**第3章 梯度下降法**

函数优化就是寻找使函数值最小的自变量。在机器学习模型训练的语境下，就是寻找使损失函数最小的模型参数值。梯度下降法是基于函数局部一阶特性的优化算法。它是神经网络和深度学习中最主要的训练算法。

本章首先回顾多元微积分基础。介绍多元函数的梯度、方向导数、偏导数等概念。在自变量空间中某一点附近，函数可以用它在该点的切平面近似表示。切平面的朝向和倾斜程度蕴含在函数在该点的梯度之中。这些信息就是函数在该点局部的一阶信息。

之后，本章介绍梯度下降法。梯度下降法利用梯度确定函数值下降最快的方向，然后向该方向前进一段距离。迭代地重复此步骤，希望使函数值不断下降，乃至寻找到函数的全局最小点。具备了多元微积分的相关知识后，能深刻地理解梯度下降算法。

由于梯度下降算法只利用了局部一阶特性，所以它是短视的。再加上梯度下降法本身的离散特性，会带来的种种问题。本章举例介绍几个梯度下降法遭遇的问题。这些问题的成因以及规避和改进办法，将在下一章介绍函数二阶特性后加以说明。

最后，本章介绍运用梯度下降法训练逻辑回归模型。阅读完本章，读者应能透彻理解梯度下降法原理和局限，并完整地理解逻辑回归模型。

**3.1 多元微积分**

本节名为“多元微积分”，其实我们主要关注多元微分。微分刻画了函数的局部近似特性。寻找函数的最小点就利用了这些局部近似特性。

**3.1.1 梯度**

首先，回忆一下一元函数的可导性及其导数：

（3.1）

如果极限（3.1）存在则在可导。是自变量空间中的某一点。是自变量的一个变化量。在的图像中用一个线段连接和两点。这个线段称为割线。式（3.1）极限里的商是割线的斜率。随着趋近于0，割线趋近于在的切线。割线斜率的极限是切线的斜率。如图3-1所示。

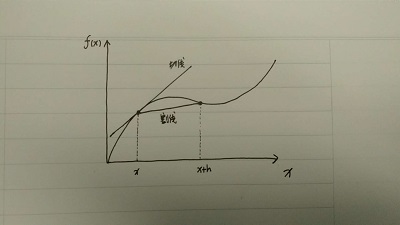


图3-1 一元函数的割线、切线和斜率

也是自变量从变化到时，函数值的平均变化率。是平均变化率的极限——在的瞬时变化率。

在一元情况下，自变量只能沿着轴前后运动。可以如式（3.1）那样用瞬时变化率定义导数。但在多元情况下自变量是向量，它可以沿无数方向运动。这时就不能用瞬时变化率定义的导数。

一元函数的导数还有另一种定义，即在附近用直线近似表示。这种定义可以扩展到多维的情况。对，构造一个以为自变量的仿射变换：

（3.2）

令余项。根据式（3.1）有：

（3.3）

所以，作为的函数，可以写成一个仿射变换加余项的形式：

（3.4）

容易看出作为两个连续函数的差是连续的，而且。所以有。变化量趋向于消失时，余项也趋向于消失。但这还不够。根据式（3.3），有：

（3.5）

变化量趋近于0时，余项与变化量之比趋近于0。也就是随着变化量的消失，余项也消失，而且余项比变化量消失得更快。这种情况称是的高阶无穷小。

当时仿射变换的图像经过点。的图像是一条截距为，斜率为的直线。如果将的图像平移，使原来的移动到。平移后的图像是经过的斜率为的直线——在的切线。如图3-2所示。

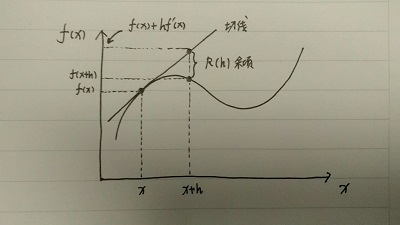


图3-2 可导函数的仿射近似——切线

反过来，如果在附近的变化可以写成：，是的高阶无穷小，那么有：

（3.6）

式（3.6）说明在可导，导数。

综上所述，在可导等价于：自变量为的函数可以被一个仿射函数近似表示，且与之间的误差是变化量的高阶无穷小。该仿射函数的斜率就是。

现在将这种可导性的定义扩展到多元函数。将变化量作为自变量，如果可以被一个仿射变换近似表示：

（3.7）

其中是变化量的长度的高阶无穷小：

（3.8）

则称多元函数在可导**。**式（3.7）中的是一个向量，称为在的梯度（gradient）。的近似仿射变换是：

（3.9）

如果自变量是维，则的图像是维空间中一个超平面。该超平面经过点。将该图像平移，使移动到。平移后的超平面称为在的切平面。切平面是在附近的一阶近似。切平面的特性就是的局部一阶特性。根据第1章的介绍，切平面的法向量是维向量，即给梯度添加一维常量-1。

仿射函数的全部特性体现在中：的方向决定超平面的朝向，的大小决定超平面的倾斜程度。所以的局部一阶特性都包含在梯度中。如图3-3所示。

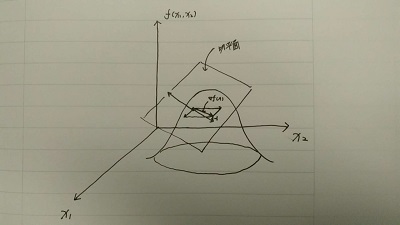


图3-3 多元函数的切平面

**3.1.2 方向导数**

如果在可导**，**如何讨论在的瞬时变化率呢？指定一条经过的直线，然后讨论当自变量沿着这条直线运动时在的瞬时变化率。经过的直线可定义为：

（3.10）

式（3.10）定义了一条经过点的直线。其中是单位向量。它的方向决定了直线的走向。是实数。决定了离的距离，。可以把该直线当作自变量空间中一个以为原点，以的方向为正方向的坐标轴。点的值就是在这个坐标轴上的坐标。

接下来在基础上定义一个复合函数：

（3.11）

是自变量为的一元函数。根据式（3.1），在0的导数是：

（3.12）

是在沿的方向导数（directional derivative）。上文提到过，直线定义了一个坐标轴，在该坐标轴上。如果限制自变量只能在这条坐标轴上变化，这就相当于定义了一个一元函数。方向导数是这个一元函数在瞬时变化率。如图3-4所示。坐标轴是有方向的，其方向由的方向确定。所以确定了方向相反的坐标轴，函数的变化率相反。即。

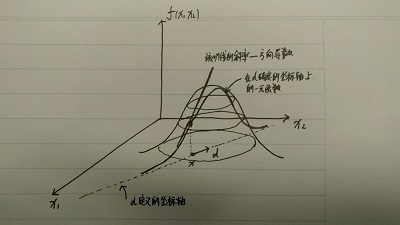


图3-4 方向导数

在沿各个方向的方向导数都蕴含在梯度中。因为在可导**，**根据式（3.7）有：

（3.13）

其中是的高阶无穷小：

（3.14）

因为，所以当趋近于0时也趋近于0。这时有：

（3.15）

所以是的高阶无穷小。另外因为，所以式（3.13）表明在0的导数是，即。现在有：

（3.16）

其中是与之间的夹角。

由式（3.16）可知：方向导数等于梯度向的投影长度。当与同向（）时，最大。所以是变化率最大的方向，其变化率是。相反，沿着的反方向（）最小，为。是变化率为负且最小的方向，即函数值下降最快的方向。

在2维的情况下可以用切平面阐述梯度与方向导数的关系。切平面的法向量是。第3维是-1说明向下指向平面的下方。在平面的投影是，它指向切平面的上坡方向，指向切平面的下坡方向。

自变量沿任意方向的运动可分解为两个分量：沿方向的分量和垂直于的分量。垂直于的方向上。自变量沿的运动一部分摊在了垂直分量上，这部分运动不会导致变化。所以的变化率就打了折扣，折扣系数正是自变量运动沿方向分量所占的“份额”——。如图3-5所示。

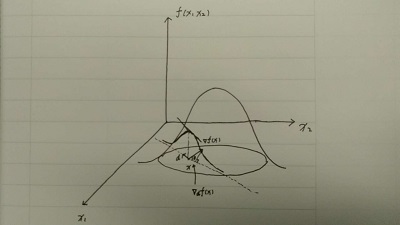


图3-5 梯度与方向导数

**3.1.3 偏导数**

在点对第分量的偏导数，是把其他分量当作常数时对的导数。这时候将看作关于的一元函数。根据导数的定义：

（3.17）

其中是第个标准基向量。保持不变，只有发生变化，变化量是。有个偏导数：。

根据式（3.12），有：

（3.18）

对的偏导数就是沿的方向导数。偏导数是方向导数的特例。它们的方向分别是个坐标轴的正方向。

与的内积是的第分量。根据式（3.17），有：

（3.19）

所以梯度的第分量是对自变量第分量的偏导数。于是就有了梯度的计算式：

（3.20）

偏导数是唯一的，所以也是唯一的。

3.1.4 **驻点**

函数的驻点（stationary point）是梯度为零向量的点。在驻点的切平面的法向量是：

（3.21）

法向量垂直指向下方，即切平面是水平的。如图3-6所示。

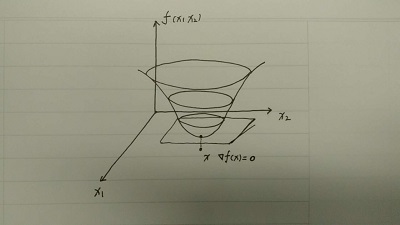


图3-6 驻点的切平面

在驻点沿任意方向的方向导数是，所以在驻点向任意方向的方向导数都为0。

**3.1.5 局部极小点**

如果是的局部极小点（local minima），则在周围存在一个半径为的邻域，该邻域所有点的函数值都不小于。用公式表示就是：

（3.22）

如果对于全部都有，则是的全局最小点（global minima）。很显然，全局最小点是局部极小点。但是局部极小点不一定是全局最小点。类似还可以定义局部极大点（local minima）和全局最大点（global maxima）。如图3-7所示。

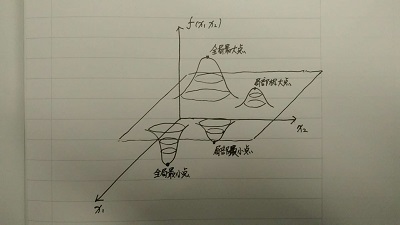


图3-7 局部极小点和全局最小点

一个局部极小点一定是驻点，即。运用反证法。假设。从点出发沿方向产生一个位移，。根据在的可导性，有：

（3.23）

是的高阶无穷小。于是有：

（3.24）

这说明式（3.23）等号右边的后两项当趋近于0时的极限是负值。所以对于足够小的，当时有：

（3.25）

随着趋近于0，在无限靠近的同时保持。这与是的局部极小点矛盾。所以只能是零向量，即是驻点。类似可以证明，局部极大点也一定是驻点。

驻点是局部极小点的必要非充分条件。驻点也有可能是局部极大点或者鞍点（saddle point）。鞍点的梯度也是零向量，但在任意一个邻域内都同时存在函数值更大和更小的点。如图3-8所示。

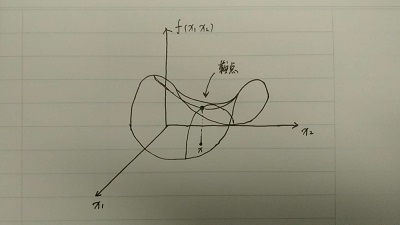


图3-8 鞍点

仅靠一阶特性难以判断驻点的类型。第4章介绍赫森矩阵后会揭示：驻点的类型由赫森矩阵特征值的符号决定。

**3.2 梯度下降法**

为了寻找函数的全局最小点，可以先找到满足必要条件的点——驻点。但大多数时候这并不可行。举个简单的例子，有一个二次型（quadratic form）函数：

（3.26）

满足。该二次型对每一个自变量的偏导数是：

（3.27）

令梯度是零向量，求，相当于解元1次方程组：

（3.28）

一般情况下，这需要的时间复杂度。当非常大时（这在神经网络和深度学习中是必然的），求驻点解析解是不可接受的。例子（3.27）还仅仅是简单的二次型的情况。当情况更复杂时，梯度为零向量的解析解甚至是不存在的。这就需要迭代的数值解法。

**3.2.1 反梯度场**

如果是元函数，则是维向量。可导函数在自变量空间中每一个点都有一个反梯度向量，指向下降最快的方向。这就形成了一个速度场。可以将画成箭头，尾部移到的位置，把这个速度场呈现出来。如图3-9所示。

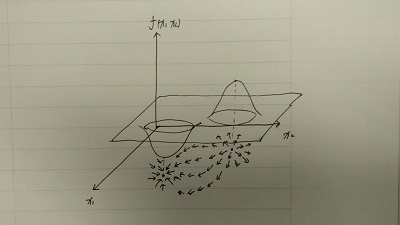


图3-9 反梯度场

速度场在每一点指定了该位置的速度——方向和速率。在反梯度场情况下，速度指向函数下降最快的方向，速率大小是函数的下降速率。将一个粒子（particle）从任意位置放入速度场中，它就会按照场指定的方向和速率运动。在反梯度场情况下，粒子就是朝函数下降最快的方向运动。

这里澄清一下反梯度场的物理意义。在2维情况下，将的图像看作一幅起伏不平的地形，设重力加速度是。有一个质量是的小球被放在任意位置。地面对小球的支持力垂直于坡面指向上方。即该支持力是沿着在该点向上指的法向量的方向，即的方向。与平面的夹角的正弦是。该支持力的垂直分量抵消重力，水平分量的大小则是。水平分量的方向是向平面的投影的方向。所以小球在地形上受到的水平作用力正是。小球的水平加速度就是。所以粒子在反梯度场中的运动并非模拟小球在函数地形上自然滚落。

局部极小点的梯度为零向量。它们是反梯度场的静止点。如果一个粒子处于静止点上，它将不发生运动。同理局部极大点也是静止点。但是，局部极小点是稳定静止点，或者说吸引子（attractor）。当粒子偏离局部极小点一个小位移，它将被吸引向局部极小点。局部极大点和鞍点是不稳定静止点，或者说排斥子（repeller）。粒子位于局部极大点或鞍点时，它也是静止的，但是一旦有一个微小的扰动使它发生极小的位移，它将被推得远离该局部极大点或鞍点。如图3-10所示。

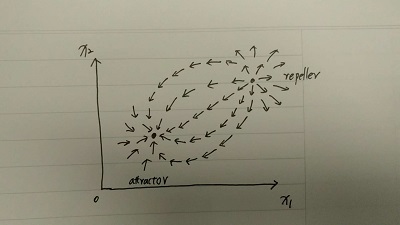


图3-10 吸引子和排斥子

我们可以从任意位置开始模拟粒子的运动。除非粒子初始位置刚好是不稳定静止点，否则粒子将向函数值下降的方向运动，并最终无穷逼近吸引子——局部极小点。

从理论上，反梯度场只能保证粒子运动到局部极小点，不能保证运动到全局最小点。收敛的速度也没有保证。实践中无法精确模拟粒子的运动，而只能以一种离散的方式近似模拟。这会将带来更多问题，甚至不收敛。

**3.2.2 梯度下降法**

在计算机中模拟粒子在速度场中的运动，属于数值积分问题。梯度下降法（gradient descent, GD）是一个简单的数值积分算法。伪代码如下：

randomly initialized

while :

是一个预设的阈值。当时认为已经足够接近零向量，算法停止。也可以采用其他停止标准。例如循环次数达到预设的最大值，或者函数值的下降幅度小于阈值。

是另一个预设值，称为学习率（learning rate, LR）或步长。每一步迭代自变量向方向运动，运动的距离是。是梯度下降算法的一个关键参数。可以保证的是，在某个点，能够找到一个合适的步长使得。也就是步长可以使函数值下降。这是因为：

（3.29）

是的高阶无穷小。式（3.29）与式（3.23）相似，可以证明存在一个，当时有。称是确保下降的方向。只是对于不同的，也不同，而且无法计算出的值。所以只能选择固定的步长。

**3.2.3 梯度下降法的问题**

因为是的局部近似特性，在距离过远的地方的形状会有大的、未知的变化，这是无法体现的。所以如果设置得过大，函数值有可能不降反升。较小的更有可能保证函数值下降。但是如果过小收敛的速度会很慢。

函数图像的“地形”千奇百怪，有很多病态情况都会梯度下降法的效果产生负面的影响。本节试举几例。

“悬崖”如图3-11所示。如果悬崖底是局部极小点所在位置，当解靠近悬崖底时，过大的会使解一步跨过崖底，爬上了对面的崖顶。崖顶有非常大的梯度，一下将粒子弹回了很远的地方。

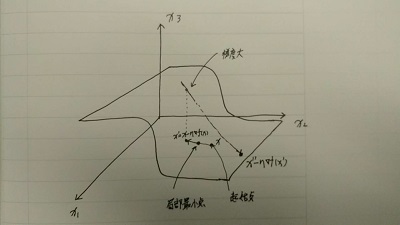


图3-11 “悬崖”对梯度下降法的影响

“峡谷”如图3-12所示。局部极小点在谷底。在峡谷里，梯度下降过程会发生震荡，轻则延缓收敛速度，重则导致不收敛。第4章介绍函数二阶特性后，我们会知道峡谷的成因与赫森矩阵的各个特征值相对大小有关。并且知道在高维情况下局部极小点和局部极大点在理论上是稀少的。大部分驻点是鞍点。

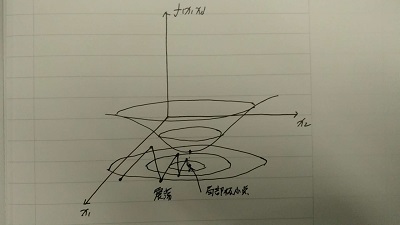


图3-12 “峡谷”对梯度下降法的影响

“广袤的平原”如图3-13所示。在这样的区域里梯度非常小，这将导致收敛缓慢。

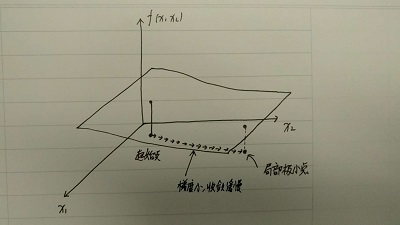


图3-13 “广袤的平原”对梯度下降法的影响

本节例举了原始梯度下降法会遇到的一些（非全部）问题。第4章将介绍一些梯度下降法的改进和变体。

**3.3 运用梯度下降法训练逻辑回归**

运用梯度下降法训练逻辑回归需要首先计算交叉熵损失函数对逻辑回归的参数和的梯度。根据式（3.21），计算梯度需要计算各个偏导数。

交叉熵损失是每一个训练样本上的交叉熵损失之和。第个训练样本的交叉熵损失是：

（3.29）

考虑对的偏导数。分两种情况考虑和。当时：

（3.30）

是的第分量。式（3.30）中的是逻辑回归对的输出。当时：

（3.31）

喜闻乐见的事情发生了，两种情况统一成一种情况：

（3.32）

在继续之前先观察一下式（3.32）。括号中的是真实标签（1/0）与预测概率之差。这个差可以看作模型在样本上的误差。更新参数时加上负梯度，所以会被加到上。是真实值与预测值的差距。这个差距以为权重分配到上。误差分配是看待梯度下降的一个视角。在后文介绍神经网络的反向传播算法时会看到误差不仅在同一层内部分配，还要在层与层之间分配。

回到梯度计算中来。损失函数是对全部训练样本的损失做平均，所以对的偏导数是每一个对的偏导数的平均：

（3.33）

现在考察对的偏导数。类似的计算揭示和两种情况统一到一个表达式：

（3.34）

损失函数对的偏导数就是：

（3.35）

有了各个偏导数就可以计算对和的梯度了：

（3.36）

有了损失函数的梯度，就可以应用梯度下降法训练逻辑回归模型了。逻辑回归模型训练的伪代码如下：

randomly initialized

randomly initialized

while :

**小结：**

本章首先回顾了多元微积分的相关知识，尤其是梯度这个概念以及它的内涵。之后介绍了原始的梯度下降法，讨论了它的种种问题。本章没有涉及改善这些问题的办法。在第4章讨论函数的局部二阶特性后，会介绍一些针对原始梯度下降法的改进措施。

本章最后介绍了如何运用梯度下降法训练逻辑回归模型。阅读完本章，读者应该对梯度下降法和逻辑回归的训练过程有了较透彻的认识。