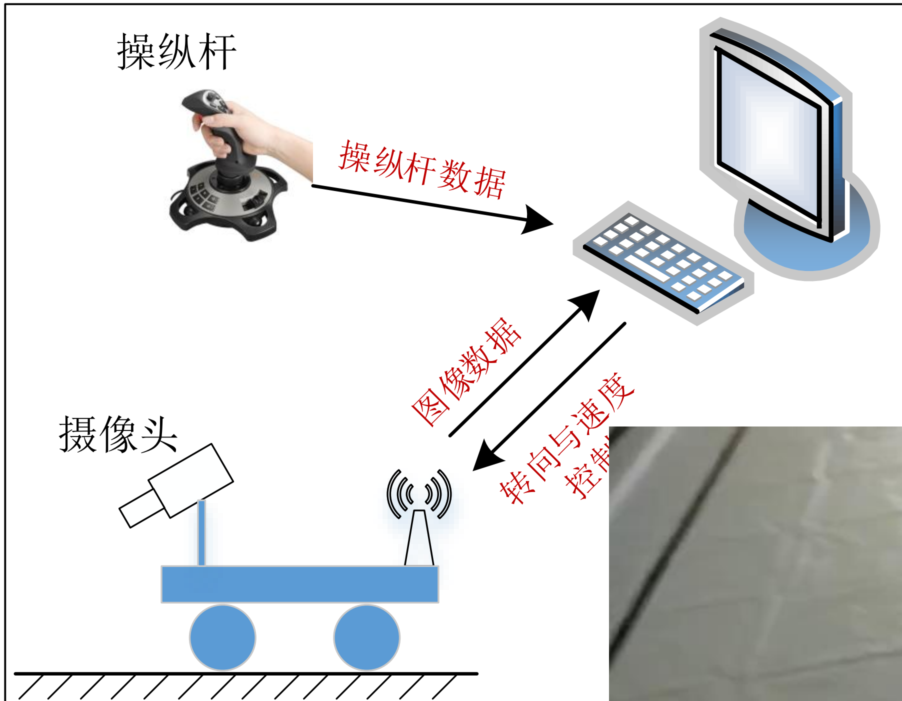
- 发送小车控制指令:
  - ➢ 创建TCP/IP网络通信对象

    control_socket = tcpip('192.168.0.1',
  - ➢ 将TCP/IP对象与小车控制器连接

    fopen(control_socket)
  - ➢ 将指令发送给小车控制器

    fwrite(control_socket, '$0,0,n,d,0,0,0

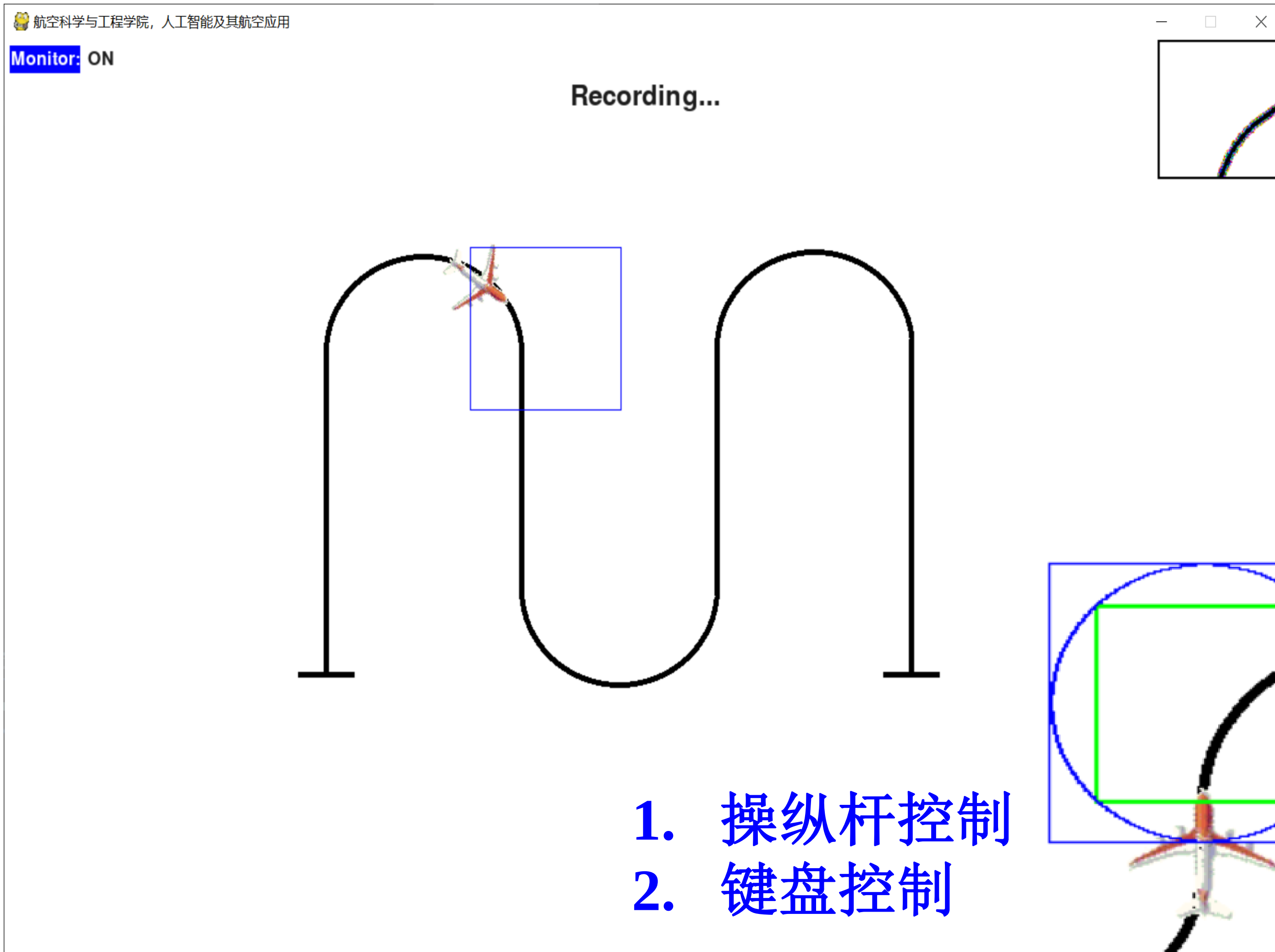
项目大作业：飞机模拟滑行

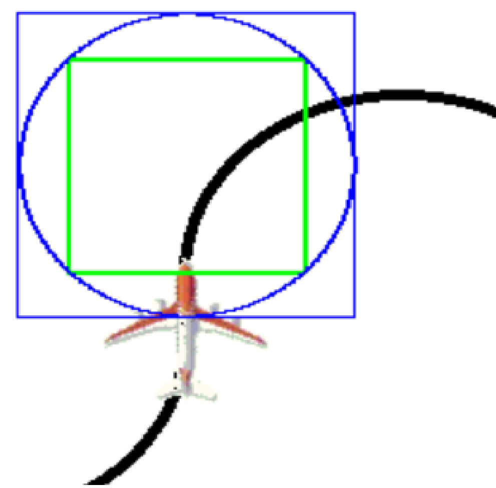


- 摄像头、操纵杆、树莓派、无线控制
- 基于视觉图像实现车辆方向控制

项目大作业：飞机模拟滑行

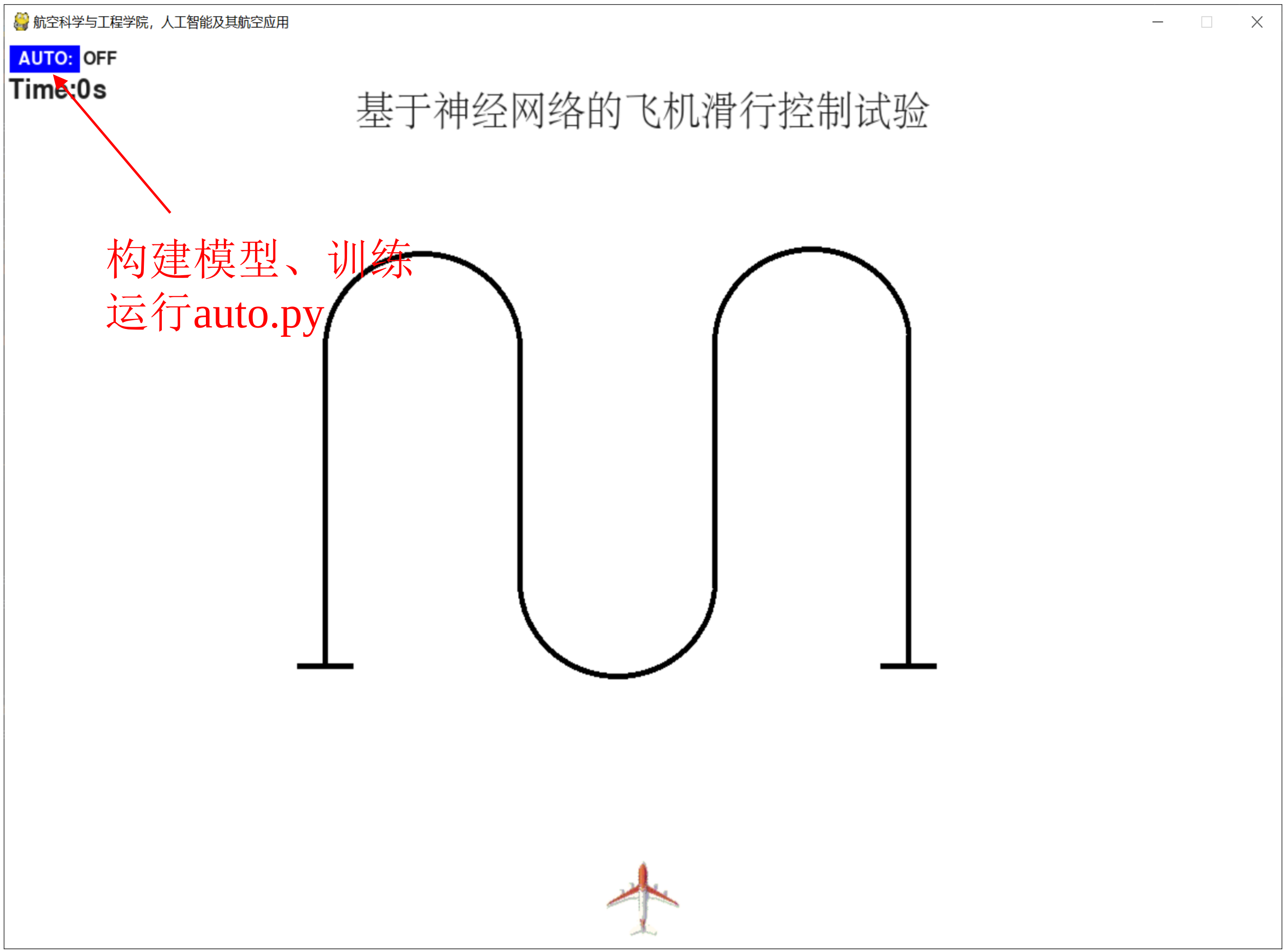
项目大作业：飞机模拟滑行





1. 操纵杆控制
2. 键盘控制

项目大作业：飞机模拟滑行



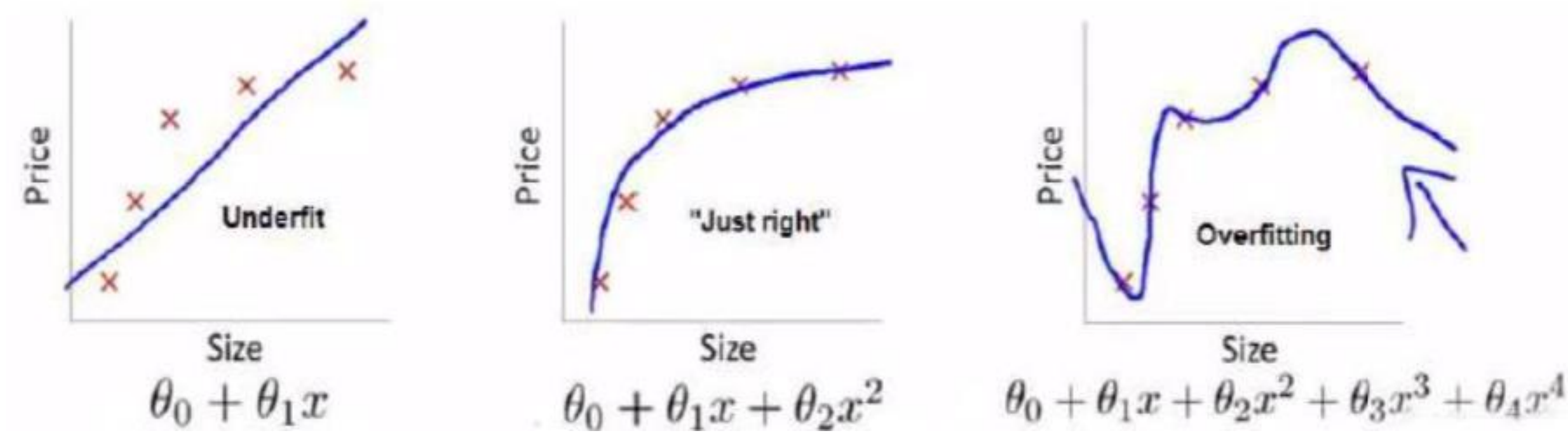
构建模型、训练
运行auto.py

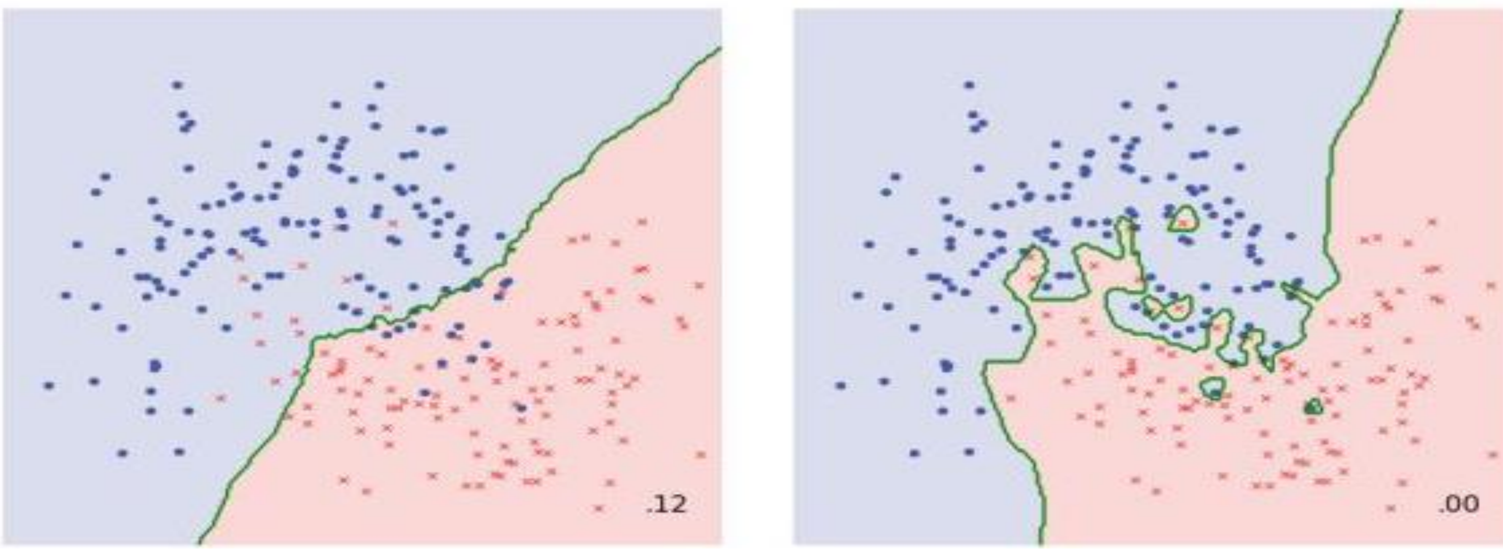
机器学习应用系统设计

如何设计机器学习系统?

如何提升机器学习系统的性能?

问题


Price
Underfit
Size
$\theta_0 + \theta_1 x$
Price
"Just right"
Size
$\theta_0 + \theta_1 x + \theta_2 x^2$
Price
Overfitting
Size
$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$


.12
.00

算法调试与优化方法

- 正则化线性回归方法——泛化预测…

$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda\sum_{j=1}^{m}\theta_j^2\right]$$

方法对新样本的预测偏差较大，
该如何改进算法的预测性能？

- 更多的训练样本
- 选用更少的特征集
- 获取更多特征的角度来收集更多的数据
- 增加多项式特征
- 减小或增大正则化参数

机器学习算法评估

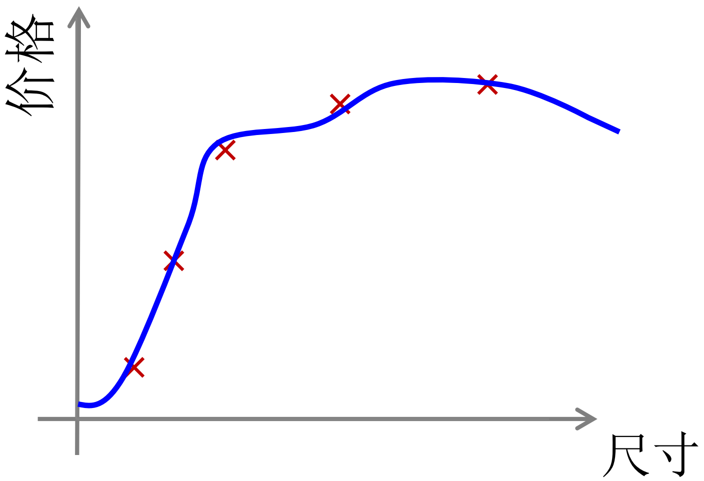
➢ 算法诊断与评估

- 通过执行测试，深入了解学习算法是否有用，以及如何去提升算法的性能。

注意：

算法的诊断需要较长时间，但能指导算法调试的方法。

算法评估



$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

对新的数据集不适用

如何评估算法

数据集：

	尺寸	价格	
70%数据	2104	400	训练集
	1600	330	
	2400	369	
	1416	232	
	3000	540	
	1985	300	
	1534	315	
30%数据	1427	199	测试集
	1380	212	
	1494	243	

训练集 →

$$(x^{(1)}, y^{(1)})$$
$$(x^{(2)}, y^{(2)})$$
$$\vdots$$
$$(x^{(m)}, y^{(m)})$$

测试集 →

$$(x^{(1)}_{test}, y^{(1)}_{test})$$
$$(x^{(2)}_{test}, y^{(2)}_{test})$$
$$\vdots$$
$$(x^{(m_{test})}_{test}, y^{(m_{test})}_{test})$$

训练集和测试集必须要包含各种类型的数据

线性回归模型的训练与测试

- 利用训练集数据计算代价函数J得到参数 θ
  - 通过计算代价函数 $J(\theta)$ 的最小值。

$$J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

- 利用测试数据来计算误差

$$J_{test}(\theta) = \frac{1}{2m_{test}}\sum_{i=1}^{m_{test}}(h_\theta(x_{test}^{(i)}) - y_{test}^{(i)})^2$$

逻辑回归模型的训练与测试

➢ 利用训练集数据计算代价函数J得到参数θ

$$J_{test}(\theta) = -\frac{1}{m_{test}} \sum_{i=1}^{m_{test}} y_{test}^{(i)} \log h_\theta(x_{test}^{(i)}) + (1 - y_{test}^{(i)}) \log h_\theta(x_{test}^{(i)})$$

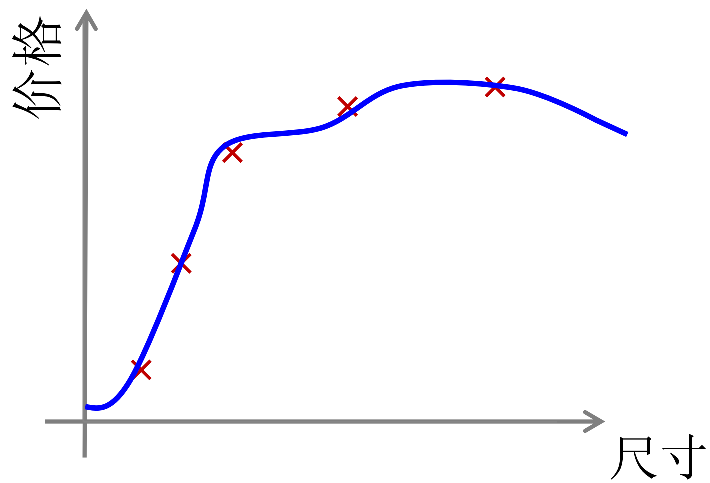
➢ 利用测试数据来计算误差

误分类的比率，对于每一个测试集实例，计算：

$$err(h_\theta(x), y) = \begin{cases} 1 \ if \ h(x) \geq 0.5 \ and \ y = 0, or \ if \ h(x) < 0.5 \ and \ y = 1 \\ 0 \ Otherwise \end{cases}$$

如何选择模型？

过拟合问题



$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

训练集误差小并不说明可很好的拟合新样本

模型选择

- ✓ 1. $h_\theta(x) = \theta_0 + \theta_1 x$
- ✓ 2. $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2$
- ✓ 3. $h_\theta(x) = \theta_0 + \theta_1 x + \cdots + \theta_3 x^3$

$\vdots$

- ✓ 10. $h_\theta(x) = \theta_0 + \theta_1 x + \cdots + \theta_{10} x^{10}$

$J_{test}(\theta)$

哪一个模型能适应一般的情况呢?

使用交叉验证集来帮助选择模型

模型评估

数据集：

	尺寸	价格	
60%数据	2104	400	训练集
	1600	330	
	2400	369	
	1416	232	
	3000	540	
	1985	300	
20%	1534	315	交叉验证集
	1427	199	
20%	1380	212	测试集
	1494	243	

训练集：

$$(x^{(1)}, y^{(1)})$$
$$(x^{(2)}, y^{(2)})$$
$$\vdots$$
$$(x^{(m)}, y^{(m)})$$

交叉验证集：

$$(x_{cv}^{(1)}, y_{cv}^{(1)})$$
$$(x_{cv}^{(2)}, y_{cv}^{(2)})$$
$$\vdots$$
$$(x_{cv}^{(m_{cv})}, y_{cv}^{(m_{cv})})$$

测试集：

$$(x_{test}^{(1)}, y_{test}^{(1)})$$
$$(x_{test}^{(2)}, y_{test}^{(2)})$$
$$\vdots$$
$$(x_{test}^{(m_{test})}, y_{test}^{(m_{test})})$$

模型评估

训练误差：

$$J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

交叉验证误差:

$$J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}(h_\theta(x_{cv}^{(i)}) - y_{cv}^{(i)})^2$$

测试集误差:

$$J_{test}(\theta) = \frac{1}{2m_{test}}\sum_{i=1}^{m_{test}}(h_\theta(x_{test}^{(i)}) - y_{test}^{(i)})^2$$

模型选择

1. $h_\theta(x) = \theta_0 + \theta_1 x \qquad minJ(\theta) \longrightarrow \theta^{(1)} \longrightarrow J_{cv}(\theta^{(1)})$
2. $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 \qquad \longrightarrow \theta^{(2)} \longrightarrow J_{cv}(\theta^{(2)})$
3. $h_\theta(x) = \theta_0 + \theta_1 x + \cdots + \theta_3 x^3$

$\vdots$

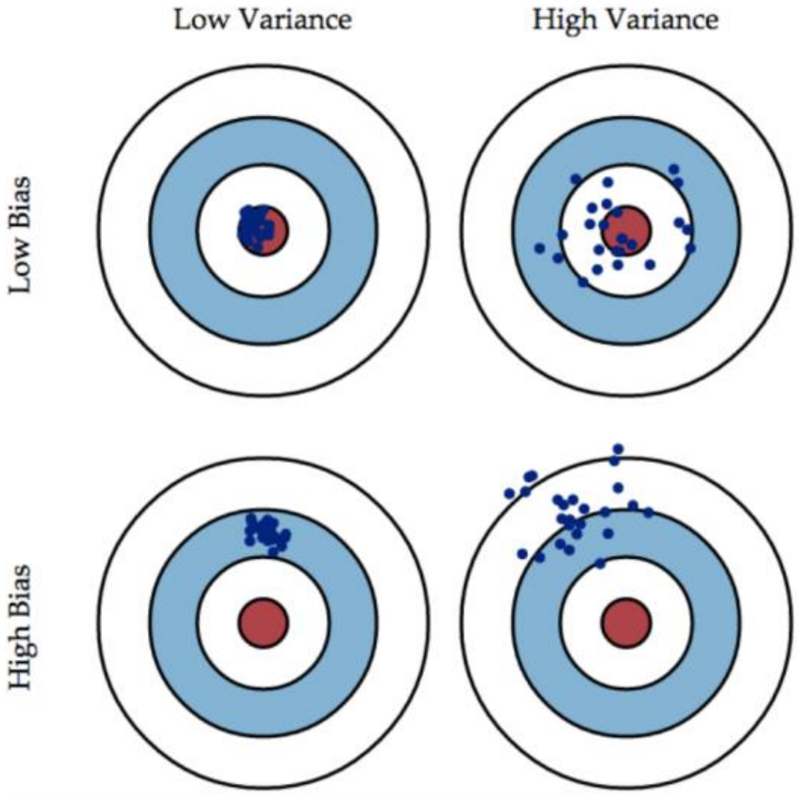
10. $h_\theta(x) = \theta_0 + \theta_1 x + \cdots + \theta_{10} x^{10} \qquad \longrightarrow \theta^{(10)} \longrightarrow J_{cv}(\theta^{(10)})$

- 根据Jcv值选择较小的模型
- 测试模型在测试集上的误差

模型偏差、方差评估

➢ 偏差

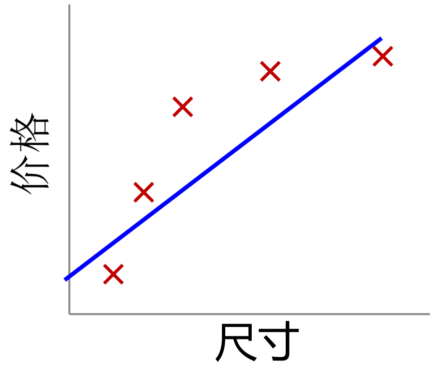
- 期望输出与真实标记的差别称为偏差，度量了学习算法的期望预测与真实结果的偏离程度，刻画了学习算法本身的拟合能力。



➢ 方差

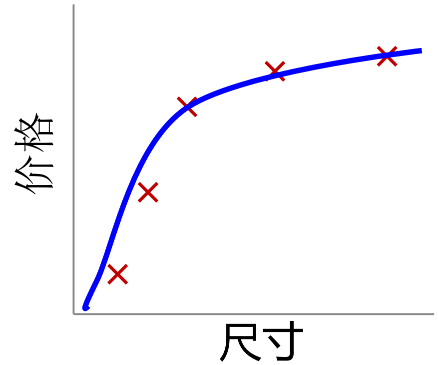
- 方差度量了同样大小的训练集(样本数相同的不同训练集)的变动所导致的学习性能的变化，即刻画了数据扰动所造成的影响。

偏差与方差



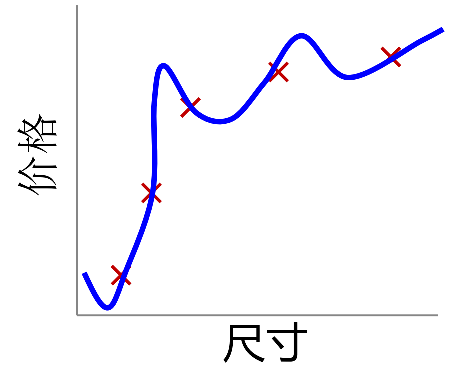
$\theta_0 + \theta_1 x$

高偏差（欠拟合）



$\theta_0 + \theta_1 x + \theta_2 x^2$

最佳



$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

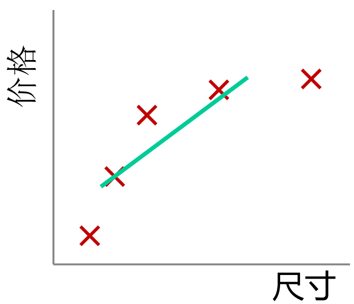
高方差（过拟合）

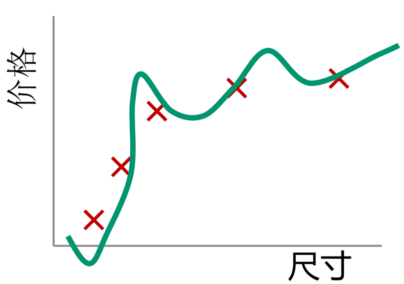
一般来说，简单的模型会有一个较大的偏差和较小的方差，复杂的模型偏差较小方差较大

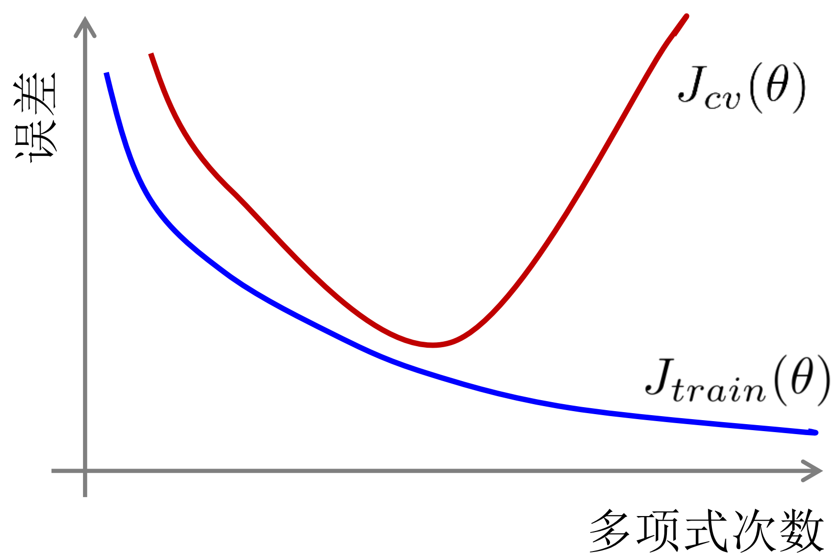
模型评估：偏差与方差

训练误差: $J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$

交叉验证误差: $J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}(h_\theta(x_{cv}^{(i)}) - y_{cv}^{(i)})^2$

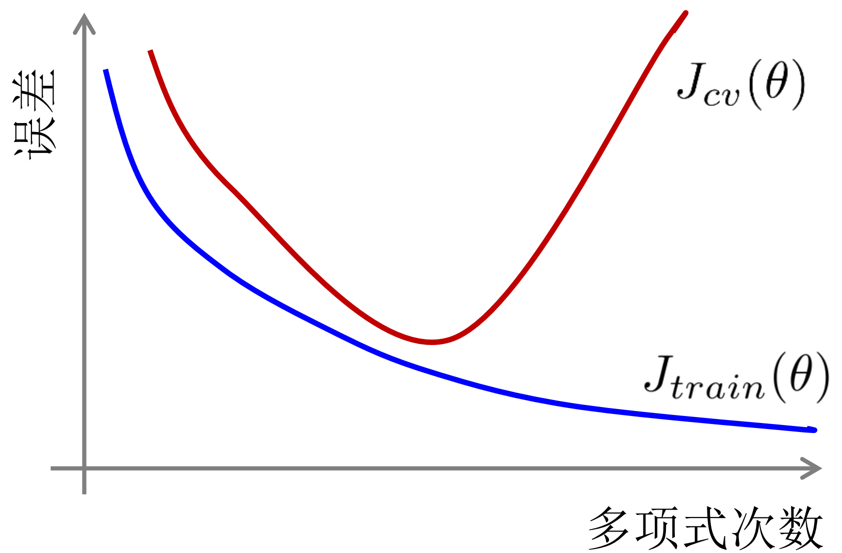






模型评估：偏差与方差

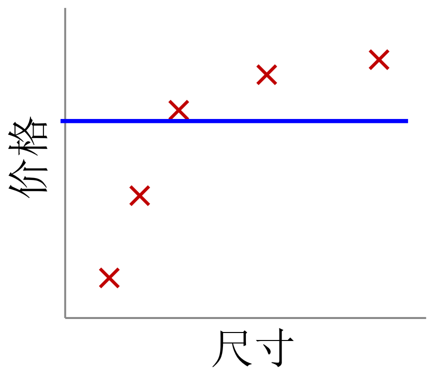
- 对于训练集，当d 较小时，模型误差较大，拟合程度低；随着d 的增长，误差减小，拟合程度提高。
- 对于交叉验证集，当d 较小时，模型误差较大，拟合程度低;但是随着d 的增长，误差呈现先减小后增大的趋势，转折点是我们的模型开始过拟合时。



正则化线性回归

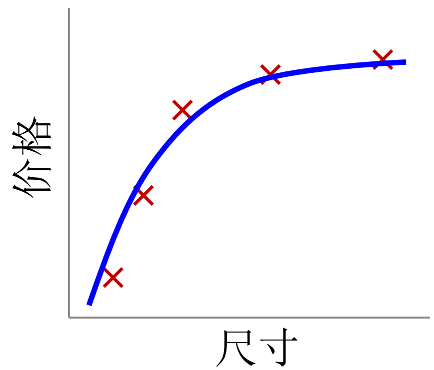
模型： $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \frac{\lambda}{2m}\sum_{j=1}^{m}\theta_j^2$$

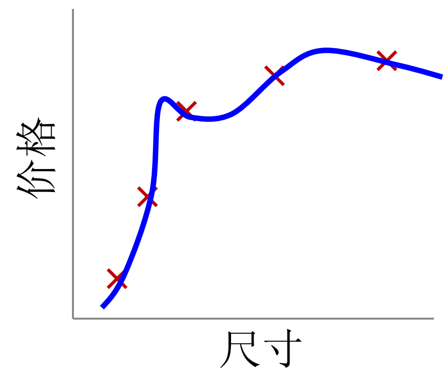


较大的 λ
高偏差(欠拟合)

$\lambda = 10000.\ \theta_1 \approx 0, \theta_2 \approx 0, \ldots$
$h_\theta(x) \approx \theta_0$



适中的 λ



较小的λ
高方差(过拟合)

λ =0

选择正则化参数

模型：$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \frac{\lambda}{2m}\sum_{j=1}^{m}\theta_j^2$$

1. 测试 $\lambda = 0$
2. 测试 $\lambda = 0.01$
3. 测试 $\lambda = 0.02$
4. 测试 $\lambda = 0.04$
5. 测试 $\lambda = 0.08$

$\vdots$

12. 测试 $\lambda = 10$

选择正则化参数

$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \frac{\lambda}{2m}\sum_{j=1}^{m}\theta_j^2$$

$$J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

$$J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}(h_\theta(x_{cv}^{(i)}) - y_{cv}^{(i)})^2$$

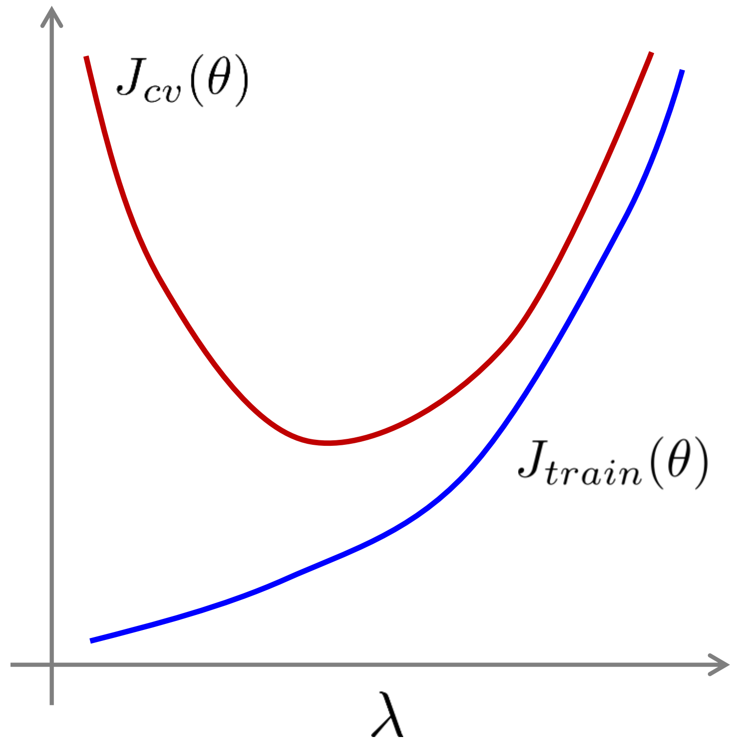
$$J_{test}(\theta) = \frac{1}{2m_{test}}\sum_{i=1}^{m_{test}}(h_\theta(x_{test}^{(i)}) - y_{test}^{(i)})^2$$

选择正则化参数

$$J(\theta)=\frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)})-y^{(i)})^2+\frac{\lambda}{2m}\sum_{j=1}^{m}\theta_j^2$$

$$J_{train}(\theta)=\frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)})-y^{(i)})^2$$

$$J_{cv}(\theta)=\frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}(h_\theta(x_{cv}^{(i)})-y_{cv}^{(i)})^2$$



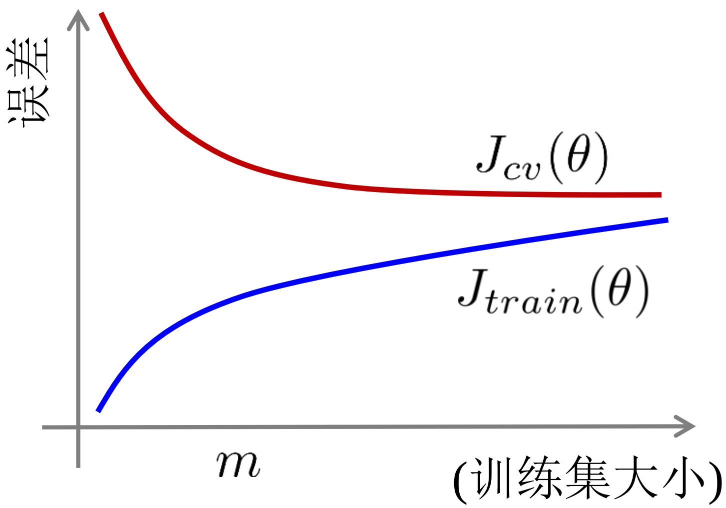
- 当 λ 较小时，训练集误差较小(过拟合)而交叉验证集误差较大
- 随着 λ 的增加，训练集误差不断增加(欠拟合)，而交叉验证集误差则是先减小后增加

利用学习曲线判断偏差、方差问题

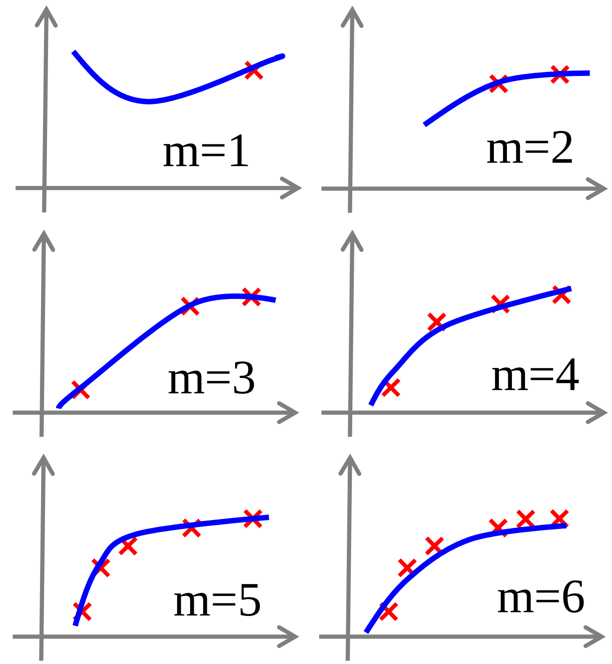
学习曲线：

$$J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

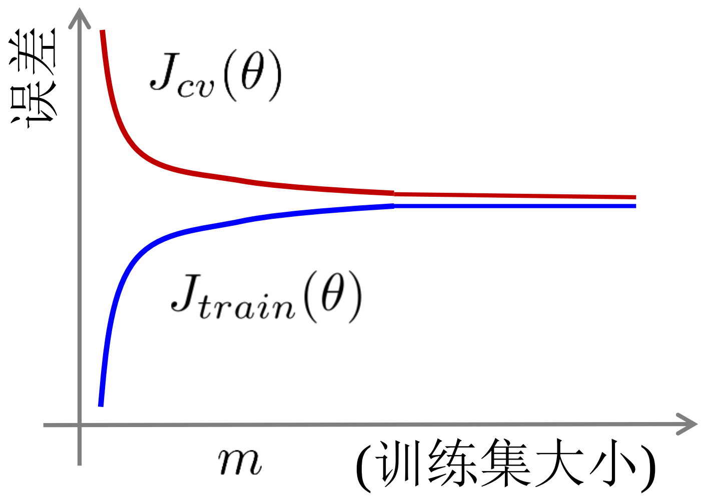
$$J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}(h_\theta(x_{cv}^{(i)}) - y_{cv}^{(i)})^2$$



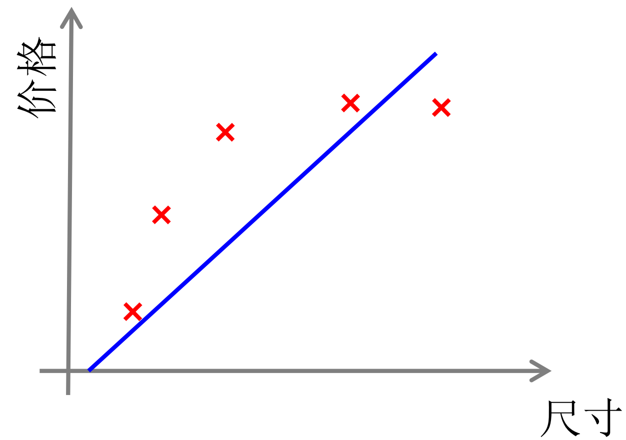
$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2$$

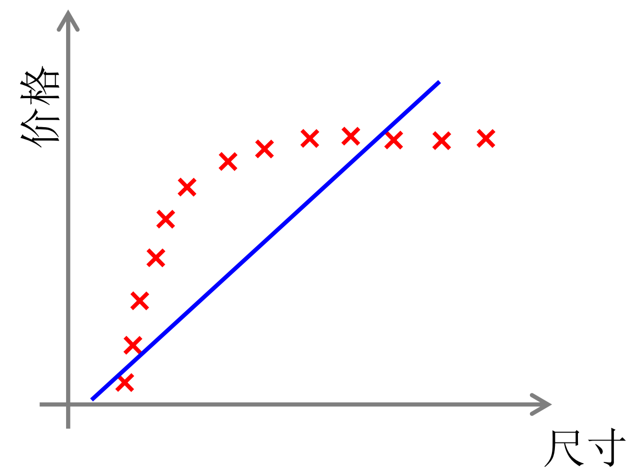


高偏差



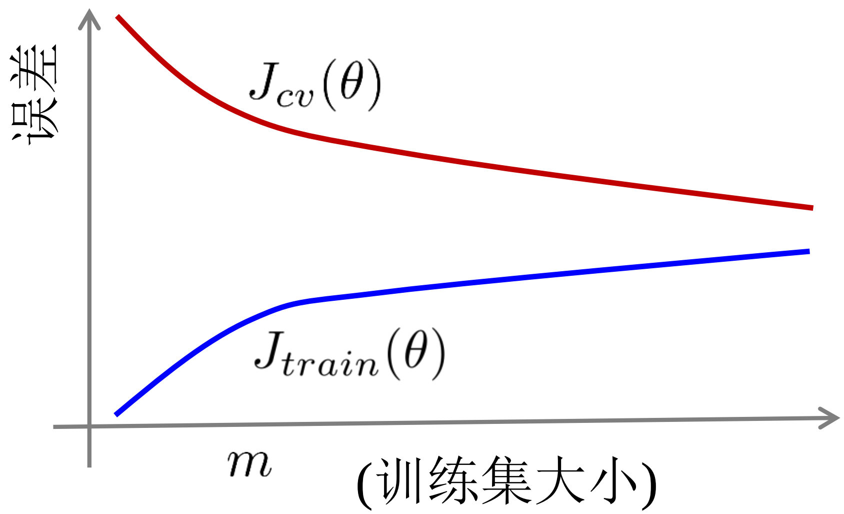
$$h_\theta(x) = \theta_0 + \theta_1 x$$



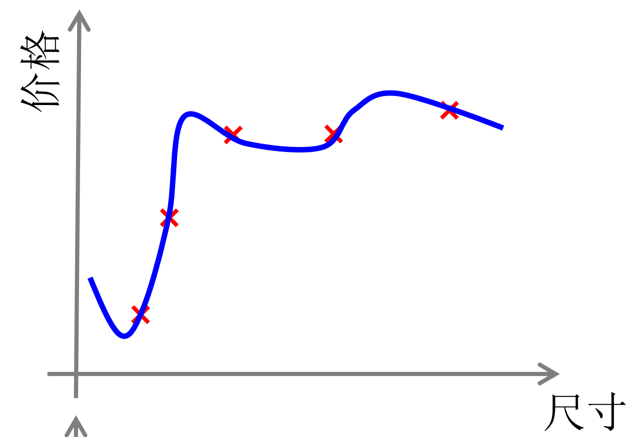


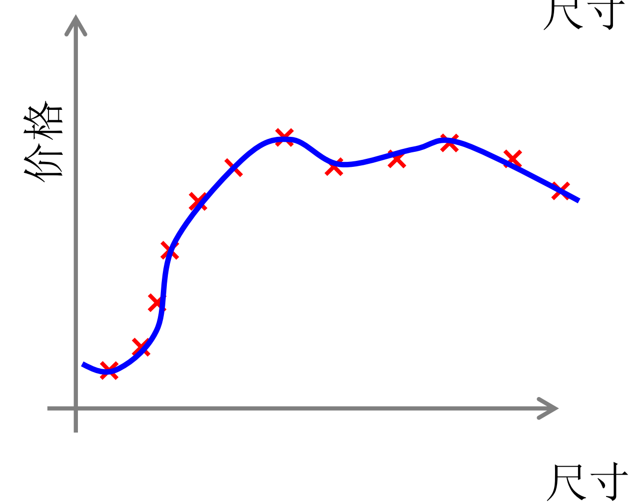
在高偏差/欠拟合的情况下，增加数据到训练集对减小误差没有太大作用

高方差



$$h_\theta(x) = \theta_0 + \theta_1 x + \cdots + \theta_{100} x^{100}$$





当交叉验证集误差远大于训练集误差时，往训练集增加更多数据可以提高模型的效果

小结

- 采用已由正则化线性回归方法预测

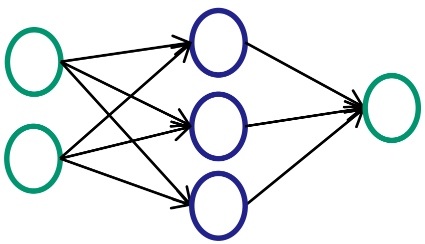
$$J(\theta)=\frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)})-y^{(i)})^2+\lambda\sum_{j=1}^{m}\theta_j^2\right]$$

对新样本的偏差较大，如何改进算法的预测性能？

- 更多的训练实例——解决高方差
- 减少特征数量——解决高方差
- 获得更多特征——解决高偏差
- 增加多项式特征——解决高偏差
- 减少正则化参数 λ——解决高偏差
- 增加正则化参数 λ——解决高方差

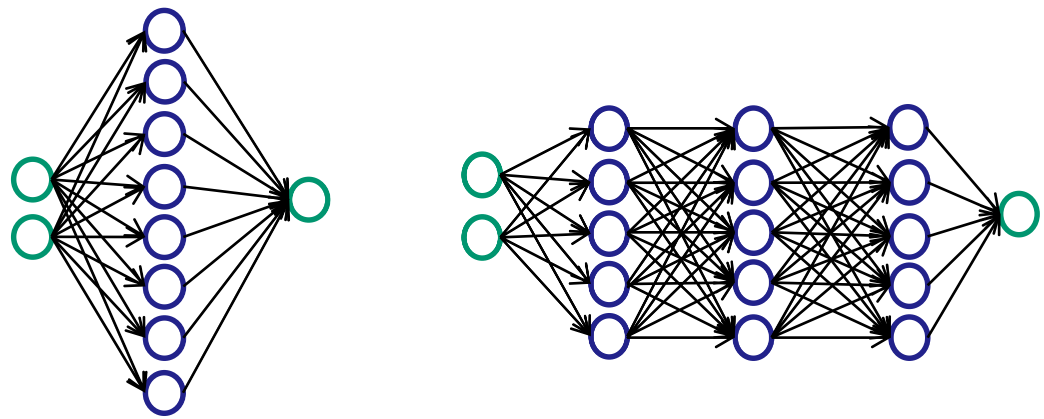
神经网络的方差和偏差

较小的神经网络
(结构简单、参数少；倾向于欠拟合)



计算成本较低

较大的神经网络
(结构复杂、参数多; 倾向于过拟合)



计算成本较高
使用正则化参数解决过拟合问题.

机器学习应用系统设计-偏斜类误差度量

- 偏斜类情况
  - 训练集中有非常多的同一种类的实例，只有很少或没有其他类的实例

账号异常（盗用）的例子

- 训练逻辑回归模型 $h_\theta(x)$，为1表示异常，为0为正常。
- 只有0.1%的账号有异常

算法1：神经网络训练

误差：1%

算法2: 对任何情况均预测0

误差：0.1%

对于偏斜类情况，需要有不同的误差度量方法

准确率/召回率（查全率）

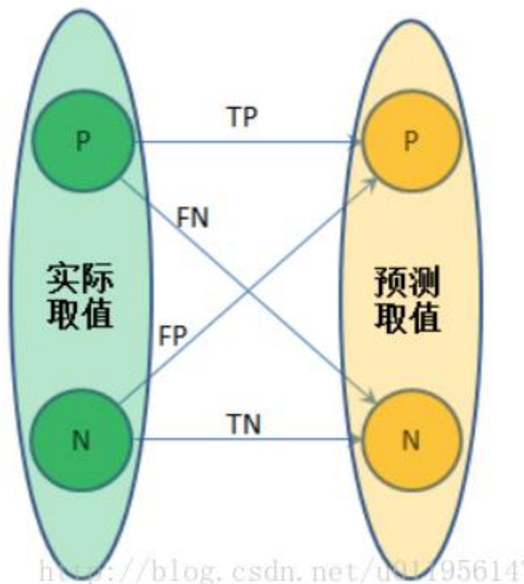
- 设数据集中有较少Y=1的样本，这些样本是需要检出的。

- **准确率**
  - 准确率 P 是指判断为正确的样本中，实际为正确样本的比率：

    P = TP / (TP + FP)

- **召回率**
  - 召回率 R 是指所有正例样本中预测为正例样本的比例：

    R = TP / (TP + FN)



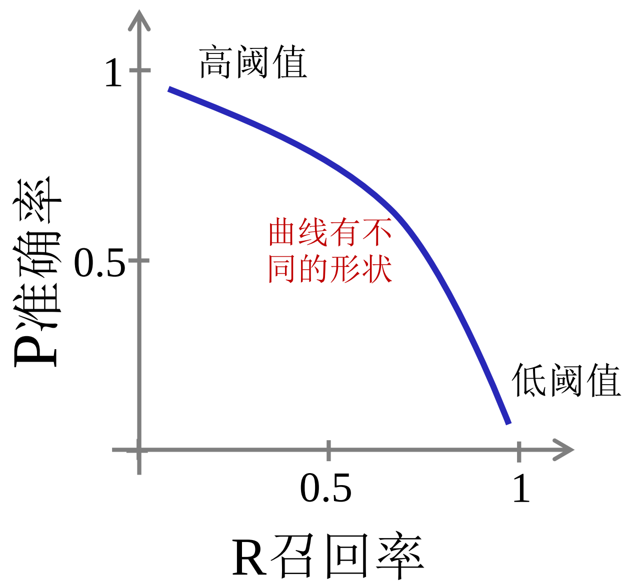
True Positive （真正, TP）	被模型预测为正的正样本
True Negative （真负 , TN）	被模型预测为负的负样本
False Positive （假正, FP）	被模型预测为正的负样本
False Negative （假负 , FN）	被模型预测为负的正样本

准确率/召回率（查全率）

- **准确率/召回率的权衡**
  - 逻辑回归：$0 \le h_\theta(x) \le 1$
  - 预测1：$h_\theta(x) \ge 0.5$
  - 预测0：$h_\theta(x) < 0.5$

- 准确率　　P = TP / (TP + FP)
- 召回率　　R = TP / (TP + FN)

- **提高阈值**
  - 提高准确率P，召回率R下降
- **减小阈值**
  - 提高召回率R，准确P下降

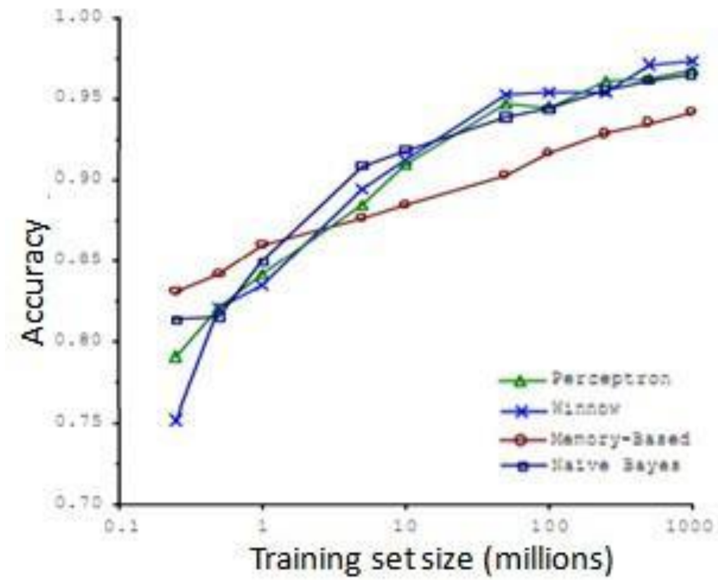


P和R指标矛盾，需要综合考虑，最常见的方法就是F1值（F1-Score）

$$F1 = \frac{2 * P * R}{P + R}$$

机器学习应用系统设计-数据

- 数据集大时，不同类型的算法表现出了较好的效果。



- 如果一个模型具有很多参数（神经网络、拥有很多特征的线性回归、逻辑回归模型）——低偏差算法
  - 可以通过大量的数据集来降低方差。
  - 得到一个低方差和低偏差的学习算法