

King of Dumbs

笨蛋之王

级别: ♦ 进阶

逻辑回归还是神经网络？ 其实是同一个神经元

逻辑回归就是只有一个神经元的神经网络。深度学习多了什么、什么时候值得用、什么时候经典模型就够了，分三个层次讲清楚。

中文 发布于 2026年10月4日



扫码打开可互动的课程

<https://kingofdumbs.com/zh/lessons/logistic-regression-or-neural-network>

你将学到

- 逻辑回归
- 神经网络
- sigmoid 函数
- 梯度消失
- 在经典机器学习和深度学习之间做选择

本课会出现的符号

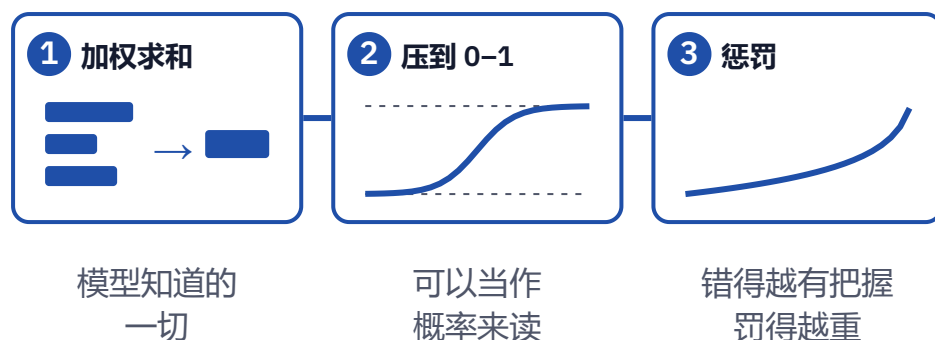
E_a 下标: 右下角的标签 (♠) · Σ 求和: 全部加起来 (♠) · x_i 下标 i : 第 i 个元素 (♠) · Π 连乘: 全部乘起来 (♠) · $\approx$ 约等于 (♠) · $\sim$ 数量级约为 / 服从...分布 (♠) · $\leq \geq$ 小于等于 / 大于等于 (♠) · $|x|$ 绝对值: 去掉正负号 (♠) · 10^{-3} 负指数: 表示除 (♠) · $h^{(l)}$ 带括号的上标: 不是乘方 (♠) · e 数 e (≈ 2.718) (♠) · $\log$ 对数: 要乘几次? (♠) · σ sigma: 含义因课而异 (♠) · λ lambda: 一个缩放用的数 (♠) · δ 小写 delta: 小误差 (♠) · $\|x\|$ 范数: 箭头的长度 (♠) · x^T 转置: 把列放倒 (♠) · $\in$ 属于 (♠) · $\mathbb{R} \mathbb{Z}$ 数集 (♠) · dE/dt 导数: 变化的快慢 (♠) · ∂ 偏导数: 只改变一个量 (♠) · ∇ 梯度: 上升最快的方向 (♠)

第一次看到画出来的神经网络，一堆小圆圈和箭头，我以为它和传统统计学是两回事。后来我得亲手写一个逻辑回归，于是冒出了这节课的笨问题：**逻辑回归是不是神经网络？**

简短的回答是：是的。逻辑回归就是只有一个**神经元**的神经网络。它们不是长得像，也不是亲戚，而是同一个模型，只是有两个名字。既然是同一个神经元，真正有意思的问题就变了：大型网络到底多了什么？什么时候值得为它付出代价？

选一张牌吧：♣ 适合从零开始，♦ 适合知道函数或平均数是什么的人，♠ 适合想看数学记法的人。随时可以换。

一个神经元，两个名字



逻辑回归由三个部分组成。值得一个个弄清楚，因为每个部分都对应一种不同的失败方式。

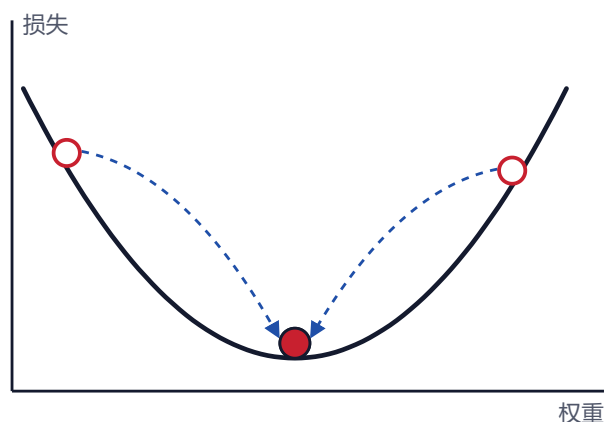
1. 加权求和。 数据的每一列乘以一个数，也就是它的权重，再全部加起来，外加一个常数。我们把结果叫作得分。模型知道的一切都在这里：如果信号藏在两列的组合里，或者藏在一条曲线里，除非你事先把它构造好交给模型，否则模型看不到。

2. sigmoid。 这是一条 S 形曲线，把任何得分压到 0 和 1 之间：很负的得分接近 0，很正的接近 1，得分为零时正好是 0.5。选它不是为了好看：假设答案是一枚做了手脚的硬币，而手脚的大小取决于一个加权求和，推出来的就是这个形状。所以它的输出可以当作概率来读。

3. 惩罚。 它把概率和正确答案作比较，重罚“很有把握却错了”。行话叫损失，这一种具体叫对数损失或交叉熵。为了学习，模型会看惩罚往哪个方向下降，然后把权重往那个方向挪一点，一遍又一遍：这就是梯度下降。

接下来是改变一切的性质：对逻辑回归来说，惩罚**只有一个底**。不管怎么往下走，都会到达同一个地方，所以随机种子无关紧要。带隐藏层的网络就没有这个保证了：它有很多山谷，最后落在哪里取决于从哪里出发。

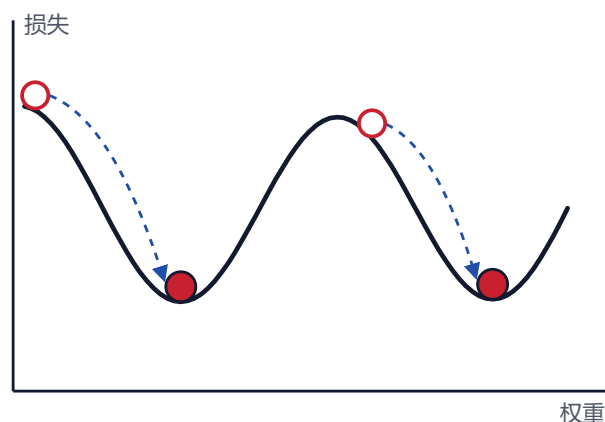
逻辑回归：只有一个底



从哪里出发都到同一处

示意形状，不是数据。重要的是有几个底。

深度网络：好几个山谷



取决于从哪里出发

我验证过：scikit-learn 的逻辑回归和手工训练的一个神经元，得到的权重一直到小数点后第三位都相同。一个实用细节：很多库默认会加一个把权重往小里压的“刹车”，如果你想解读权重，就把它关掉。

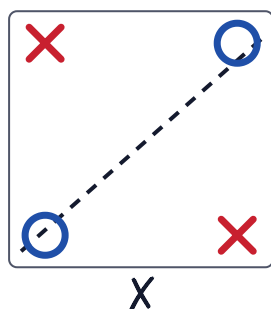
笨蛋之王说：

我曾经纠结了一个星期，以为必须在统计和深度学习之间二选一。结果发现，我早就在做只有一个神经元的深度学习了。

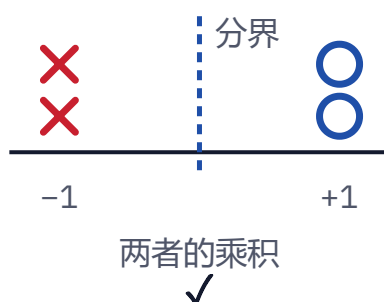


谁来制造特征

没有一条直线
能分开它们



用两者的乘积
就能分开



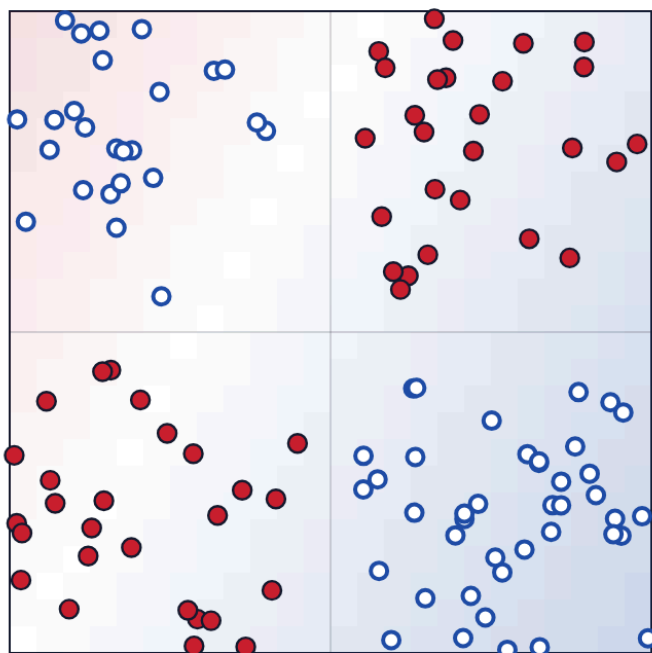
既然最后那个神经元是一样的，深度网络多出来的就是**这个神经元之前发生的事**。一个回答“是或否”的十层网络，它的最后一层就是一个逻辑回归。区别在于，它的输入不是你的列，而是**特征**：隐藏层从数据中学到的新组合。

经典的例子是异或：正方形四个角上的四个点，每条对角线上的类别相同、两条对角线的类别相反。一个神经元只会画直线，而没有任何直线能把它们分开。但如果给它一系列新数据，**两列的乘积**，这些点就自动分开了。这一列新数据就是一个特征：要么你手工构造，要么由隐藏层来构造。

这引出了我知道的最有用的经验法则。如果你的数据是表格，而且每一列本来就有含义，好的特征已经存在，隐藏层能补充的很少。如果是像素、音频或文本，任何像素的加权和都看不出一只猫：总得有人来制造特征。制造者可以是你的数据训练的隐藏层，可以是几百棵把票数加起来的树（在表格上叫**梯度提升**），也可以是一个已经用全世界数据训练好的模型，你只需要调整它的末端。

一个实用细节：在深度学习库里，网络输出的是没有压缩的得分，sigmoid 放在惩罚的计算里面，这样遇到很大的数也不会算崩。

自己试试。异或和圆环是两个经典例子：没有任何一条直线能把这些点分开。选一组数据和一个模型，然后训练：



● 第 1 类 ○ 第 0 类

数据点

直线

异或

圆环

模型

1 个神经元 (逻辑回归)

3 个参数

1 个神经元 + 手工特征:

4 个参数

$x_1 \cdot x_2$

8 个神经元的隐藏层 + 1

个神经元

33 个参数

训练

重置

权重: $w = (-0.44; 0.24) \cdot b$
 $= -0.18$

准确率: 43% (永远猜最常见的类别: 55%)。一个神经元只会画直线, 而没有任何直线能分开异或。准确率和抛硬币差不多。

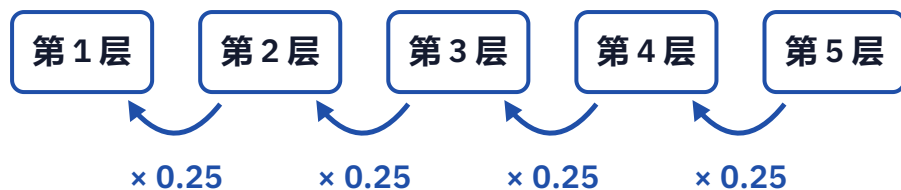


互动小工具。 在网站上试一试: <https://kingofdumbs.com/zh/lessons/logistic-regression-or-neural-network>

这个实验里我最喜欢的是中间那一行。在异或上失败的同一个神经元, 只要手工给它两个坐标的乘积, 马上就全对了。隐藏层不需要任何人告诉它该找什么, 也能到达同样的地方, 代价是要调 33 个数而不是 4 个, 而且你再也读不懂它学到了什么。

sigmoid 为什么退出了隐藏层

误差信号从最后一层传回第一层



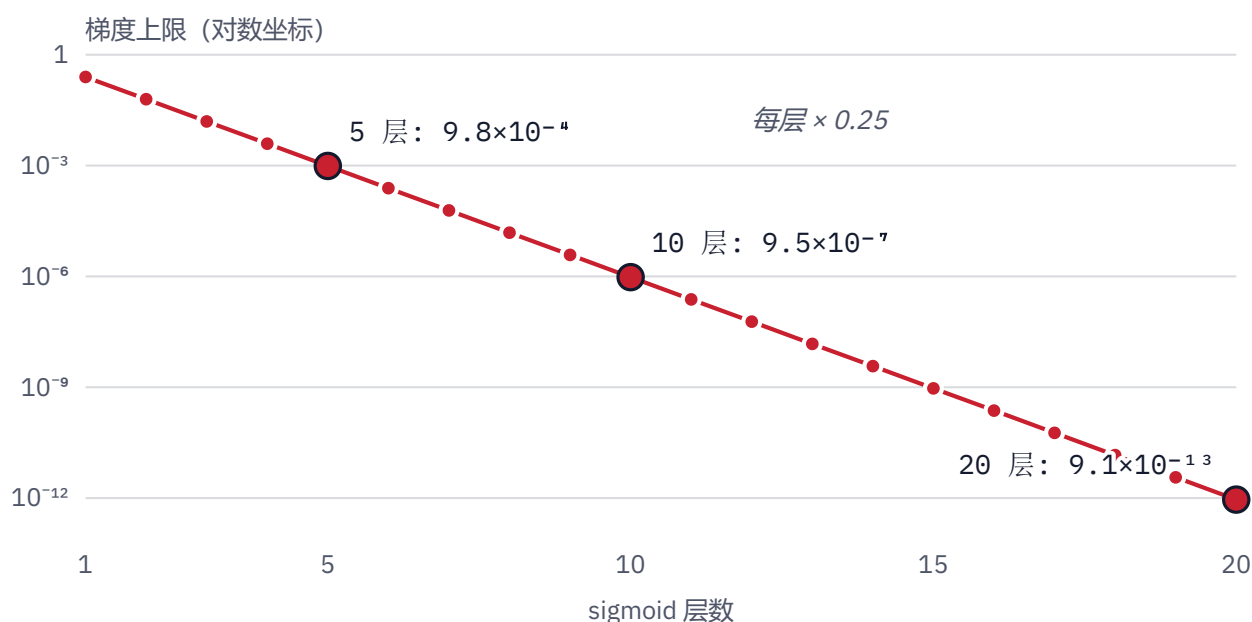
到达 ≈ 0.001

出发 1

误差

如果网络就是“很多逻辑回归神经元叠在一起”，那每层都用 sigmoid 似乎理所当然。很多年里大家确实这么做，结果并不好。

原因在于 sigmoid 的**斜率**。斜率最大只有 0.25，出现在 S 的正中间；到了两端，曲线已经平了，斜率几乎为零。训练网络时，告诉每个权重该往哪里挪、挪多少的信号（**梯度**）要从最后一层往第一层算，每经过一层就乘上那一层的斜率：这就是**反向传播**。连着几层 sigmoid，传到第一层的信号，上限就是 0.25 自乘层数那么多次，这还没算权重把它放大或缩小了多少。



这是算术，不是测量：每层把 0.25 再乘一次。

到了 10 层，传到最前面的信号还不到百万分之一。前面几层几乎学不到东西。这就是 Hochreiter 在 1991 年描述的**梯度消失问题**。实践中也同样糟糕：2010 年，Glorot 和 Bengio 训练了有 5 个隐藏层的网络，测试误差用 **sigmoid** 是 **82.61%**，用双曲正切是 27.15%，用双曲正切加更好的初始化是 15.60%。

在隐藏层里，sigmoid 让位给了 **ReLU**：一个让正数原样通过、把负数变成零的函数。

sigmoid 并没有消失，只是换了工作。离开隐藏层这条链，它就不会被自己连乘几十次，而它 0 到 1 的取值范围正好适合表示概率或者充当门。

位置	sigmoid 做什么
“是或否”分类器的输出	把最后的得分变成概率。这就是网络末端的逻辑回归
同时预测多个标签的分类	每个类别一个独立的 sigmoid：一张照片可以既是“海滩”又是“日落”
LSTM 和 GRU 网络里的门	决定记忆中有多大比例能通过，从 0 到 1
LLaMA 和 PaLM 里的 SwiGLU	一个分支通过 sigmoid 决定另一个分支能通过多少

笨蛋之王说：

sigmoid 跟我一样：被赶出了隐藏层，最后在每个大语言模型里当了门卫。



什么时候用什么

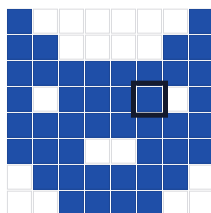
做决定时几乎从来不看“问题有多复杂”，而是看你的数据是什么形式、你有多少带正确答案的样本、谁需要理解每一个决定。这棵决策树三个层次都适用：



用于预测一个标签。往简单走很便宜，往复杂走很贵：从左边开始。

年龄	金额	访问
34	120	3
51	15	9
27	300	1

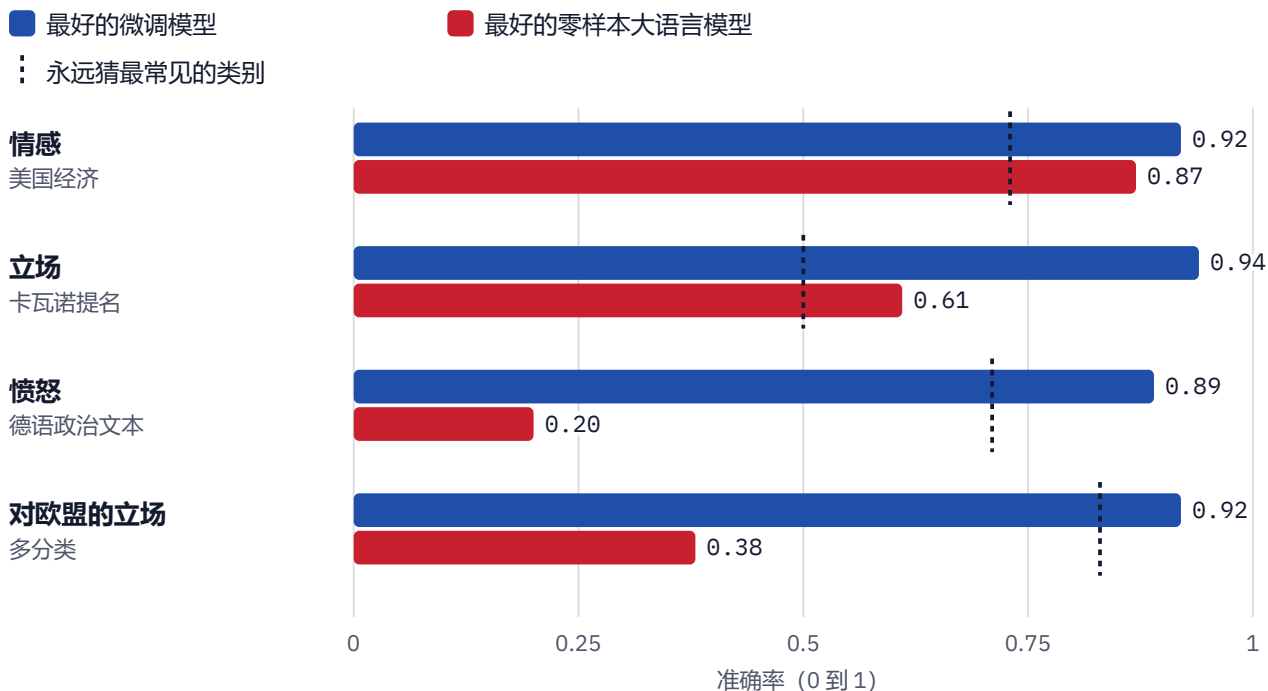
每一列
本来就有含义



单个像素
没有任何含义

表格数据：经典模型通常就够了。 Christodoulou 等人综述了 71 项临床预测研究，其中有 282 组逻辑回归和机器学习的对比。他们用 *AUC* 衡量表现，也就是模型给一个正例打的分高于一个负例的概率。在**偏倚风险**低的对比里（也就是做得规范的研究），两者没有差别。只有在偏倚风险高的研究里，机器学习才占上风。我读的是论文摘要，不是全文。在不需要任何人理解决定的大型表格上，把票数加起来的树通常胜过神经网络，而逻辑回归仍然是该最先跑的基准。

图像、音频或文本：特征必须靠学。 但这不一定需要一个巨大的模型。2024 年，Bucher 和 Martini 比较了**微调**过的小模型（已经训练好，再用几百个自己的样本调整过）和**不给示例**、只给指令的大语言模型：



准确率来自 Bucher 和 Martini (2024) 的表 1 到表 4。虚线是永远猜最常见类别能达到的准确率。

四个任务里有两个，最好的不给示例的大语言模型**还不如永远猜最常见的类别**。据作者说，微调模型有 200 到 500 个带标签的样本就能表现很好。不给示例的大语言模型适合第一天起步，也适合给最初那几百个样本打标签，但通常不是你在生产环境里想要的。

最后提醒一句：一个落在 0 到 1 之间的数，不一定是可信的概率。逻辑回归天生**校准良好**：它说 80%，大约就有 80% 的时候是对的。现代深度网络就没那么好了，不过事后只做一个调整就能修好。



最后的挑战

1. 一个逻辑回归和一个带 sigmoid 的神经元，用同样的惩罚训练，它们是什么关系？

- a. 亲戚：长得像，但结果不同
- b. 同一个模型，两个名字
- c. 神经元总是更准，因为它是深度学习

2. 为什么隐藏层不再用 sigmoid 了？

- a. 因为它算起来很慢
- b. 因为误差信号每过一层就缩小，前面几层几乎学不到东西
- c. 因为 0 到 1 之间的数没什么用

3. 你有一张 5000 行的表格（年龄、金额、访问次数），而且有人需要理解每一个决定。你从哪里开始？

- a. 给大语言模型写一个提示词
- b. 一个十层的深度网络
- c. 一个逻辑回归



三句话总结

- 1. 逻辑回归就是只有一个神经元的神经网络：同样的加权求和，同样的 S 形曲线（sigmoid），同样的惩罚。
- 2. 深度学习多出来的是前面那些层，它们负责制造特征；sigmoid 退出这些层，是因为它会把误差信号压没，如今它在输出端和门里工作。
- 3. 如果你的数据是表格，而且每一列本来就有含义，就从逻辑回归开始，几分钟就能跑完，还能告诉你复杂方案要赢多少才值得；如果是像素、音频或文本，就得有人来制造特征。



资料来源

- 逻辑回归作为广义线性模型、sigmoid 作为其典型连接函数：[斯坦福 CS229 讲义](#)。对数损失的凸性：Boyd 和 Vandenberghe, [Convex Optimization](#), §7.1.1。
- LogisticRegression 的默认设置 (L2, C=1.0)：[scikit-learn 文档](#)。
- 梯度消失和 sigmoid 导数的最大值：Hochreiter, 1991 年的博士论文 ([PDF](#), 德语) 和 1998 年的论文 ([PDF](#))。

- sigmoid、tanh 以及带归一化初始化的 tanh 的测试误差：Glorot 和 Bengio, *AISTATS* 2010, [表 1](#)。
- SwiGLU: Shazeer, [arXiv:2002.05202](#); 在 [LLaMA](#) 和 [PaLM](#) 中的使用。LSTM 的门: [Hochreiter 和 Schmidhuber, 1997](#)。
- 临床预测中机器学习与逻辑回归的对比: Christodoulou 等, *J Clin Epidemiol* 110, 12 (2019) , [doi:10.1016/j.jclinepi.2019.02.004](#)。只读了摘要。
- 表格数据上树模型与神经网络的对比: Grinsztajn 等, [arXiv:2207.08815](#)。
- 微调模型与零样本大语言模型的对比: Bucher 和 Martini, [arXiv:2406.08660](#), 表 1 到表 4 及 §5.5。
- 现代网络的校准: Guo 等, *ICML* 2017, [arXiv:1706.04599](#), 表 1。
- BCEWithLogitsLoss: [PyTorch 文档](#)。

这节课是怎么写成的： 基于我的机器学习笔记，由 Claude 协助整理并写出三个层次，再由 Claude 从西班牙语翻译，正式发布前会经过人工审校。所有数字都在 2026 年 10 月 4 日对照各篇论文的原始表格核对过。神经元实验室在你的浏览器里用梯度下降训练，和 ♠ 层次里的代码一样。插图均为原创。

答案

1. **b)** 同一个模型，两个名字. 答对了。同样的三个部分，最后得到同样的权重。
2. **b)** 因为误差信号每过一层就缩小，前面几层几乎学不到东西. 没错：每层最多 0.25。十层之后，连一点声音都传不过来。
3. **c)** 一个逻辑回归. 很好。每一列本来就有含义，决定还要能解释：逻辑回归就是起点。

King of Dumbs 的课程，kingofdumbs.com。这是离线版；可以动手玩的原版在：

<https://kingofdumbs.com/zh/lessons/logistic-regression-or-neural-network>

更新于 2026年10月4日.