

4 线性方法、基础优化和softmax回归

概要

- 线性方法

 - 神经科学的灵感

- 基础优化

- softmax 回归

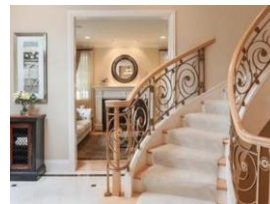
 - 回归与分类

 - Kaggle上的分类任务

线性方法

房价预测 101

- 挑选一间别墅，逛一逛，了解卖点
- 估计竞拍价



代理商的
市场价格

\$5,498,000

Price

7

Beds

5

Baths

4,865 Sq. Ft.

\$1130 / Sq. Ft.

Redfin Estimate: \$5,390,037 On Redfin: 15 days

估计
销售价格

Virtual Tour

- [Branded Virtual Tour](#)
- [Virtual Tour \(External Link\)](#)

Parking Information

- Garage (Minimum): 2
- Garage (Maximum): 2
- Parking Description: Attached Garage, On Street
- Garage Spaces: 2

Interior Features

Bedroom Information

- # of Bedrooms (Minimum): 7
- # of Bedrooms (Maximum): 7

Multi-Unit Information

- # of Stories: 2

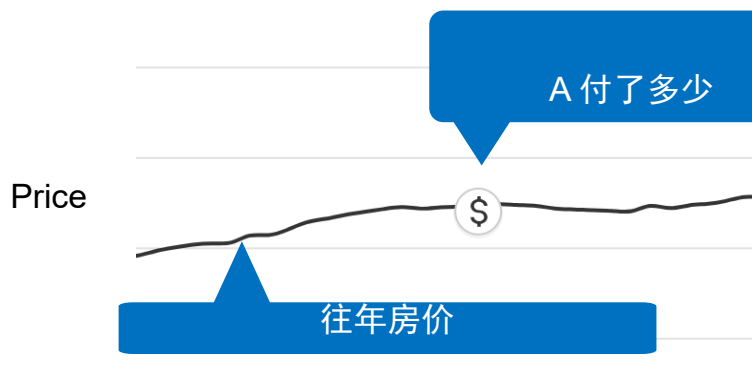
School Information

- Elementary School: El Carmelo Elementa
- Elementary School District: Palo Alto Uni
- Middle School: Jane Lathrop Stanford Mik
- High School: Palo Alto High
- High School District: Palo Alto Unified

- Kitchen Description: Countertop (Granite Dishwasher, Garbage Disposal, Hood Ove Island with Sink, Microwave, Oven Range

房价预测

非常重要，因为这是真钱.....



\$100K+ 差距



Redfin 高估房价，B 相信它

一个简易模型

➤ 假设 1，影响房价的关键因素：

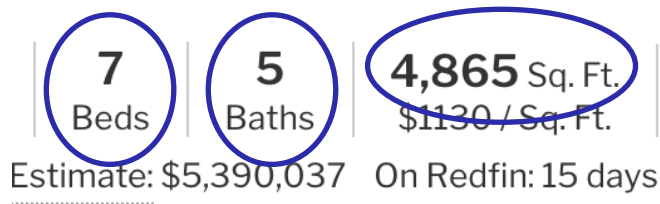
➤ 卧室数目，卫浴数目和房子大小，分别用 x_1, x_2, x_3 表示

➤ 假设 2

➤ 销售价格是关键因素的加权总和：

$$y = w_1x_1 + w_2x_2 + w_3x_3 + b$$

权重和偏差稍后确定。



线性方法

➤ 给予 n 维输入, $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$

➤ 线性方法有 n 个权重和偏差:

$$\mathbf{w} = [w_1, w_2, \dots, w_n]^T, b$$

➤ 输出是输入的加权总和:

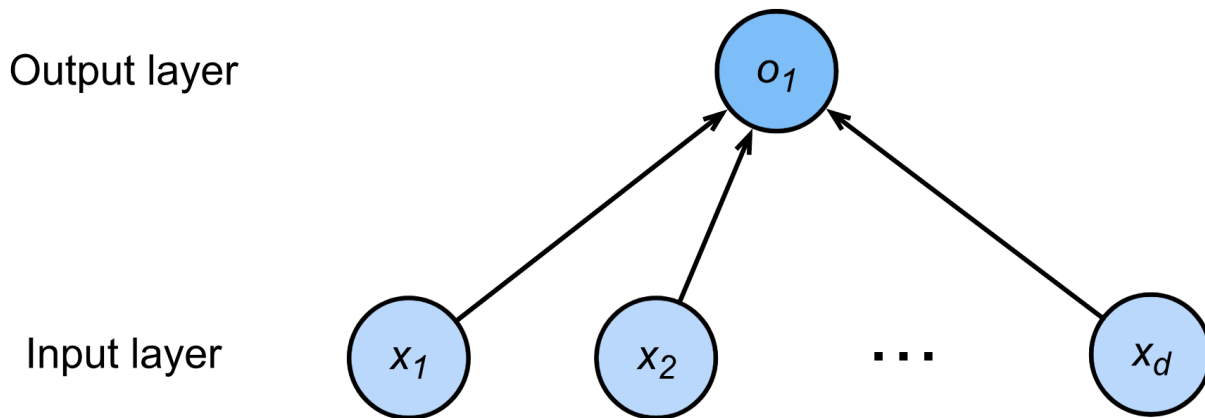
$$y = w_1x_1 + w_2x_2 + \dots + w_nx_n + b$$

➤ 矢量化版本:

$$y = \langle \mathbf{w}, \mathbf{x} \rangle + b$$

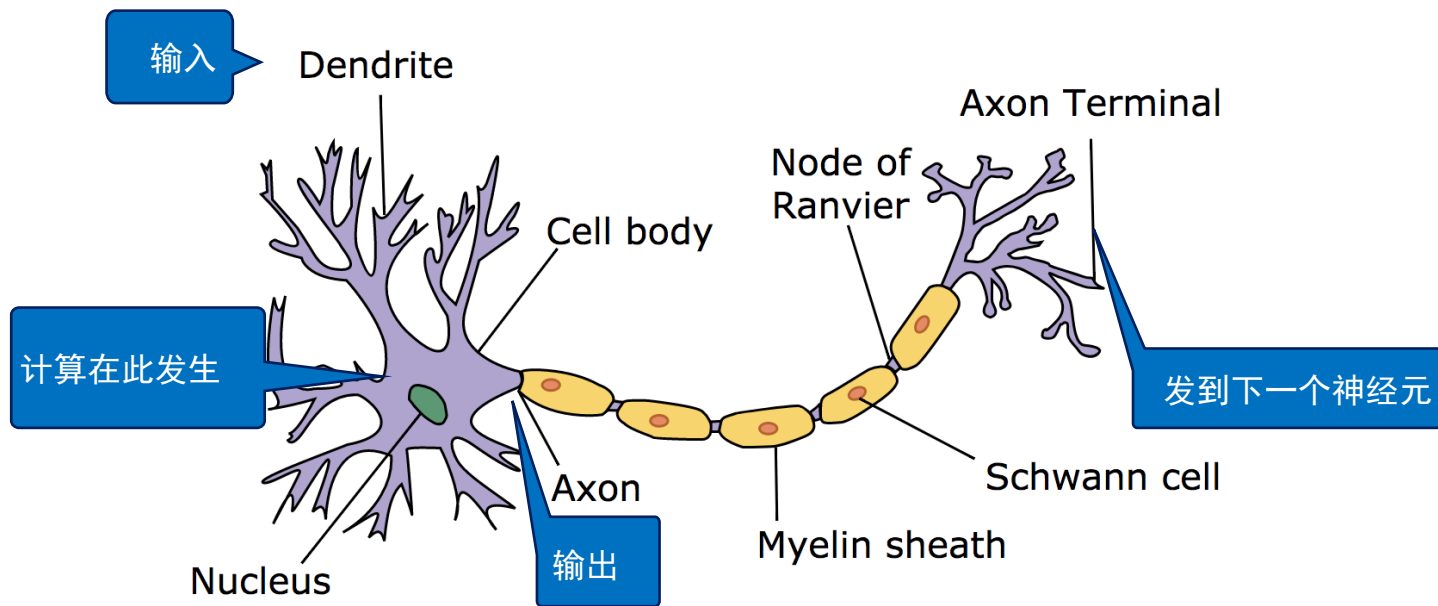
线性方法是一个单层神经网络

我们可以堆叠多个层来获得深层神经网络。



神经科学的灵感

真实的神经元



测量估计质量

- 比较真实值与估计值（实际销售价格与估计的房价）
- 以 y 作为真实值， $\hat{y}$ 作为估计值，比较损失

平方损失： $\ell(y, \hat{y}) = (y - \hat{y})^2$

训练数据集

➤ 收集多个数据点以训练参数（如 在过去6个月内出售的房屋）

➤ 训练数据集

➤ 训练数据集越大越好

➤ 假设有 n 个房屋

$$\mathbf{X} = [\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_n]^T, \mathbf{y} = [y_0, y_1, \dots, y_n]^T$$

学习参数

➤ 训练损失

$$\ell(\mathbf{X}, \mathbf{y}, \mathbf{w}, b) = \frac{1}{n} \sum_{i=1}^n (y_i - \langle \mathbf{x}_i, \mathbf{w} \rangle - b)^2 = \frac{1}{n} \| \mathbf{y} - \mathbf{X}\mathbf{w} - b \|^2$$

➤ 最小化学习参数的损失

$$\mathbf{w}^*, \mathbf{b}^* = \operatorname{argmin}_{\mathbf{w}, b} \ell(\mathbf{X}, \mathbf{y}, \mathbf{w}, b)$$

封闭解

- 将偏差添加到权重中 $\mathbf{w}^*, \mathbf{b}^* = \operatorname{argmin}_{\mathbf{w}, b} \ell(\mathbf{X}, \mathbf{y}, \mathbf{w}, b)$

$$\ell(\mathbf{X}, \mathbf{y}, \mathbf{w}) = \frac{1}{n} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2$$

$$\frac{\partial}{\partial \mathbf{w}} \ell(\mathbf{X}, \mathbf{y}, \mathbf{w}) = \frac{2}{n} (\mathbf{y} - \mathbf{X}\mathbf{w})^T \mathbf{X}$$

- 损失是凸性的，因此最优解满足：

$$\frac{\partial}{\partial \mathbf{w}} \ell(\mathbf{X}, \mathbf{y}, \mathbf{w}) = 0$$

$$\Leftrightarrow \frac{2}{n} (\mathbf{y} - \mathbf{X}\mathbf{w})^T \mathbf{X} = 0$$

$$\Leftrightarrow \mathbf{w}^* = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X} \mathbf{y}$$

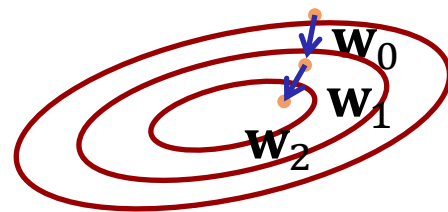
基础优化

梯度下降

- 选择一个起点 w_0
- 重复更新权重 $t=1,2,3$

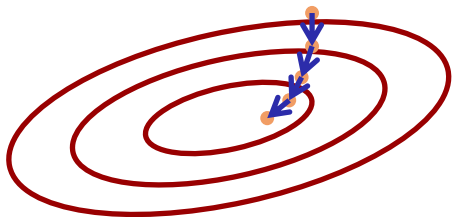
$$w_t = w_{t-1} - \eta \frac{\partial \ell}{\partial w_{t-1}}$$

- 梯度：更新权重的方向
- 学习率：一个超参数指定梯度的步长

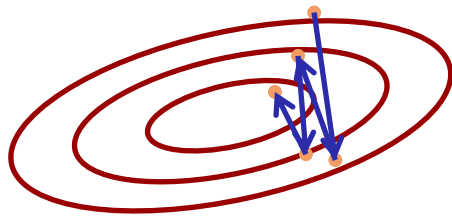


选择学习率

不要太小



不要太大



小批量随机梯度下降（SGD）

➤ 计算整个训练数据的梯度太昂贵了

➤ DNN模型需要几分钟到几小时

➤ 解决方案：SGD

➤ 随机抽样（小批量） b 个样本 $i_1, i_2, \dots, i_b$ 来估算损失，进行反向梯度传播

$$\frac{1}{b} \sum_{i \in I_b} \ell(\mathbf{x}_i, y_i, \mathbf{w})$$

➤ b 是批量大小，另一个重要的超参数

选择批量值

不要太小

批量值太小，难以充分利用
计算资源

不要太大

批量值太大，浪费计算资源；
例如当 x_i 都相同时

总结

➤问题：

- 估计一个真正的值

➤模型：

- $y = \langle \mathbf{w}, \mathbf{x} \rangle + b$

➤损失：

- 平方损失 $\ell(y, \hat{y}) = (y - \hat{y})^2$

➤小批量随机梯度（mini-batch SGD）学习

- 选择一个起点
- 重复
 - 计算梯度
 - 更新参数

softmax回归

回归与分类

- 回归估计连续值
- 分类预测离散类别

MNIST: 对手写数字进行分类
(10 类)

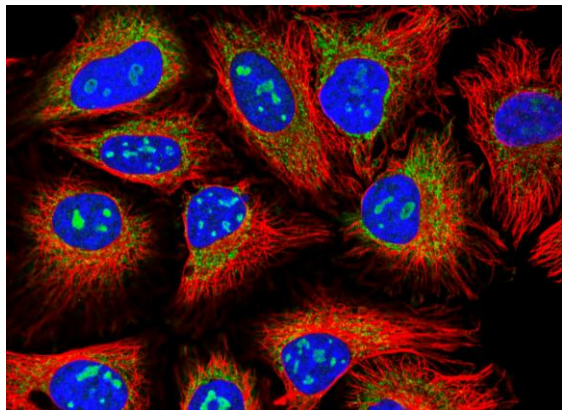


ImageNet: 对自然对象进行分类
(1000 类)



Kaggle上的各种分类任务

➤ 将人类蛋白质显微镜图像分为28类



- 0. Nucleoplasm
- 1. Nuclear membrane
- 2. Nucleoli
- 3. Nucleoli fibrillar
- 4. Nuclear speckles
- 5. Nuclear bodies
- 6. Endoplasmic reticu
- 7. Golgi apparatus
- 8. Peroxisomes
- 9. Endosomes
- 10. Lysosomes
- 11. Intermediate fila
- 12. Actin filaments
- 13. Focal adhesion si
- 14. Microtubules
- 15. Microtubule ends
- 16. Cytokinetic bridge

<https://www.kaggle.com/c/human-protein-atlas-image-classification>

Kaggle上的各种分类任务

➤ 将恶意软件分为9类



<https://www.kaggle.com/c/malware-classification>

Kaggle上的各种分类任务

➤ 将维基百科上的恶语评论分为7类

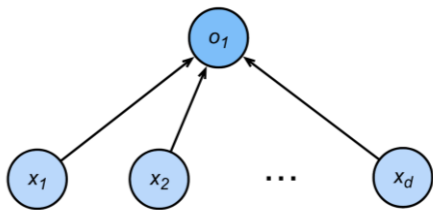
comment_text	toxic	severe_toxic	obsc
Explanation\nWhy the edits made under my usern...	0	0	0
D'aww! He matches this background colour I'm s...	0	0	0
Hey man, I'm really not trying to edit war. It...	0	0	0
"\nMore\nI can't make any real suggestions on ...	0	0	0
You, sir, are my hero. Any chance you remember...	0	0	0

<https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge>

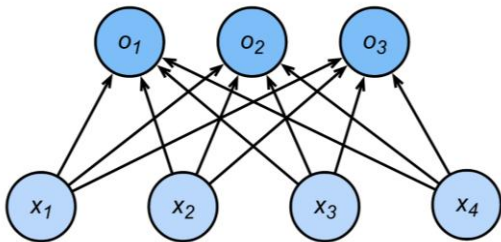
从回归到多类分类

预测类别
 $\operatorname{argmax}(o_1, o_2, o_3)$
• max 不可求导
• 定义一个损失函数

➤ 单个连续数值输出



➤ 输出每个类别的置信度分数



softmax函数

➤ $\text{softmax}([x_1, x_2, \dots, x_n]^T) = \left[\frac{e^{x_1}}{\sum_{i=1}^n e^{x_i}}, \frac{e^{x_2}}{\sum_{i=1}^n e^{x_i}}, \dots, \frac{e^{x_n}}{\sum_{i=1}^n e^{x_i}} \right]$

➤ 用 exp 获得

➤ 大于0的值

➤ 除以总和以获得

➤ 概率分布

➤ 非负

➤ 总和为1

➤ 例: $[1, -1, 2]$ 的 softmax 为 $[0.26, 0.04, 0.7]$

softmax梯度

$$\text{softmax}([x_1, x_2, \dots, x_n]^T) = \left[\frac{e^{x_1}}{\sum_{i=1}^n e^{x_i}}, \frac{e^{x_2}}{\sum_{i=1}^n e^{x_i}}, \dots, \frac{e^{x_n}}{\sum_{i=1}^n e^{x_i}} \right]$$

$$\frac{\partial}{\partial \mathbf{x}} \text{softmax}(\mathbf{x}) = \mathbf{x}\mathbf{x}$$

是否用平方损失？

- 对 $\mathbf{y}$ 进行一位有效编码 (One-hot encoding)

$$\mathbf{y} = [y_1, y_2, \dots, y_n]^T, y_i = \begin{cases} 1 & \text{if } i = y \\ 0 & \text{otherwise} \end{cases}$$

- 需要 softmax 结果接近 $\mathbf{y}$

- 用平方损失: $\mathbf{y} = [0, 0, 1]$

softmax 结果	损失
[0.3, 0, 0.7]	0.18
[0.17, 0.17, 0.66]	0.173

对数似然

- softmax函数给出了一个向量 $\hat{\mathbf{y}}$ ，可以将其视为“对给定任意输入 $\mathbf{x}$ 的每个类的条件概率”
- 假设整个数据集 $\{\mathbf{X}, \mathbf{Y}\}$ 具有 n 个样本, 其中索引 i 的样本由特征向量 $\mathbf{x}^{(i)}$ 和独热标签向量 $\mathbf{y}^{(i)}$ 组成。可以将估计值与实际值进行比较:

$$P(\mathbf{Y} | \mathbf{X}) = \prod_{i=1}^n P(\mathbf{y}^{(i)} | \mathbf{x}^{(i)}).$$

- 根据最大似然估计，我们最大化 $P(\mathbf{Y} | \mathbf{X})$ ，相当于最小化负对数似然：

$$-\log P(\mathbf{Y} | \mathbf{X}) = \sum_{i=1}^n -\log P(\mathbf{y}^{(i)} | \mathbf{x}^{(i)}) = \sum_{i=1}^n l(\mathbf{y}^{(i)}, \hat{\mathbf{y}}^{(i)}),$$

- 其中, 对于任何标签和模型预测 $\hat{\mathbf{y}}$, 损失函数为:

$$l(\mathbf{y}, \hat{\mathbf{y}}) = - \sum_{j=1}^q y_j \log \hat{y}_j$$

- 交叉熵损失 (cross-entropy loss)

交叉熵损失

- 交叉熵损失 (cross-entropy loss)

$$l(\mathbf{y}, \hat{\mathbf{y}}) = - \sum_{j=1}^q y_j \log \hat{y}_j$$

- 由于 $\mathbf{y}$ 是一个长度为 q 的独热编码向量, 除了一个项以外的所有项 j 都消失了
 - 由于所有 $\hat{y}_j$ 都是预测的概率, 它们的对数永远不会大于 0
- 如果正确地预测实际标签, 即 $P(\mathbf{y} | \mathbf{x}) = 1$, 损失函数不能进一步最小化
 - 但往往不可能。例如, 数据集中可能存在标签噪声 (比如某些样本可能被误标), 或输入特征没有足够的信息来完美地对每一个样本分类

softmax及其导数

➤ 利用softmax的定义, 得到

$$\begin{aligned}l(\mathbf{y}, \hat{\mathbf{y}}) &= - \sum_{j=1}^q y_j \log \frac{\exp(o_j)}{\sum_{k=1}^q \exp(o_k)} \\&= \sum_{j=1}^q y_j \log \sum_{k=1}^q \exp(o_k) - \sum_{j=1}^q y_j o_j \\&= \log \sum_{k=1}^q \exp(o_k) - \sum_{j=1}^q y_j o_j.\end{aligned}$$

➤ 考虑相对于任何未规范化的预测 o_j 的导数, 我们得到:

$$\partial_{o_j} l(\mathbf{y}, \hat{\mathbf{y}}) = \frac{\exp(o_j)}{\sum_{k=1}^q \exp(o_k)} - y_j = \text{softmax}(\mathbf{o})_j - y_j.$$

➤ 换句话说, 导数是softmax模型分配的概率与实际发生的情况 (由独热标签向量表示) 之间的差异
➤ 从这个意义上讲, 这与我们在回归中看到似然的梯度正是由此得出的。

信息论基础

- 信息论的核心思想是量化数据中的信息内容。信息论中 该数值被称为分布 P 的熵（entropy）：

$$H[P] = \sum_j -P(j)\log P(j)$$

- 信息论的基本定理之一指出, 为了对从分布 p 中随机抽取的数据进行编码, 至少需要 $H[P]$ “纳特 (nat)” 对其进行编码。“纳特”相当于比特 (bit), 一个纳特 ≈ 1.44 比特。
- 如果我们不能完全预测每一个事件, 那么我们有时可能会感到“惊异”。克劳德·香农用信息量量化 $\log \frac{1}{P(j)} = -\log P(j)$
 - 在观察一个事件 j 时, 并赋予它（主观）概率 $P(j)$ 。当我们赋予一个事件较低的概率时, 我们的惊异会更大, 该事件的信息量也就更大。熵 是当分配的概率真正匹配数据生成过程时的信息量的期望
- 交叉熵
 - 如果把熵 $H(P)$ 想象为“知道真实概率的人所经历的惊异程度”, 交叉熵从 P 到 Q , 记为 $H(P, Q)$ 。把交叉熵想象为“主观概率为 Q 的观察者在看到根据概率 P 生成的数据时的预期惊异”
 - 当 $P = Q$ 时, 交叉熵达到最低。在这种情况下, 从 P 到 Q 的交叉熵是 $H(P, P) = H(P)$ 。
- 简而言之, 我们可以从两方面来考虑交叉熵分类目标:
 - 最大化观测数据的似然
 - 最小化传达标签所需的惊异

线性回归与softmax回归

➤ 数值大小上的意义

- 回归问题的数值有，分类问题的数值没有意义

问题	线性回归 (回归问题)	softmax回归 (分类问题)
模型	$\langle \mathbf{w}, \mathbf{x} \rangle + b$ $\mathbf{w} \in \mathbb{R}^n, b \in \mathbb{R}$	$\text{softmax}(\mathbf{W}\mathbf{x} + \mathbf{b})$ $\mathbf{W} \in \mathbb{R}^{k \times n}, \mathbf{b} \in \mathbb{R}^k$
损失	平方损失	交叉熵

总结

- 线性方法

- 神经科学的灵感

- 基础优化

- softmax 回归

- 回归与分类
 - Kaggle上的分类任务