1. SVM用了一种数学trick来处理无穷多的feature
2. 回归(regression)：目标值连续 分类(Classify)：目标值离散
3. 学习算法的任务就是要输出一个函数，当新来一个x的时候，能够用来预测y，其实一开始要对作个假设，比如是一次函数，二次函数，或者什么，然后把参数求出来带进去，所以叫“回归”！然后你的预测值要和样本的真实值的误差尽量的小，其实是个最优化问题，目标函数，这个就是代价函数(cost function)

# 机器学习第一个算法：梯度下降

对参数求导，然后变一点，再求导，再变一点，一直到求导=0。最开始的时候给参数赋一个初值，随便赋，他们的初值并不重要，重要的是有一个算法，一直在良好的改变它，有点像最优化的思想，不求一步最优，但求步步为营，最后达到最优。

参数向量的更新是同步的，就是一次求出所有，再一次更新，再迭代下一次，而不是求出某一个分量，然后更新代价函数J，然后再求偏导，虽然这样也可以做，但是这个不是我们大家心中定义的梯度下降算法。

梯度下降的更新，其中永远是个正数，叫做学习率。如果取得太小，每次改变很小，那就需要很多次改变，你的算法会很慢；如果太大，有可能造成无法收敛的情况，步子迈得太大了。那么，到底取多少比较合适呢？好问题！！调参！！在实践中得出经验吧！！！

在梯度下降的过程中，如果算法正确，选取得当，每次的步伐会越来越小，因为导数越来越小，所以不用把调低，有点柯西序列的意思。

线性回归的梯队下降法称为Batch梯度下降，因为每次求都要遍历整个数据集，看公式，有个求和号

有一些梯度下降法不需要遍历整个数据集，也有些方法不需要梯度下降就可以求出参数

线性回归的梯度下降可以扩展到多参数的情况，其实跟上面的情况是一样的

其中令

### 梯度下降中的一些细节问题

1. 特征缩放（就是各个特征之间的数量级差的太多了，那么你的坐标轴就很不好画，这个时候要把他们的量级统一一下，就像你画排序算法的结果分析的时候，横坐标数据规模n每次增加10倍，但是时间消耗会越来越大，到后面两个坐标轴的数量级差太远，这不行）

特征缩放不需要太精确，只要让梯度下降比较好的运行就行了，所以不需要太抠边界。

WHY？为什么要进行特征缩放？因为不进行特征缩放，算法会很慢，而且数据容易溢出。

WHEN？什么时候特征缩放。

特征值之间的数量级大于1，我觉得就要进行特征缩放了。特征值都很大或者都很小，没问题，最怕的是它们之间的相对差距，相对，相对！！理解相对！！

HOW？怎么进行特征缩放

方法1（归一化）：如果提前知道每个特征的取值范围，可以用每个特征除以它的取值范围，将其都压缩到区间。其实在1这个数量级的基本都ok，比如都是可以的，但是不要超过两个数量级，比如100或者0.01这种就要特征缩放。

方法2（均值归一化）：，

1. 学习率怎么取值。

首先要选取一个使得梯度下降是正确工作的，就是真的在梯度下降。直观来看，就是的值随着梯度下降的次数的增加变得越来越小，这说明算法在正确运行。可以画个图，纵坐标是，横坐标是迭代次数，一目了然。

如果越来越大，可能是的值取大了。

如果随着迭代次数的增加出现周期性增减，也可能是取得太大导致。

通常来说，一开始可以每隔10倍选一个，然后看随着迭代次数的增加的曲线，然后确定一个大致的取值区间。

1. 到底用一个什么样的函数来拟合数据呢？有一些算法会自动得出应该用什么函数来拟合，牛逼！也会自动得出怎么样来组合特征以更简单的求解，而不是用最直观的基本特征。

### 正规方程一步到位求解

怎么推的没说，先背一波。虽然正规方程可以一步求解，看起来很牛逼，但是你想想，当特征的数量很大的时候，矩阵相乘的代价是，时间上可能会比梯度下降法慢，一般当的时候是拐点，当然这跟计算速度也有关系，越快的话，拐点就在越后面。

# 问题

代价函数不用平方误差还能用什么？？？？？？？？？

# 作业

手撸一个单变量线性回归和多变量线性回归，3个变量吧。

# 机器学习的第二个算法：支持向量机（SVM，要会手推哦亲）

# 机器学习的第三个算法：反向传播算法