1. 讲机器学习的术语，训练样本和训练集，然后根据术语引出假设，然后再来推假设
2. 案例：房价预测

size	rooms	price
2104	3	400
1416	2	232
1634	3	315
852	2	178

线性回归

线性回归是利用数理统计中回归分析，来确定两种或两种以上变量间相互依赖的定量关系的一种统计分析方法，运用十分广泛。其表现形式就是y=a*x+b 多个x就依次排列

设x1=size，x2=rooms

h=x1*θ_1+x2*+θ_2+θ_0

设x0=1，则h就等于xi*θ_i的求和，i是0到2

如果有n个特征，就是0到n

也等于θ的转置乘以x，θ叫做学习算法的参数

学习算法的任务就是得到一个合适的参数

这里一定只能用线性回归吗？并不是，只是作为第一个算法，我们选择了线性，当然可以有其他更加复杂的假设，更高阶的函数

如何选择一个合适的参数，让h对新的输入做出正确的预测？

一个合乎情理的做法，有一个训练集合，然后使用h对训练结合进行预测，然后让实际预测和真实值之间的平方差尽可能小，有m个样本，就需要对每个样本的预测值与真实值之间平方差和最小，一般我们会对公式乘以1/2，为了简化数学运算

将整个式子赋给一个新的函数，那么整个问题就变成了使新函数的值最小的问题

$J(\theta) = \frac{1}{2}\sum_{i=1}^{n}(h_\theta(x^{(i)}) - y^{(i)})^2$

批梯度下降算法：思想，给参数一个初始值，然后不断改变参数，使新函数一直变小，知道达到我们满意的程度

下山，不断找最近下山的方向

但是不同的位置得到的不同局部最优值

一个样本最终结果：θ_i:=θ_i-α(h_i(x)-y)*x_i

阿尔法，代表学习速度，也就是下山的迈步大小，这个值手动设置，如果过小，算法会很长时间才会收敛最终得到结果，如果过大，会越过最小值

m个样本的最终结果：

θ_i=θ_i-$\alpha\sum_{j=1}^{n}(h_\theta(x^{(j)})-y^{(j)})-x_i^{(j)}$

当接近局部最小值的时候，梯度会越来越小，算法的步子也会越来越小，收敛越来越慢

如何检测收敛：两次迭代中变化是否很多，如果没怎么边，认为收敛

还可以检测参数的值，变化很小的时候，认为收敛

这个算法如何环顾一周，找到最合适的方向？

在计算偏导的时候，实际上就是下降最陡峭的方向

批梯度下降算法问题：每次都要对所有样本求和，因此训练集合很大的时候，算法效率很低，这个时候可以使用，随机梯度下降

随机梯度下降算法

随机梯度下降通过随机选取小量的m个训练输入来工作。我们将这些随机的训练输入标记为X1,X2,...,Xm，并把它们称为一个小批量数据（mini-batch），然后再做梯度下降

拟合：形象的说，拟合就是把平面上一系列的点，用一条光滑的曲线连接起来。因为这条曲线有无数种可能，从而有各种拟合方法。拟合的曲线一般可以用函数表示，根据这个函数的不同有不同的拟合名字。

过拟合： 仅仅反应了集合数据的特点，但反应不出隐藏在数据下面的一般性规律

欠拟合：一些（很多）集合数据没有被成功拟合出来

参数学习算法：有固定的参数用来进行数据拟合的算法

非参数学习算法：一个会随着训练集合的大小而线性增长参数数目的算法

局部加权回归（Loess）

不需要太担心特征的选择

非参数学习算法

主要思想：对预测周围的数据做线性回归，而忽略掉远距离的点