

# 线性神经网络

## 一、线性回归

<https://promhub.top/%E7%BA%BF%E6%80%A7%E5%9B%9E%E5%BD%92.html>

## 二、逻辑回归

在逻辑回归中可以通过一条直线来划分两种不同的类型，以此可以解决二分类问题。在逻辑回归中通过表示为：

$$\hat{y} := \begin{cases} 1 & \text{if } f(x) > 0.5 \\ 0 & \text{if } f(x) \leq 0.5 \end{cases} \quad (2-1)$$

以输出值为 0.5 作为界限进行区分。

拟合的方程  $f(x)$  写为：

$$f(x) = w \cdot x \quad (2-2)$$

与线性回归相似，在单特征时有一个特征  $w$ ，一个输入  $w$ ，一个输出  $y$ ，偏置  $b$ 。

这样进而就可以由  $f(x)$  写出一个  $x$  关于  $\hat{y}$  的表达式。

$$\hat{y} = \begin{cases} 1 & \text{if } x > f^{-1}(0.5) \\ 0 & \text{if } x \leq f^{-1}(0.5) \end{cases} \quad (2-3)$$

这样就可以通过输入  $x$  得出输出  $y$ 。

这个函数是间断的，不适用于使用梯度下降法进行处理，这样就可以使用更加平滑的 sigmoid 函数来代替。sigmoid 函数可以写为

$$f(z) = \frac{1}{1 + e^{-z}} \quad (2-4)$$
$$z = w \cdot x$$

### 2.1 极大似然估计

概率描述的是一个事件发生的可能性，似然性是放过来在事件发生后反推在什么参数条件下，这个事件发生的概率最大（例如下雨（果）的时候有乌云（因/证据/观察的数据）的概率，即已经有了果，对证据发生的可能性描述）。数学语言可以将其写为：

$$\begin{aligned} P(x|\beta) &= a \\ L(\beta|x) &= a \\ P(x|\beta) &= L(\beta|x) \end{aligned} \quad (2-5)$$

二者是相等的。

在二分类问题中可以概率写为：

$$\begin{aligned} P(y|x, \beta) &= P(y=1)^y P(y=0)^{1-y} \\ P(y|x, \beta) &= \frac{1}{1 + e^{-z}}^y \left(1 - \frac{1}{1 + e^{-z}}\right)^{1-y} \end{aligned} \quad (2-6)$$

。

得到最大值：

$$L(\beta) = \prod_{i=1}^n P(y_i|x_i, \beta) = \prod_{i=1}^n \frac{1}{1 + e^{-w \cdot x_i}}^{y_i} \left(1 - \frac{1}{1 + e^{-w \cdot x_i}}\right)^{1-y_i} \quad (2-7)$$

对其取对数可得：

$$\log(L(\beta)) = \sum_{i=1}^n \left( y_i \cdot \log\left(\frac{1}{1 + e^{-w \cdot x_i}}\right) + \log\left((1 - y_i) \cdot \left(1 - \frac{1}{1 + e^{-w \cdot x_i}}\right)^{1-y_i}\right) \right) \quad (2-8)$$

## 2.2 损失函数

损失函数是用于衡量预测值与实际值的偏离程度，即模型预测的错误程度。通过寻找误差最小时的参数来确定出最佳的模型。

使用平方损失：

$$Q = \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (2-9)$$

待入到逻辑回归中损失函数就是：

$$Q = \sum_{i=1}^n \left( y_i - \frac{1}{1 + e^{-w \cdot x}} \right) \quad (2-10)$$

因为这个函数不是凸函数，所以使用对数损失函数。

$$Q = -\log(L(\beta)) \quad (2-11)$$

## 三、多元线性回归

多元线性回归相比于线性回归，特征与输入多余线性回归。形式为：

$$\hat{y} = w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4 + \dots + b \quad (3-1)$$

可以将其写成矩阵的形式： $(x_1 \ x_2 \ x_3) \cdot \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} + (b) = w_1 x_1 + w_2 x_2 + w_3 x_3 + b$  相当于是一个点乘。

进而可以用矩阵  $X = \begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{pmatrix}$  来表示数据集，一行表示一组数据，一列对应与某特征值。使用矩阵  $W = (w_1 \ w_2 \ w_3)$  来表示特征，偏置可以表示为： $(b)$ 。这样计算矩阵：

$$\hat{y} = X \times W^T + b = \begin{pmatrix} (x_{11} \ x_{12} \ x_{13}) \cdot \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} + b \\ (x_{21} \ x_{22} \ x_{23}) \cdot \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} + b \\ (x_{31} \ x_{32} \ x_{33}) \cdot \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} + b \end{pmatrix} \quad (3-2)$$

使用平方损失函数：

$$Q = \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (3-3)$$

## 四、Softmax 回归

在前面的逻辑回归，它主要应用与二分类问题，对于多分类问题，有多个输入，多个特征，多个输出。

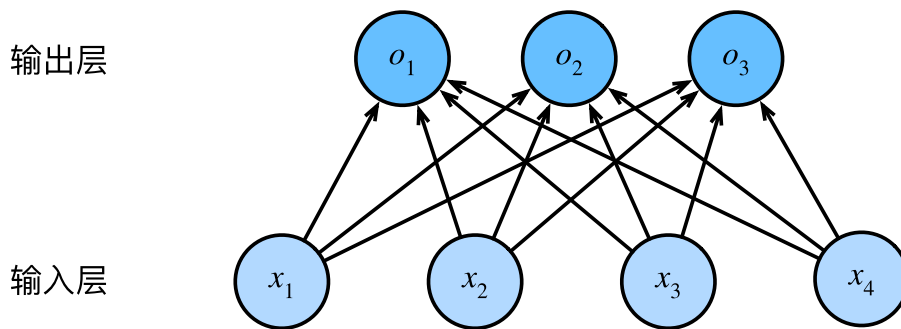


Figure 4 - 1: 网络图

可以表示为：

$$\begin{aligned} o_1 &= x_1 w_{11} + x_2 w_{12} + x_3 w_{13} + b_1 \\ o_2 &= x_1 w_{21} + x_2 w_{22} + x_3 w_{23} + b_2 \\ o_3 &= x_1 w_{31} + x_2 w_{32} + x_3 w_{33} + b_3 \end{aligned} \quad (4 - 1)$$

使用三个线性的表达式表示。

将其写成矩阵形式，可以表示为：

$$\begin{pmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

$$\begin{pmatrix} (x_1 \ x_2 \ x_3) \cdot \begin{pmatrix} w_{11} \\ w_{12} \\ w_{13} \end{pmatrix} + b_1 \\ (x_1 \ x_2 \ x_3) \cdot \begin{pmatrix} w_{21} \\ w_{22} \\ w_{23} \end{pmatrix} + b_2 \\ (x_1 \ x_2 \ x_3) \cdot \begin{pmatrix} w_{31} \\ w_{32} \\ w_{33} \end{pmatrix} + b_3 \end{pmatrix} \quad (4 - 2)$$

在这个例子中输入的个数为 3，输出的个数为 3，特征的个数为 3，可以简写为  $O = XW^T + b$ 。

因为由上述的线性表达式得出的结果可能为正可能为负，三个输出之和也可能大于 1，这样都会违背概率的定理，所以使用 softmax 函数来规范得到的结果。Softmax 函数可以写为：

$$\begin{aligned} \hat{y} &= \text{softmax}(o) \\ \hat{y}_i &= \frac{\exp(o_i)}{\sum_k \exp(o_k)} \end{aligned} \quad (4 - 3)$$

这样就可以得到  $\hat{y}_1 = \frac{\exp(o_1)}{\sum_k \exp(o_k)}$ ,  $\hat{y}_2 = \frac{\exp(o_2)}{\sum_k \exp(o_k)}$ ,  $\hat{y}_3 = \frac{\exp(o_3)}{\sum_k \exp(o_k)}$ 。可以取  $\text{argmax}_i(o_i) = \text{argmax}_i(\hat{y}_i)$  作为输出值。

可以得出 softmax 回归的计算表达式为：

$$\begin{aligned} o^i &= x^i w + b \\ \hat{y}^i &= \text{softmax}(o_i) \end{aligned} \quad (4 - 4)$$

#### 4.1 交叉熵

交叉熵可以表示为：

$$H(y^i, \hat{y}^i) = - \sum_{i=1}^n P(x_i) \log(P(x_i)) \quad (4 - 5)$$

$p(x_i)$ 代表随机事件 $x_i$ 的概率。

信息量是对信息的度量。接受到的信息量跟具体发生的事件有关。越小概率的事情发生了产生的信息量越大，越大概率的事情发生了产生的信息量越小。有 2 个不相关的事件  $x$  和  $y$ ，那么我们观察到的俩个事件同时发生时获得的信息应该等于观察到的事件各自发生时获得的信息之和 $h(x, y) = h(x) + h(y)$ ，由于  $x, y$  是俩个不相关的事件，那么满足  $p(x, y) = p(x) \times p(y)$ 这样就得出信息量的公式：

$$h(x) = -\log_2(p(x)) \quad (4-6)$$

信息熵是在结果出来之前对可能产生的信息量的期望——考虑该随机变量的所有可能取值，即所有可能发生事件所带来的信息量的期望。就可以用前面的公式 4-5 表示。

## 4.2 交叉熵损失函数

交叉熵损失函数可以定义为：

$$l(\theta) = \frac{1}{n} \sum_{i=1}^n H(y^i, \hat{y}^i) \quad (4-7)$$

## 五、引用

1. 《3.4. softmax 回归 — 动手学深度学习 2.0.0 documentation》. 见于 2024 年 5 月 4 日. [https://zh.d2l.ai/chapter\\_linear-networks/softmax-regression.html](https://zh.d2l.ai/chapter_linear-networks/softmax-regression.html).
2. 《3.4. softmax 回归 — 〈动手学深度学习〉 文档》. 见于 2024 年 5 月 4 日. [https://zh-v1.d2l.ai/chapter\\_deep-learning-basics/softmax-regression.html](https://zh-v1.d2l.ai/chapter_deep-learning-basics/softmax-regression.html).
3. 《多元线性回归模型（公式推导+举例应用）-CSDN 博客》. 见于 2024 年 5 月 4 日. [https://blog.csdn.net/qq\\_46117575/article/details/135450470](https://blog.csdn.net/qq_46117575/article/details/135450470).
4. 《一文详解 Softmax 函数 - 知乎》. 见于 2024 年 5 月 4 日. <https://zhuanlan.zhihu.com/p/105722023>.
5. 《用人话讲明白逻辑回归 Logistic regression - 知乎》. 见于 2024 年 5 月 4 日. <https://zhuanlan.zhihu.com/p/139122386>.
6. 知乎专栏. 《通俗理解信息熵》. 见于 2024 年 5 月 4 日. <https://zhuanlan.zhihu.com/p/26486223>.