**中心极限定理** （唐义鸿）

大量随机变量近似服从正态分布 示例如下

import numpy as np

import matplotlib.pyplot as plt

pop = np.random.randint(1,100,3000) #生成3000个随机数

means\_size\_100=[]

for \_ in range(10000):

sample = np.random.choice(pop, 100)

means\_size\_100.append(sample.mean()) #随机取100个求和，取10000次

means\_size\_100 = np.array(means\_size\_100)

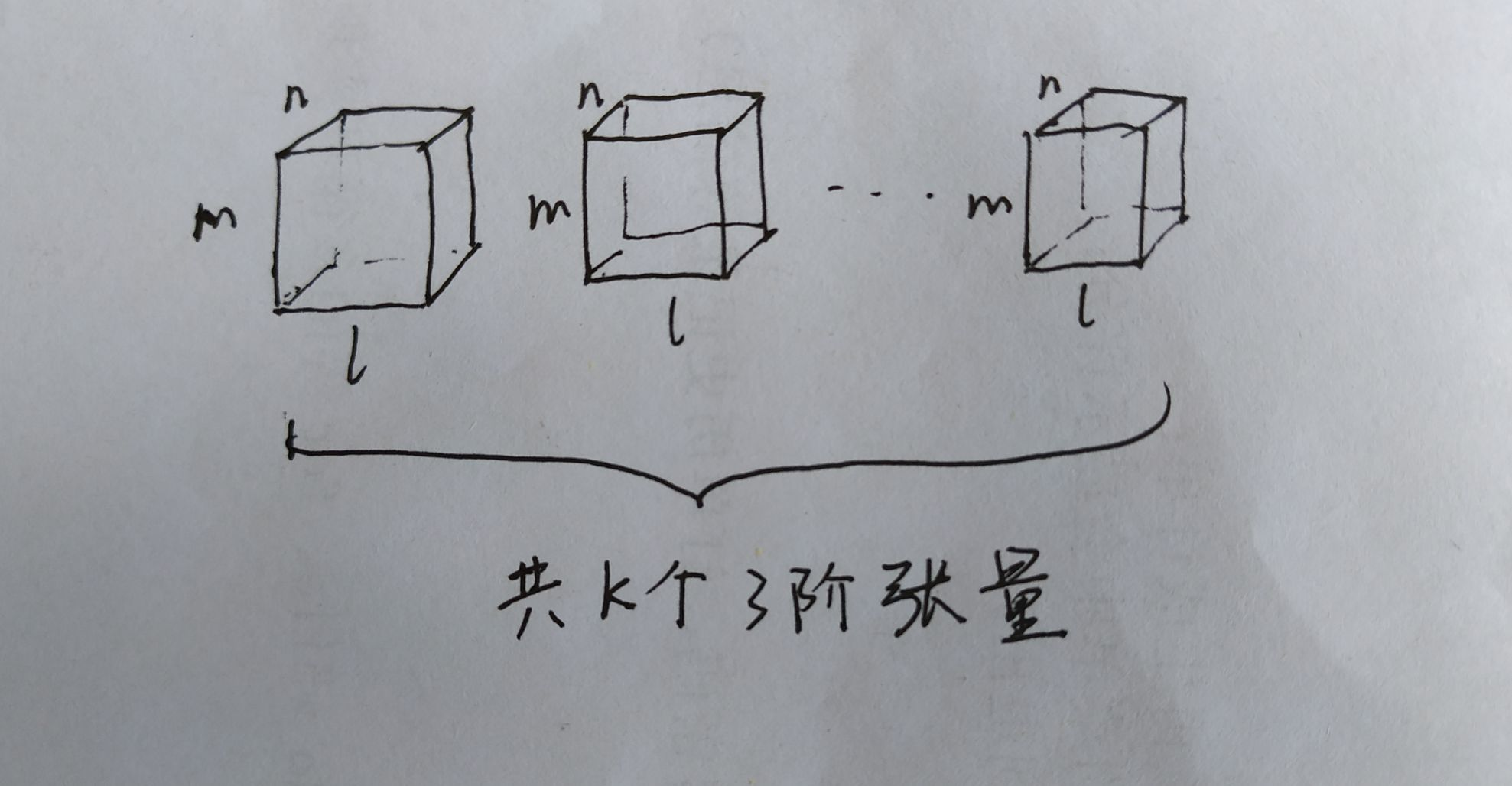
plt.hist(means\_size\_100) #得到的数据近似符合正态分布

plt.show()

**梯度**（梁明总结）

1. 定义：梯度本质是一种向量，表示函数变化速度最快的方向。
2. 前提：函数在平面区域内有一阶连续偏导数。
3. 公式：grad f(x,y) = fx(x,y)i + fy(x,y)j
4. 梯度与方向导数的关系：
5. 方向相同：方向导数取得最大值，且值为梯度的模。
6. 方向相反：方向导数取得最小值。
7. 方向正交：函数的变化率为0

**求导（标量，向量**）（张德龙总结）

1. 向量y对标量x求导：将y中的每一个元素分别对x求导写成列向量（分子布局）
2. 矩阵y对标量x求导：同样将y中每一个元素分别对x求导写成矩阵形式。
3. 标量y对向量x求导：用y分别对x中的每一个元素求导写成行向量（梯度向量）
4. 向量y对向量x求导：结合1、2可得jacobian矩阵，矩阵中同一行中的元素为（ y(i) , x(1 到 n) ）,同一列中的元素为（ x(j), y(1到n) ), 其中i、j分别为对应行、列序号。
5. 矩阵y对向量x求导：同一行中y中对应元素分别对x中对应元素求导。
6. 标量y对矩阵x求导：y对x中每个元素分别求到，写成矩阵（梯度矩阵）
7. 矩阵y对矩阵x求导：Y（m，l）/X（n，k）可以看作Y（m，l）对x（n，1）得到立体3阶张量X（n，k）包含k个向量x，依此求导得 （胡涛总结）​​

**链式法则** （胡涛总结）

链式法则简单来说就是对复合函数求导，再综合上面的向量或矩阵求导即可

**线性回归** (邓楠总结)

1. 作用：预测诸如房价这样一个连续值的输出

2. 步骤

(1) 提取影响预测值的特征并赋予参数，构造计算预测值的假设函数

(2) 构造平方差损失函数，对参数求偏导

(3) 梯度下降：参数重复更新，每次减去该处该参数的偏导值，以求损失函数值最小。确定学习率时注意太大会无法收敛，太小会下降时间太长

3. 优化方法：MBGD，每次只使用一批样本来训练。样本太大样本中可能有重复冗余，浪费算力，太小不能充分利用算力

**自动微分**

（李思梦总结）

1. 反向积累
2. 建立计算图；
3. 正向：求图值，存储中间结果；
4. 去掉不需要的路径；
5. 以相反的顺序评估图；
6. 复杂性
7. 计算复杂度：O(n),n是#运算，用来计算所有的导数；
8. 通常类似于远期成本，记录所有中间结果；
9. 与正向累积比较：

计算梯度的时间复杂性，O（n\*k）计算k变量的梯度；

1. 记录复杂性；

大概流程：

假设M部分，然后O（M）代表头部结果，O（N/M）代表存储一个部分的结果；

设m=n的0.5次方，然后记录复杂性O（m）；

应用于深度神经网络；

仅丢弃简单的层，通常少于30%的额外开销；

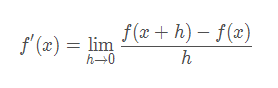
训练10倍大网络或10倍大批量；

**迭代加深搜索算法（应旻昊总结）**

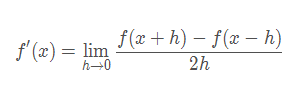
通过限制每次dfs的最大深度进行的搜索。令maxd表示最大的搜索深度，那么dfs就只能在0~maxd之间来进行，如果在这个范围内找到了解，就退出大循环，否则maxd++，扩大搜索范围。所以可想而知，倘若没有高效及时的退出无解的情况，那么时间上的开销也是会比较大的。

## 1.数值微分：

数值微分是根据导数的原始定义来计算微分：



它强调一开始就带入数据，以求得近似解，其实现简单，但运算量太大了，而且会产生误差，因此有中心差近似的方法：



**中心差近似**可以减小误差，但无法消除误差，这种方法我们可以用于梯度检查。

## 2.符号微分：

符号微分法则强调直接的代数运算和完整的数学表达式，借助代数软件和其中的一些微分表达式，对数学表达式进行‘自动微分’。问题如果能表示成纯数学符号形式，这种方法尤为有效。此法对于完整的（closed form）数学表达式的依赖是其一大缺点，另一缺点是表达式膨胀（expression well）,稍不注意问题的求解速度就会因求解表达式的极速膨胀而变慢。

## 3.自动微分（automatic differentiation）:

(1)自动微分法介于数值微分法和符号微分法之间，它将符号积分用于基本的算子（如常、幂、指、对、三），然后带入数值，保存中间结果，最后再运用于整个函数。它的计算实际上是一种图计算。

(2)自动微分的两个模式：

①前向模式(forward mode aka forward accumulation)

前向模式先将表达式转化为计算图，然后从上到下计算每一步的导数，(下一步的导数计算利用了上一步导数的计算，链式法则原理，避免了重复计算，解决了表达式膨胀问题)前向模式可求出问题的值和导数。

②反向模式(reverse mode aka reverse accumulation aka backpropagation)

反向模式用通过正向遍历求得计算图中每个节点(参数)的值，然后再利用链式法则反向计算每个节点的偏导数，反向模式可以求得每个参数的导数。

4.自动微分的实现（jupyter notebook）

1. 对仅有一个未知数的函数自动求梯度 例：y = e^x + 3\*x^2

from mxnet import nd, autograd as ag #导入autograd模块

x = nd.array([2, 3]) #创建x并赋初值

x.attach\_grad() #申请存储梯度所需要的内存

with ag.record(): #记录有关梯度的计算

y = x.exp() + 3 \* x \*\* 2 #给出函数表达式

y.backward() #backward函数实现自动求梯度

print(x.grad) #输出x的梯度

2. 对多元函数求梯度 例：z = -log ( e^x / (e^x + e^y) )

from mxnet import nd, autograd as ag #导入autograd模块

x, y = nd.array([2]), nd.array([3]) #创建x, y并赋初值

x.attach\_grad()

y.attach\_grad() #申请存储梯度所需要的内存

with ag.record(): #记录有关梯度的计算

z = -nd.log(x.exp() / (x.exp() + y.exp())) #给出函数表达式

z.backward() #backward函数　　　　　　　实现自动求梯度

print(x.grad) #输出x的梯度

print(y.grad) #输出y的梯度

3. 对复合函数求梯度 例：z = y^2 + y + x y = x^2 求dz/dx

from mxnet import nd, autograd as ag #导入autograd模块

x = nd.array([2, 3]) #创建x并赋初值

x.attach\_grad() #申请存储梯度所需要的内存

with ag.record(): #记录有关梯度的计算

y = x \*\* 2

z = y \*\* 2 + y + x #给出函数表达式

z.backward() #backward函数实现自动求梯度

print(x.grad) #输出x的梯度

（叶钰莹总结）

**linear regression(线性回归)**

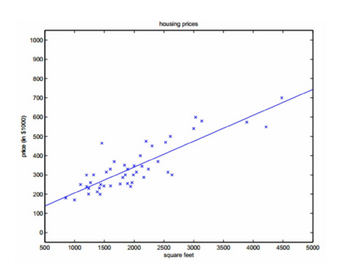
1．线性回归的模型是指对于一个样本C:\Users\LENOVO\Documents\Tencent Files\984219275\FileRecv\MobileFile\Image\9D]LU@@HHCN{S~A5`TCSRJ9.png，它的输出值是其特征的线性组合：

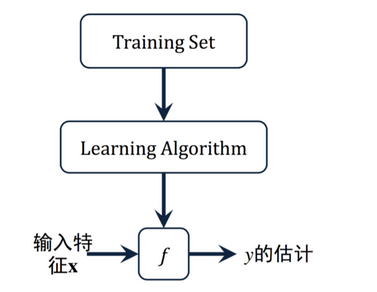
C:\Users\LENOVO\Documents\Tencent Files\984219275\FileRecv\MobileFile\Image\V%$KI_P9OSCT9Q{A3Z411WW.png

（其中C:\Users\LENOVO\Documents\Tencent Files\984219275\FileRecv\MobileFile\Image\4E~~KT(TK71%5C}3L8L)4FM.png称为截距（bias）,上式增加C:\Users\LENOVO\Documents\Tencent Files\984219275\FileRecv\MobileFile\Image\1ZUA[5X92`H4@)AD]M7@3D7.png把C:\Users\LENOVO\Documents\Tencent Files\984219275\FileRecv\MobileFile\Image\4E~~KT(TK71%5C}3L8L)4FM.png吸收到向量表达式中，简化了形式，故C:\Users\LENOVO\Documents\Tencent Files\984219275\FileRecv\MobileFile\Image\9D]LU@@HHCN{S~A5`TCSRJ9.png上有p+1维度）

图示：

Eg:(房价问题)

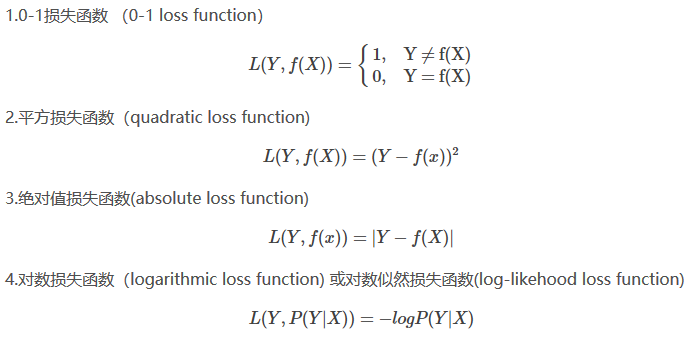




2.**loss function（损失函数）**

①概念：用来估量模型的f(x)与真实值Y的不一样程度，是一个非负实值函数，通常用L(Y,f(x))表示。

②常见的损失函数：



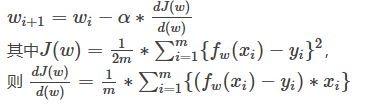
③线性损失函数的求解方法：

1. 最小二乘法
2. 梯度下降法

**3.梯度下降算法（GD）**

目标函数下降最快的方向为梯度的负方向。因此梯度下降算法为沿梯度的负方向逐步的使目标函数值最小化的过程。

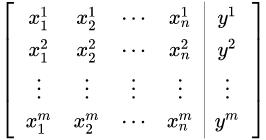
对应的数学表达式：



（王树仁总结）

在统计学中，**线性回归（Linear regression）**是利用称为线性回归方程的最小平方函数对一个或多个自变量和因变量之间关系进行建模的一种回归分析。 这种函数是一个或多个称为回归系数的模型参数的线性组合。

训练数据的数学定义，假设我们训练数据中有m 个样本，每个输入样本中有n个特征输入和一个标注的输出：

​​

第i个样本，记作：[|]

**数学模型：**

线性回归的数学模型定义很简单，就是一维的线性方程：

C:\Users\thinkpad\AppData\Roaming\Tencent\Users\736704660\TIM\WinTemp\RichOle\$X4K3K]2E{M84J%Y6)H]O}Y.png

而训练数据中的第i个样本，记作：

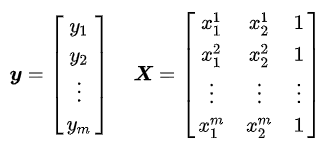
C:\Users\thinkpad\AppData\Roaming\Tencent\Users\736704660\TIM\WinTemp\RichOle\5UU{FZ{O6574E7%@{@MMOOC.png

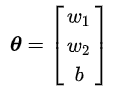
从上可以看出线性回归的数学假设是，训练数据中所有输出和所有输入的特征之间是一个线性表达式的关系。

**损失函数：**

在线性回归的计算式当中，主要是有两个需要计算的数学公式，一个是计算损失函数，一个是计算损失函数的梯度。损失函数为 度量所选数值通过函数表达的最优解

首先用向量来定义训练样本，定义如下：

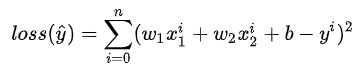


定义要学习的参数向量为：

给定当前参数，计算出来的输出数组的向量计算表达式看起来就很简单了，如下：



我们再来看如上线性回归定义的损失函数，原始定义如下：



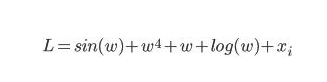
调整为向量表达式之后变成如下了：



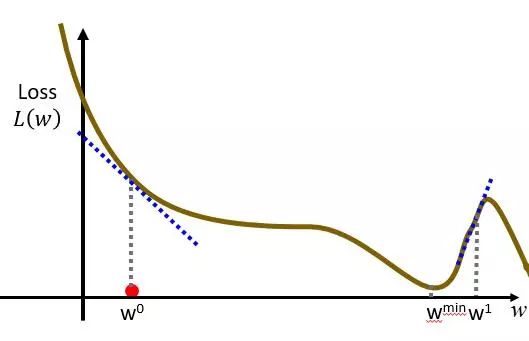
**basic optimization：**

一维变量下，变量只能在一条直线上移动，向左或向右。

假设损失函数是有一个参数 w 的函数，例如



是要输入的数据特征，可以看做常量。它的图像假设为：

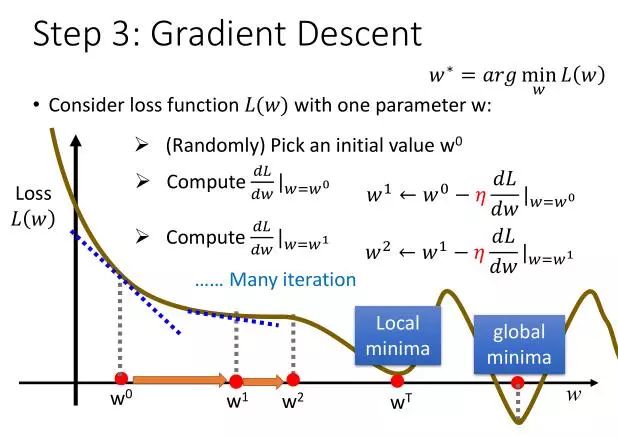


从图像上可以看到要找的使得 Loss 值最小的就是,如果开始选取的是，那需要给加上一个值得到。

如果开始选取的是，那需要给减去一个值得到。

是加上还是减去，我们可以观察到，当 w 的斜率为负的时候就加，斜率为正的时候就减。斜率等于 0 的时候就代表是一个最小值点了。

所以我们可以先把 w 随便取一个值 ，然后求出该点的斜率，也就是 的导数值 ，然后用 减去这个导数值乘以一个 ，叫做 learning rate ,就是控制它从 走到 的速度。为什么是减去呢，因为斜率的符号和要加还是要减刚好是反着的。就是下边的数学表达式

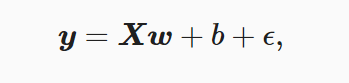


根据得到，再根据得到，一直迭代，直到使得 Loss 值最小，当然从图上就可以看到我们可能会到达一个 Local min，也就是局部最小。而线性回归不需要考虑这个问题，因为它如果有极小值，那么这个极小值一定是最小值

（徐明雪总结）

线性回归从零开始实现：

1. 生成数据集

核心：随机生成特征，x1,x2;使用真实权重和偏差及随机噪声项生成标签，y。

1. 读取数据

定义data\_iter函数，遍历数据集.

每次拿出一个batch\_size的样本出来。

1. 初始化模型参数

权重初始化为正态随机数，均值0，标准差0.01，偏差0。

创建w,b的梯度，通过迭代来求w和b。

w.attach\_grad()

b.attach\_grad()

1. 定义模型

定义Linreg函数，返回y。

1. 定义损失函数

定义squared\_loss函数，注意使y的形状变为y\_hat的形状。

1. 定义优化算法

定义sgd函数，通过迭代模型参数来优化损失函数。

1. 训练模型

（钱鹏屿总结）

1. **linear regression**

目的：给定一个线性模型和数据对，求解出模型的最优参数。

数学推导：

所有样本用坐标表示为(xi,yi) i={1,2...m} 即共有m个各样本点。

线性方程形式：

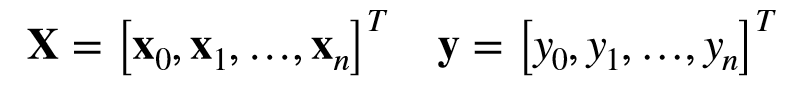
估计值：

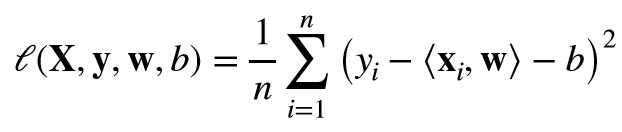
误差：  展开

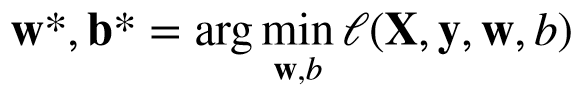
多维推广：

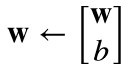
令

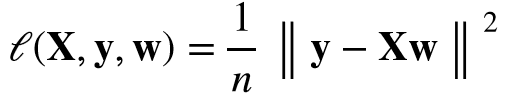
有

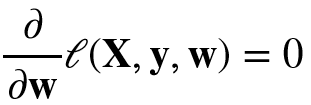
令

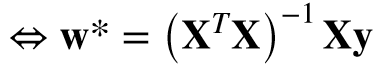
则

即为求最优参数**w**∗和b∗使得样本误差值最小。

封闭解**w**∗求法：



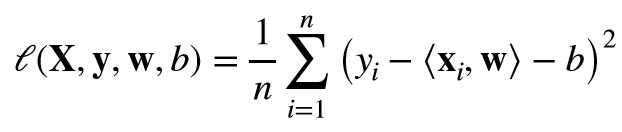


最优解**w**∗满足

1. **basic optimization**

线形规划目标：如何不断的调整参数w和b，来使样本误差值越来越小，最终令预测尽可能接近真实值。——优化函数

梯度下降法：每一次迭代按照一定的学习率η沿梯度更新参数，直至收敛。



样本误差 分别对**w**和b求偏微分，分别表示样本误差在**w**和b方向的梯度。

更新参数的方法：