一、 逻辑回归

1.1 作为一个神经元的逻辑回归

一位大人物曾经说过：“每一个机器学习研究者应该像熟悉自家后院一样熟悉线性模型”。线性模型是许多非线性模型的基础。它良好的数学结构为理解非线性模型提供了深刻的洞见。本书是关于神经网络的，而神经网络的基本组成单元——神经元，正是一个线性模型。

我们以逻辑回归（logistic regression）为例讨论线性模型。假如样本由 n 个数值型特征组成：$x_1,x_2,\cdots,x_n$ ，逻辑回归模型的计算式是：

$$f(x)=\frac{1}{1+\mathrm{e}^{-(b+w_1x_1+w_2x_2+\cdots+w_nx_n)}}=\frac{1}{1+e^{-\left(b+\sum_{i=1}^{n}w_ix_i\right)}} \quad [1.1.1]$$

上式将各个特征 x_i 乘上对应的权重系数 w_i 后相加再加上 b ， b 称作偏置（bias）。这称为对 $x_1,x_2,\cdots,x_n$ 的一个仿射变换（affine transform）。下文用 Σ 表示这个仿射变换的结果 $\Sigma=b+\sum_{i=1}^{n}w_ix_i$ 。仿射变换的几何意义将在下一节介绍。

对 Σ 施加 logistic 函数：

$$\mathrm{logistic}(\Sigma)=\frac{1}{1+e^{-\Sigma}} \quad [1.1.2]$$

logistic 函数的图像是一条 S 型曲线（sigmoid curve）。当 Σ 趋向于负无穷时函数值趋向于 0 ；当 Σ 趋向于正无穷时函数值趋向于 1 。logistic 函数值在 (0,1) 区间内。当 $\Sigma=0$ 时函数值是 0.5 。

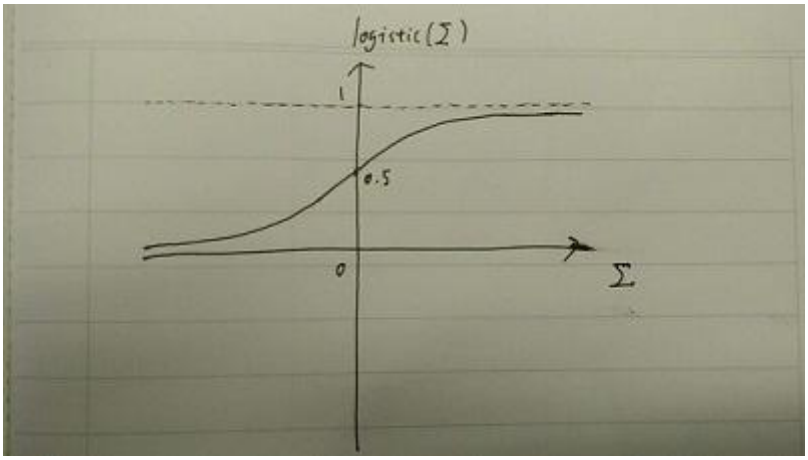

图 1.1.1 logistic 函数的图形

在二分类问题中，用 n 个数值型特征表示一个样本，例如用体重、身高、年龄表示一个人，而且这些样本属于两个互斥的类别——A 类或 B 类。如果要求模型告诉我们某一个样本属于哪一类，则逻辑回归的输出可作为样本属于 A 类的概率：

$$p_A = \frac{1}{1+e^{-\Sigma}} \quad [1.1.3]$$

因为样本必属于 A 类或者 B 类，且不能既属于 A 类又属于 B 类，所以属于 A 类的概率与属于 B 类的概率之和为 1 。于是样本属于 B 类的概率就是：

$$p_B = 1 - p_A = 1 - \frac{1}{1+e^{-\Sigma}} = \frac{e^{-\Sigma}}{1+e^{-\Sigma}} \quad [1.1.4]$$

A 类与 B 类概率之比的对数（以 e 为底）是：

$$log\frac{p_A}{p_B} = log\frac{1}{1+e^{-\Sigma}} \cdot \frac{1+e^{-\Sigma}}{e^{-\Sigma}} = log(e^{\Sigma}) = \Sigma \quad [1.1.5]$$

根据模型给出的概率判定样本类别的决定权在人。如果规定当 A 类的概率大于 0.5 ，即 A 类概率大于 B 类概率时判定为 A 类，那么根据 [1.1.5] 有：

$$\Sigma = log\frac{p_A}{p_b} > log^1 = 0 \quad [1.1.6]$$

以 0.5 为概率阈值时，类别的判定取决于 Σ 是否大于 0 。Σ 大于 0 具有什么几何意义？逻辑回归为什么属于线性模型？将在下文揭晓。在那之前我们先看看逻辑回归的一种图示表示：

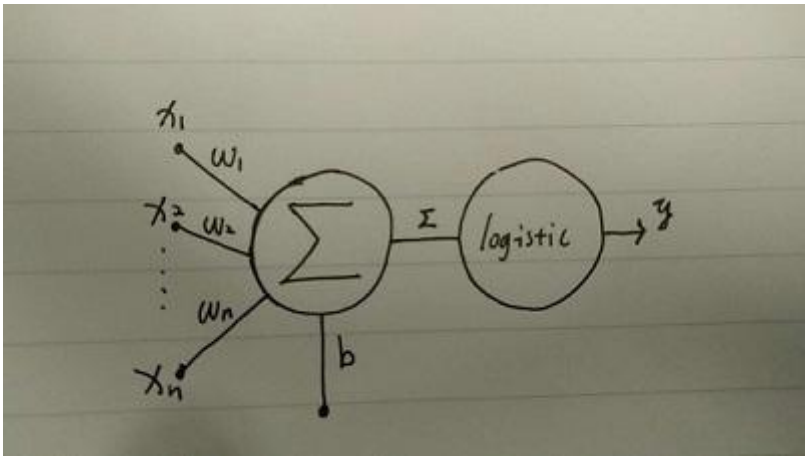

图 1.1.2 逻辑回归的神经元表示

将输入值加权求和再加偏置之后施加 logistic 函数——该图示表达的就是逻辑回归。在人工神经网络（本书以下省略人工二字）语境下，图示的结构就是一个神经元。在这里 logistic 函数被称为激活函数（activation function）。激活函数不限于 logistic 函数。如果激活函数取阶跃函数：

$$f(x) = \begin{cases} 0, \ \Sigma < 0 \\ 1, \ \Sigma \geq 0 \end{cases} \quad [1.1.7]$$

这就是最早的神经元模型——感知机（perceptron）。感知机这个名称一直流传下来，以至于今天多层全连接神经网络（multi-layer full connected network）还被称为 MLP （multi-layer perceptron）。关于激活函数的类型和性质本书后文还有阐述。无论取哪种激活函数，仿射+激活都是神经元的基本结构。

单个神经元是神经网络乃至深度学习的基本砖石。在深入考察神经元的能力和局限之前我们先回顾一下基础的向量几何。

1.2 基础向量几何

1.2.1 向量

一个包含 n 个数值型特征 $x_1,x_2,\cdots,x_n$ 的样本可用一个 n 维向量表示：

$$x=\begin{pmatrix}x_1\\x_2\\\vdots\\x_n\end{pmatrix}=(x_1 \quad x_2 \quad \cdots \quad x_n)^{\mathrm{T}} \quad [1.2.1.1]$$

本书用字母表靠后的斜体小写字母表示向量，例如 x,y 。下标表示向量的分量，例如 x_2 表示向量 x 的第 2 分量。n 是分量的个数。样本有多少特征 x 就有多少分量。本书向量指列向量（分量竖着排列）。有时为了省纸面空间把列向量写成行向量的转置（transpose），如[1.2.1.1] 第二个等号后。转置就是将行向量变成列向量，或将列向量变成行向量：

$$\begin{pmatrix}x_1\\x_2\\\vdots\\x_n\end{pmatrix}^{\mathrm{T}}=(x_1 \quad x_2 \quad \cdots \quad x_n) \text{ and } (x_1 \quad x_2 \quad \cdots \quad x_n)^{\mathrm{T}}=\begin{pmatrix}x_1\\x_2\\\vdots\\x_n\end{pmatrix} \quad [1.2.1.2]$$

本书阐述时用 2 或 3 维维量，因为这利于可视化。在现实中样本的特征数量会远远超过 3 ，但是理论结果可以扩展到任意维度。

如果将各个分量看作坐标值，那么向量 x 表示坐标系上的一个点（point）。 x 也可以看作是从原点指向这个点的一个有长度有方向的“箭头”。这二者都是向量的几何表现形式，如图 1.2.1.1 。论述中根据视角不同采用这两种表现形式。

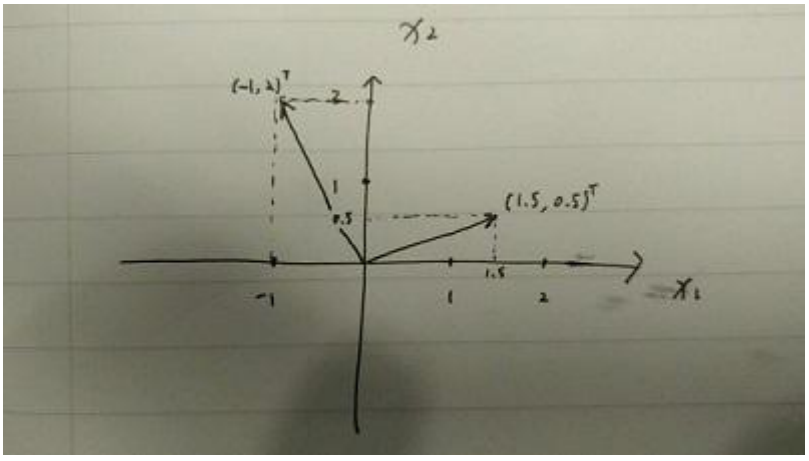

图 1.2.1.1 向量的几何表示

1.2.2 向量的和、数乘与零向量

向量可以求和。令 x 和 y 是两个相同维数的向量，定义它们的和是：

$$x+y=\begin{pmatrix}x_1\\x_2\\\vdots\\x_n\end{pmatrix}+\begin{pmatrix}y_1\\y_2\\\vdots\\y_n\end{pmatrix}=\begin{pmatrix}x_1+y_1\\x_2+y_2\\\vdots\\x_n+y_n\end{pmatrix} \quad [1.2.2.1]$$

即将 x 和 y 的对应分量相加作为它们的和的分量。如果用“箭头”来表示向量，那么向量和是以 x 和 y 为相邻两边组成的平行四边形的对角线，从原点指向相对的顶点。该顶点也是 x 和 y 之和的点表示。

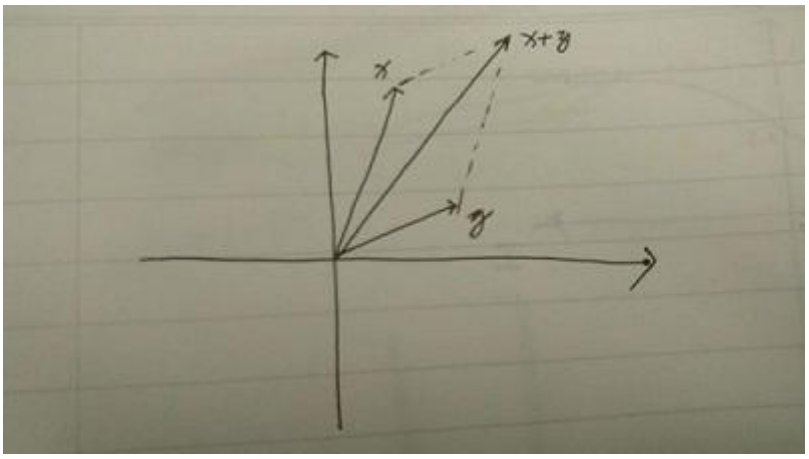

图 1.2.2.1 向量和的几何表示

可以用一个标量（实数）k 去乘一个向量，即向量的数乘。定义为：

$$kx=\begin{pmatrix}kx_1\\kx_2\\\vdots\\kx_n\end{pmatrix} \quad [1.2.2.2]$$

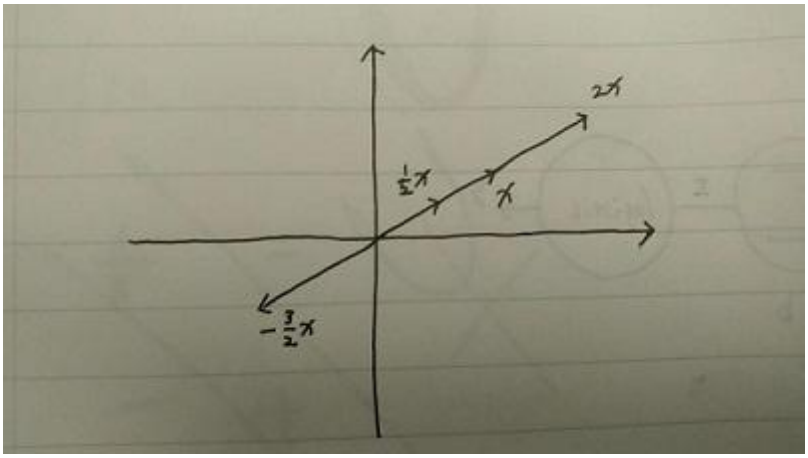

图 1.2.2.2 向量的数乘——缩放

标量乘的几何意义是对向量进行缩放。用 2 乘 向量 x ，保持方向不变，长度变成了 x 的 2 倍。用 0 乘任何向量得到“零向量”：

$$0x = \begin{pmatrix} 0x_1 \\ 0x_2 \\ \vdots \\ 0x_n \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \mathrm{O} \quad [1.2.2.3]$$

用 O 表示零向量。 O 是坐标系原点。O 与任何向量相加都得到该向量本身。如果用 -1 乘向量 x ，有：

$$(-1)x = \begin{pmatrix} -x_1 \\ -x_2 \\ \vdots \\ -x_n \end{pmatrix} = -x \quad [1.2.2.4]$$

显然 $-x$ 与 x 相加得 O 。向量的减法定义为：

$$y - \mathrm{x} = y + (-x) \quad [1.2.2.5]$$

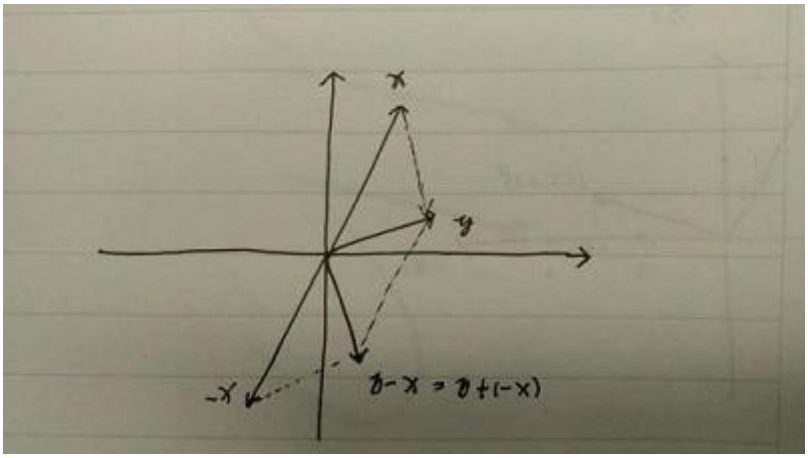

图 1.2.2.3 向量减法图示——三角形的第三边

图中可见，将 $y-x$ 平移，使其尾部与 x 重合，头部与 y 重合。则 $y-x$ 构成以 x 和 y 为临边的三角形的第三边。

向量作为“箭头”是有长度的。这个长度是向量表示的点与原点的距离。对于 2 维向量来说长度是：

$$\text{length}(x) = \sqrt{{x_1}^2 + {x_2}^2} \quad [1.2.2.6]$$

3 维乃至更高维依此类推。

1.2.3 向量的内积、模与投影

向量 x 和 y 的内积（inner product）的定义是：

$$\langle x,y \rangle = \sum_{i=1}^{n} x_i y_i \quad [1.2.3.1]$$

也就是将 x 和 y 的对应分量相乘再加和。 于是 x 与自身的内积就是：

$$\langle x,x \rangle = \sum_{i=1}^{n} {x_i}^2 \quad [1.2.3.2]$$

x 与自身的内积一定大于等于 0 。只有当 x 的所有分量都是 0 ，即 x 为 O 时它与自身的内积为 0 。将向量 x 的模 $\|x\|$ 定义为 x 与自身的内积的平方根：

$$\|x\| = \sqrt{\langle x,x\rangle} = \sqrt{\sum_{i=1}^{n} {x_i}^2} \quad [1.2.3.3]$$

当 n 为 2 时 $\|x\|$ 与 [1.2.2.6] 中定义的向量长度一致。所以可以将任意维向量的“长度”定义为模的平方根。向量 kx 的长度是：

$$\|kx\| = \sqrt{\sum_{i=1}^{n} k^2 {x_i}^2} = \sqrt{k^2 \sum_{i=1}^{n} {x_i}^2} = |k|\|x\| \quad [1.2.3.5]$$

用标量 $1/\|x\|$ 乘 x 得到向量 $x/\|x\|$ 。向量 $x/\|x\|$ 的长度为 1 ，方向与 x 一致。也就是将 x 缩放到长度 1 。长度为 1 的向量称为单位向量（unit vector）。

对于 2 维或 3 维的情况，上节提到 $y-x$ 构成以 x 和 y 为临边的三角形的第三边。$y-x$ 的模就是该三角形第三边的长度。三角形第三边长度可用余弦公式求得：假如三角形三条边长是 a,b,c 。 a 和 b 之间的夹角是 θ ，则 c 是：

$$c^2 = a^2 + b^2 - 2ab\cos\theta \quad [1.2.3.6]$$

当 θ 为 $\pi/2$ 时 [1.2.3.6] 就是毕达哥拉斯定理。将三角形的边长代换为向量长度，有：

$$\|y-x\|^2 = \|x\|^2 + \|y\|^2 - 2\|x\|\|y\|\cos\theta \quad [1.2.3.7]$$

根据模的定义将上式展开并消除等号两边相同项，得到：

$$\langle x,y\rangle = \|x\|\|y\|\cos\theta \quad [1.2.3.8]$$

向量 x 和 y 的内积等于它们的长度之积再乘以它们之间夹角的余弦。在 2 或 3 维上向量之间的夹角有直观的定义，运用余弦定理得到 [1.2.3.8] 。在更高维度上向量的夹角反过来由 [1.2.3.8] 定义。如果向量 x 和 y 的内积为 0 ，则它们之间夹角的余弦是 0 ，则夹

角 θ 是 $\pi/2$ 。在这种情况下称 x 和 y 正交（orthogonal）。若 x 和 y 都不是 O ，则它们的“箭头”互相垂直。O 与任何向量正交。

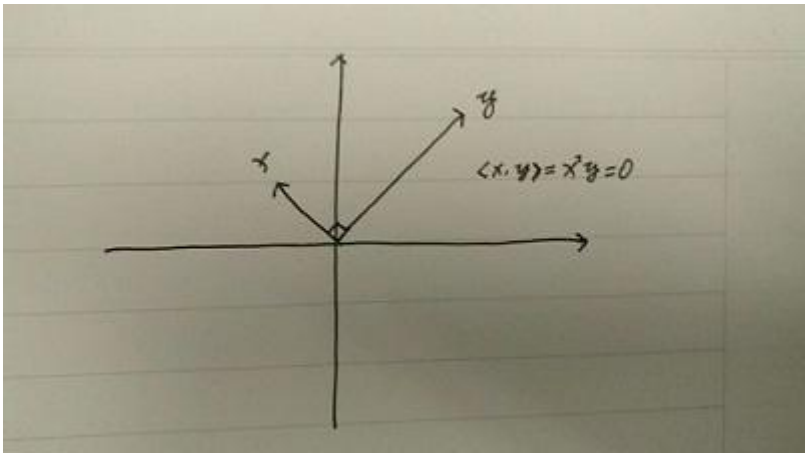

图 1.2.3.1 向量正交

如果 x 和 y 之间夹角为 θ ，那么 $\|x\|\cos\theta$ 是 x 向 y 方向的投影长度，如图 1.2.3.2 。它等于 $\langle x,y\rangle/\|y\|$ 。如果 y 是单位向量，则 x 向 y 方向的投影长度就是 $\langle x,y\rangle$ 。

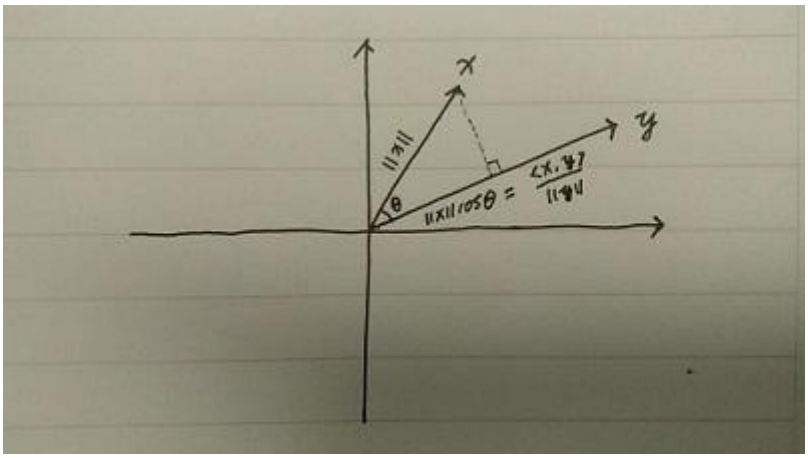

图 1.2.3.2 向量投影

如果把列向量看作 $n \times 1$ 矩阵，向量的内积可以用矩阵乘积表示：

$$\langle x,y\rangle = \sum_{i=1}^{n} x_i y_i = (x_1 \quad x_2 \quad \cdots \quad x_n)\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = x^{\mathrm{T}} y \quad [1.2.3.9]$$

1.2.4 线性空间、基和线性变换

对任何一个 n 维向量 x 可以进行如下分解：

$$x=\begin{pmatrix}x_1\\x_2\\\vdots\\x_n\end{pmatrix}=x_1\begin{pmatrix}1\\0\\\vdots\\0\end{pmatrix}+x_2\begin{pmatrix}0\\1\\\vdots\\0\end{pmatrix}+\cdots+x_n\begin{pmatrix}0\\0\\\vdots\\1\end{pmatrix}=\sum_{i=1}^{n}x_ie_i \quad [1.2.4.1]$$

每一个 e_i 是一个 n 维向量，其第 i 个分量是 1 ，其余分量都是 0 。称 x 可以被 $e_{i,\ i=1\ldots n}$ 线性表出。任意 n 维向量都可以被 $e_{i,\ i=1\ldots n}$ 线性表出。而 $e_{i,\ i=1\ldots n}$ 中的任何一个向量 e_j 都无法由其他 $\mathrm{e}_{i,\ i\neq j}$ 线性表出。这是因为除了 e_j 其他基向量的第 j 个分量都是 0 。这种情况称 $e_{i,\ i=1\ldots n}$ 线性独立。

全体 n 维向量构成 n 维线性空间。非严格地说，线性空间是向量的集合，其中全部向量都可以由一组线性独立的向量线性表出。这组向量是该线性空间的一组基。基向量的个数是该线性空间的维数。 $e_{i,\ i=1\ldots n}$ 是 n 维线性空间的一组基（basis）。或者说 $e_{i,\ i=1\ldots n}$ 张成了 n 维线性空间。

$e_{i,\ i=1\ldots n}$ 称为 n 维线性空间的标准基。除标准基外，任何 n 个线性无关的向量 $b_{i,\ i=1\ldots n}$ 都能构成 n 维空间的一组基。$b_{i,\ i=1\ldots n}$ 是线性独立的，则其中任一个 b_j 都无法由 $\mathrm{b}_{i,\ i\neq j}$ 线性表出。b_j 可以分解成两个向量之和 $b_j=b_j^1+b_j^2$ 。其中 b_j^1 可由 $b_{i,\ i\neq j}$ 线性表出，而 b_j^2 正交（垂直于）每一个 $b_{i,\ i\neq j}$ 。 b_j 提供了其他 $b_{i,\ i\neq j}$ 无法提供的某个方向的信息，为构成整个 n 维空间做出了它独有的贡献。

现在介绍线性变换（linear transform）。如果变换 f 是线性的，则对于任两个向量 x 和 y 以及任两个实数 a 和 b 有：

$$f(ax+by)=af(x)+bf(y) \quad [1.2.4.2]$$

如果 f 是线性变换，则必存在一个向量 w ，对任意向量 x 有：

$$f(x)=w^{\mathrm{T}}x \quad [1.2.4.3]$$

这样构造 w ：

$$w = \begin{pmatrix} f(e_1) \\ f(e_2) \\ \vdots \\ f(e_n) \end{pmatrix} \quad [1.2.4.4]$$

w 的第 i 分量是对标准基第 i 个向量 e_i 施加 f 得到的值。由于 f 是线性变换，有：

$$f(x) = f\left(\sum_{i=1}^{n} x_i e_i\right) = \sum_{i=1}^{n} x_i f(e_i) = (f(e_1) \quad f(e_2) \quad \cdots \quad f(e_n)) \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = w^{\mathrm{T}} x \quad [1.2.4.5]$$

这就证明了 [1.2.4.3] 的结论。线性变换的图像一定过原点，因为：

$$f(O) = f(0x) = 0f(x) = 0 \quad [1.2.4.6]$$

1.2.5 直线、（超）平面与仿射变换

对于任意 $w \neq 0$ ，令 x 为任意与 w 内积是常数 c 的向量：

$$w^{\mathrm{T}} x = \langle w,x \rangle = \|w\| \|x\| \cos\theta = c \quad [1.2.5.1]$$

也就是说 x 满足 $\|x\| \cos\theta = c/\|w\|$ 。即 x 向 w 方向的投影是常数 $c/\|w\|$。如果 x 是 2 维的，将其看作二维空间中一个点，则这样的点都位于垂直于 w 的一条直线上。

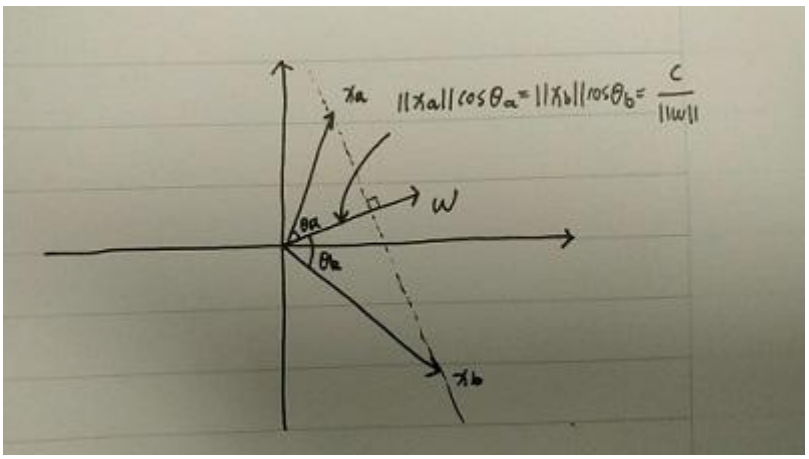

图 1.2.5.1 二维空间中与特定非零向量内积相同的所有点位于垂直于该向量的直线上

在 3 维空间中与某 $w \neq O$ 内积为常数的点构成垂直于 w 的平面。在更高维空间中这样的点构成垂直于 w 的超平面（hyperplane）。

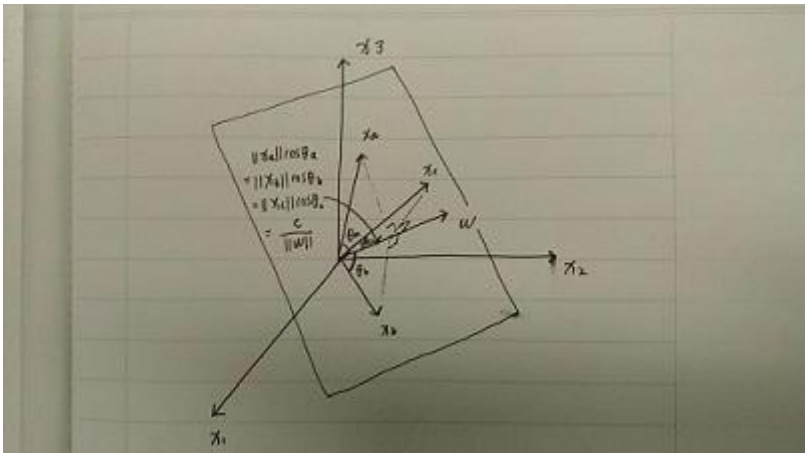

图 1.2.5.2 三维空间中与特定非零向量内积相同的所有点位于垂直于该向量的平面上

w 称为直线/（超）平面的法向量（norm）。对于任意非 0 实数 k ， kw 也是同一个直线/（超）平面的法向量。因为该直线/（超）平面上任意向量与 kw 的内积也为常数：

$$(kw)^{\mathrm{T}}x = \langle kw,x\rangle = |k|\|w\|\|x\|\cos\theta = |k|c \quad [1.2.5.2]$$

[1.1] 节曾提到仿射变换，将其重新写在这里：

$$y = b + \sum_{i=1}^{n} w_i x_i \quad [1.2.5.3]$$

如果自变量是 2 维，将因变量 y 移到等号另一侧得到：

$$w_1x_1 + w_2x_2 - y = \begin{pmatrix} w_1 & w_2 & -1 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ y \end{pmatrix} = -b \quad [1.2.5.4]$$

在 x_1x_2y 三维空间中所有满足式 [1.2.5.3] 的点与 $w = (w_1 \quad w_2 \quad -1)$ 的内积为常数 $-b$ 。所以该仿射变换的图像是 3 维空间中一张平面。该平面法向量是 w 。当 x_1 和 x_2

等于 0 时 y 的值是 b ，称为该平面的截距。可以看出仿射变换是线性变换加上一个常量 b 。线性变换的图像是截距 $b=0$ 的直线\（超）平面，它过原点。

若 w_1 和 w_2 的绝对值较大，则 w 更贴近 x_1x_2 平面。以 w 为法向量的平面较陡。反之若 w_1 和 w_2 的绝对值较小，则 w 更贴近 y 轴，以 w 为法向量的平面较平。平面的倾斜程度与 w_1 和 w_2 的绝对值大小有关，也就是与 w 的模的大小有关。

$(w_1 \quad w_2 \quad 0)$ 是 w 在 x_1x_2 平面上的投影。它的方向决定了平面的朝向。总之， $w=(w_1 \quad w_2 \quad -1)$ 是 2 维仿射变换的图像（3 维空间中一张平面）的法向量。 $(w_1 \quad w_2)$ 的模决定了平面的倾斜程度，方向决定了平面的朝向。下图集中展示这几个概念：

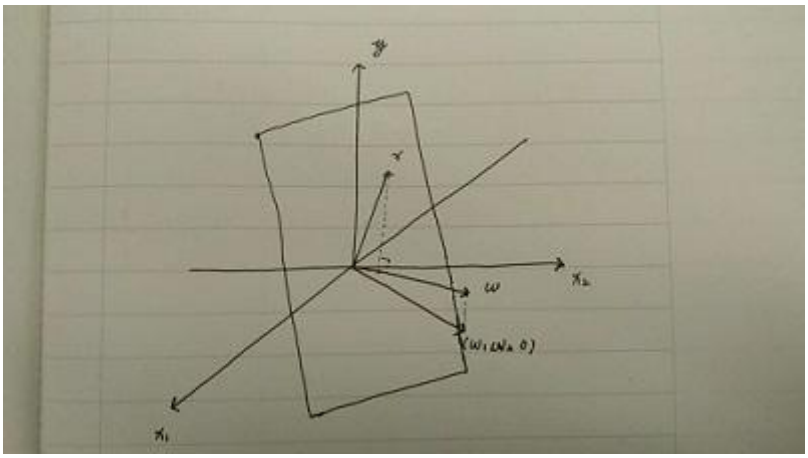

图 1.2.5.3 法向量决定平面的朝向和倾角

仿射变换是一类最简单的变换。仿射变换的图像在自变量为 1 维的情况下是直线，在自变量为 2 维的情况下是平面，在更高维情况下是超平面。这类图像在各处的特性是相同的。想象一个平坦的斜坡。无论从斜坡上的哪一点出发，向同样的方向走都是同样的倾斜程度。

仿射函数是更复杂的函数在某一点局部的最粗糙的近似，称一阶近似。训练逻辑回归或神经网络的梯度下降法就是一种基于损失函数局部一阶近似特性的优化算法。这些内容本书后文将详加介绍。

1.3 从几何角度理解逻辑回归的能力和局限

回忆一下逻辑回归的表达式：

$$f(x)=\frac{1}{1+e^{-\left(b+\sum_{i=1}^{n} w_i x_i\right)}} \quad [1.3.1]$$

用我们现在的语言表述，它是对输入向量 x 做仿射变换后再施以 logistic 函数。还是以 2 维为例，某条以 $(w_1 \quad w_2)$ 为法向量的直线上的点与 $(w_1 \quad w_2)$ 的内积为常量。所以对它们施加仿射变换 $b+\sum_{i=1}^{2} w_i x_i$ 后的值都相同，再施加 logistic 函数后的结果也都相同。逻辑回归先对输入用某个法向量 $(w_1 \quad w_2)$ 做仿射变换，这样就丢弃了垂直于该法向量方向上的信息。

于是逻辑回归的输出只在 $(w_1 \quad w_2)$ 的方向上有变化，在垂直于 $(w_1 \quad w_2)$ 的方向上无变化。如果用阈值 t 去卡这个输出：

$$output(\mathrm{x}) = \begin{cases} 0, & f(x) < 0 \\ 1, & f(x) \geq 0 \end{cases} \quad [1.3.2]$$

则只能得到垂直于 $(w_1 \quad w_2)$ 的直线分界线。这也就是逻辑回归属于线性模型的原因。仿射变换是“罪魁祸首”。它先对信息进行了过滤。即使之后再施以 S 形的 logistic 函数也于事无补：

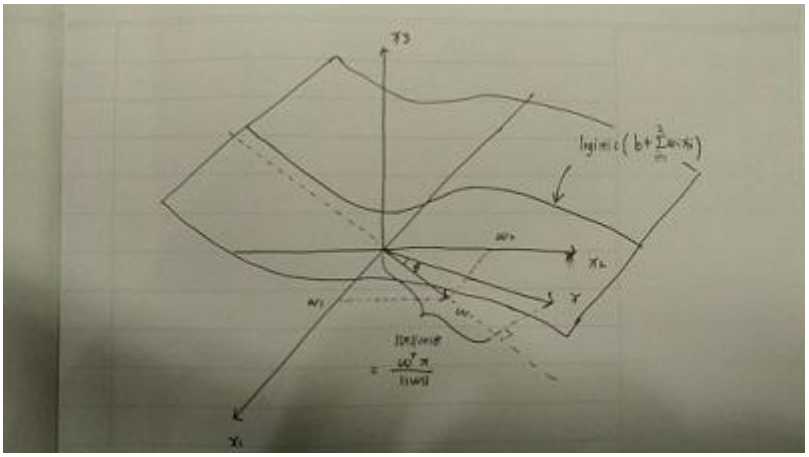

图 1.3.1 单个神经元只能形成直线/超平面分界线/面

神经元可以选用其他激活函数。但只要激活函数仍是单调的，则只能形成一条直线分界线（在更高维情况下就是超平面分界面）。单调激活函数神经元无法处理异或问题（XOR）。因为异或问题是线性不可分的——无法用直线将正负样本点分隔开：

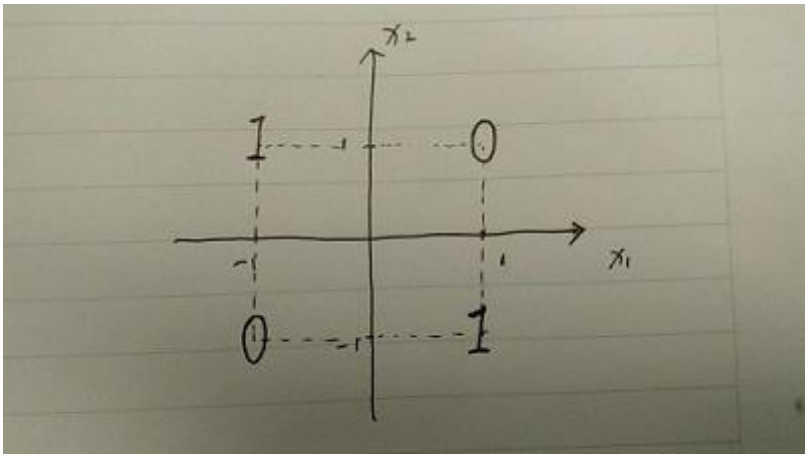

图 1.3.2 异或问题

如果使用非单调函数作为激活函数，比如平方函数：

$$f(x) = \left(b + \sum_{i=1}^{n} w_i x_i\right)^2 \quad [1.3.3]$$

通过调整 $(w_1 \quad w_2)$ 的方向，[1.3.3] 可以解决异或问题：

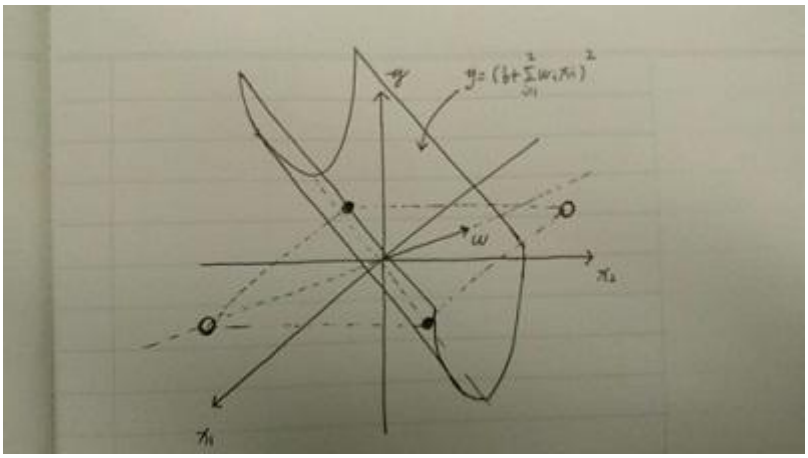

图 1.3.3 使用平方函数激活函数的单个神经元解决异或问题

但是 [1.3.3] 的能力也仅限于用两条平行直线去分割自变量空间。就算使用正弦 sin 函数也只能是形成无数条平行分割线。它们对于例如同心圆状分布的数据是无能为力的：

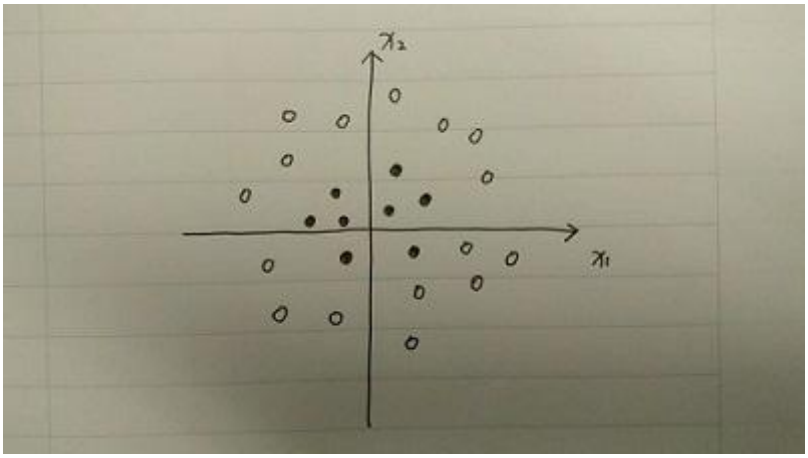

图 1.3.4 同心圆状数据

同心圆数据正类点在中央，负类点围绕在外围。在 x_1x_2 平面内不存在一条直线将正负两类点分开。该数据含有各个方向的分类信息，而单个神经元只取一个方向的信息，所以无法用一个神经元分开正负两类样本点。要想获得超越线性的能力，必须将多个神经元连接成网络。允许多个神经元合作形成非线性分界线。这是本书第二部分的内容，这里暂且不表。

本章介绍了作为线性模型的逻辑回归模型。它同时也是一个神经元。在回顾了必要的向量几何背景知识后，我们从几何角度阐述了逻辑回归的能力及其限制。我们知道逻辑回归如何划分自变量空间取决于权值向量 w 和偏置 b 。但本章没有阐述对于某一个特定分类问题如何确定 w 和 b 的值。这就是逻辑回归模型的训练问题。