

如何在美国买房



- 看中一个房，参观了解
- 估计一个价格，出价

标价

\$5,498,000

Price

7

Beds

5

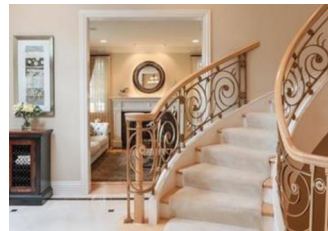
Baths

4,865 Sq. Ft.

\$1130 / Sq. Ft.

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

预计价格



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

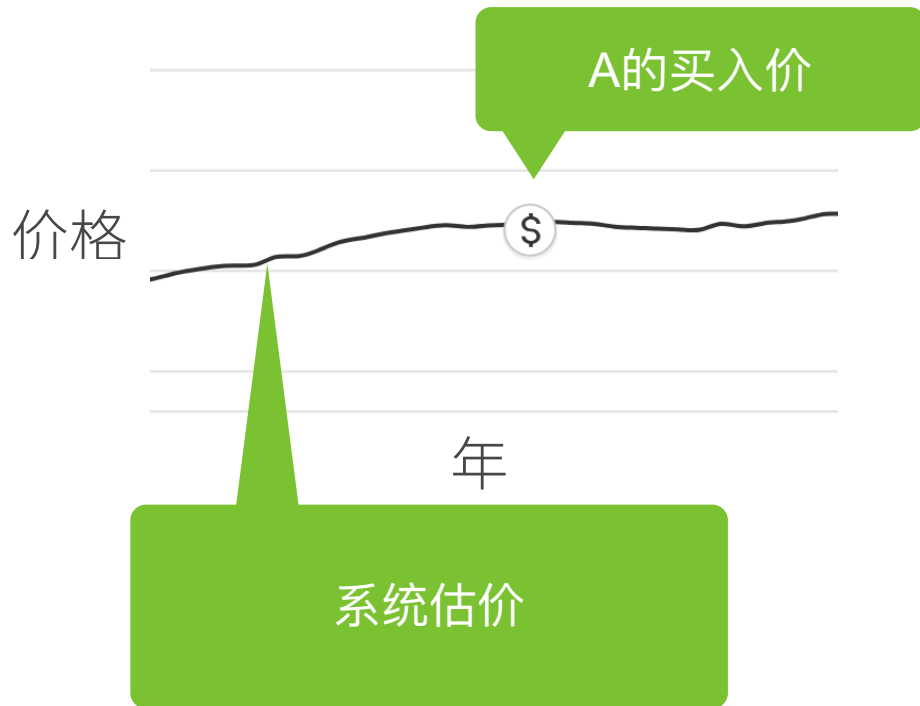
- Elementary School: El Carmelo
- Elementary School District: Palo Alto
- Middle School: Jane Lathrop
- High School: Palo Alto High
- High School District: Palo Alto U

- Kitchen Description: Countertop, Dishwasher, Garbage Disposal, Island with Sink, Microwave, Oven

房价预测



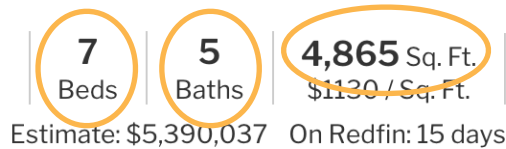
很重要，这是真钱



\$100K+ 差价



一个简化模型



- 假设 1：影响房价的关键因素是卧室个数，卫生间个数，和居住面积，记为 x_1, x_2, x_3
- 假设 2：成交价是关键因素的加权和

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

权重和偏差的实际值在后面决定



线性模型

- 给定 n 维输入 $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$
- 线性模型有一个 n 维权重和一个标量偏差

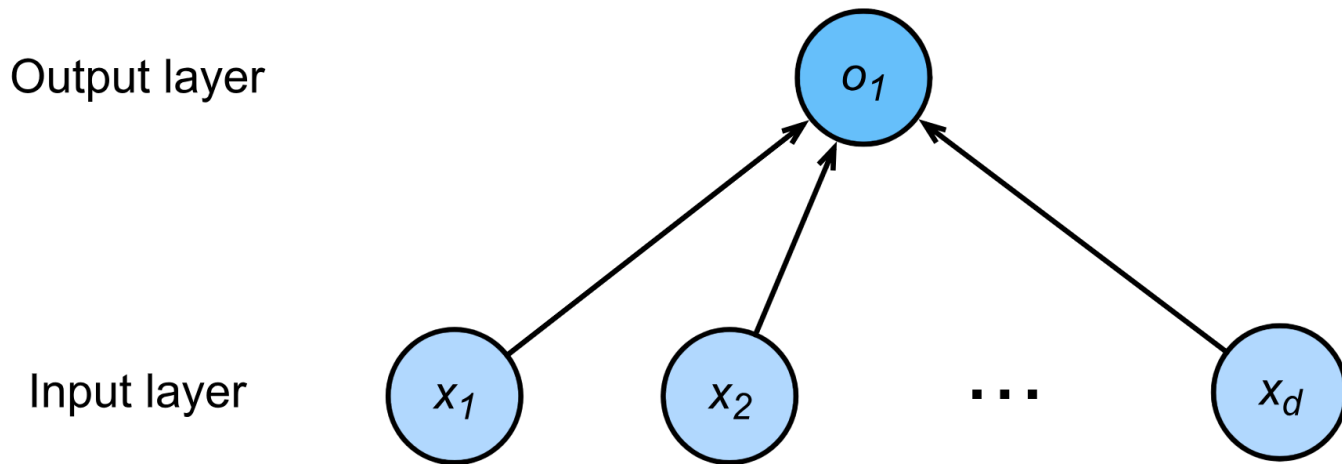
$$\mathbf{w} = [w_1, w_2, \dots, w_n]^T, \quad b$$

- 输出是输入的加权和

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

向量版本: $y = \langle \mathbf{w}, \mathbf{x} \rangle + b$

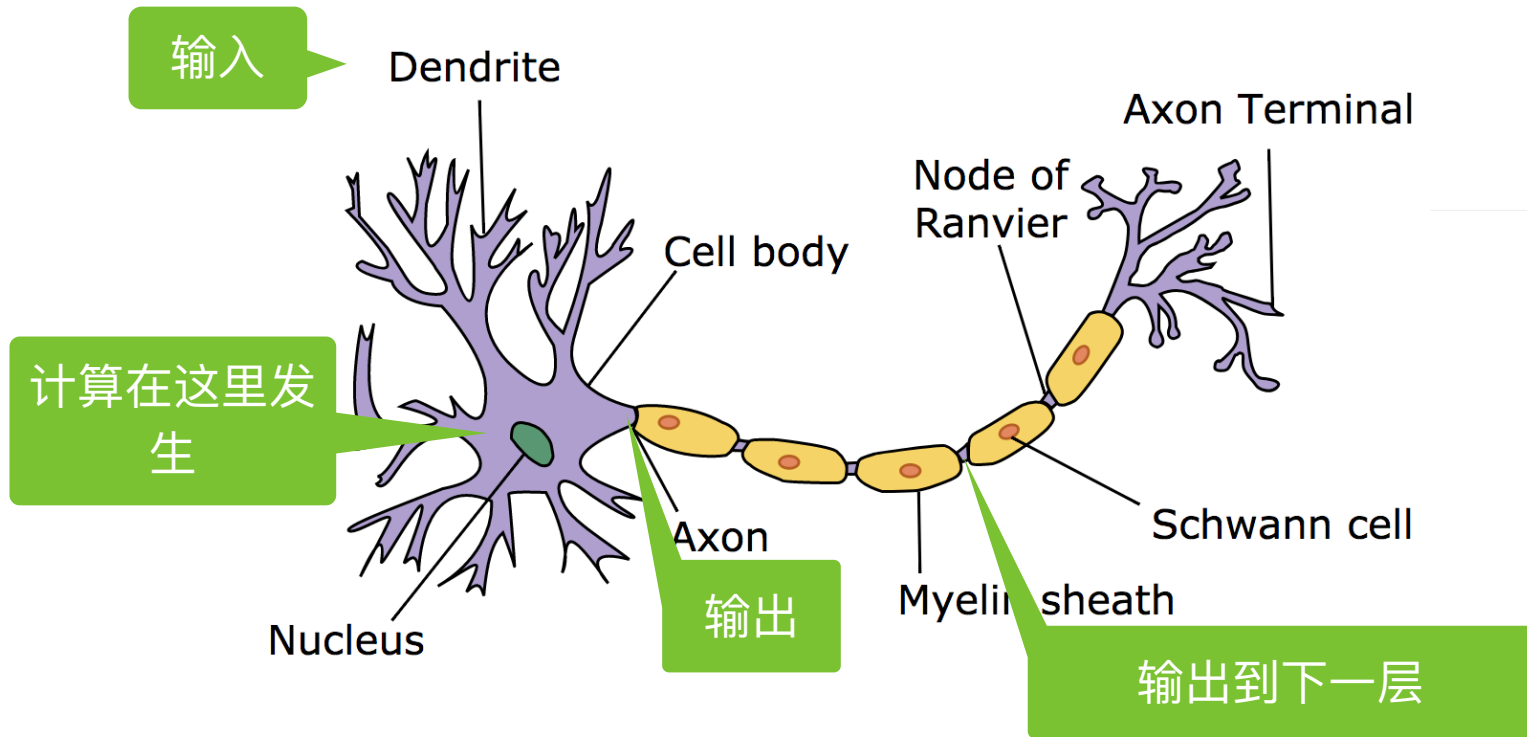
线性模型可以看做是单层神经网络



神经网络源于神经科学



真实的神经元





衡量预估质量

- 比较真实值和预估值，例如房屋售价和估价
- 假设 y 是真实值， \hat{y} 是估计值，我们可以比较

$$\ell(y, \hat{y}) = \frac{1}{2} (y - \hat{y})^2$$

这个叫做平方损失



训练数据

- 收集一些数据点来决定参数值（权重和偏差），例如过去6个月卖的房子
- 这被称之为训练数据
- 通常越多越好
- 假设我们有 n 个样本，记

$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]^T \quad \mathbf{y} = [y_1, y_2, \dots, y_n]^T$$



- 训练损失

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

- 最小化损失来学习参数

$$\mathbf{w}^*, \mathbf{b}^* = \arg \min_{\mathbf{w}, b} \ell(\mathbf{X}, \mathbf{y}, \mathbf{w}, b)$$

显示解



- 将偏差加入权重 $\mathbf{X} \leftarrow [\mathbf{X}, \mathbf{1}] \quad \mathbf{w} \leftarrow \begin{bmatrix} \mathbf{w} \\ b \end{bmatrix}$

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

- 损失是凸函数，所以最优解满足

$$\begin{aligned} \frac{\partial}{\partial \mathbf{w}} \ell(\mathbf{X}, \mathbf{y}, \mathbf{w}) &= 0 \\ \Leftrightarrow \frac{1}{n} (\mathbf{y} - \mathbf{X}\mathbf{w})^T \mathbf{X} &= 0 \\ \Leftrightarrow \mathbf{w}^* &= (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} \end{aligned}$$

总结



- 线性回归是对 n 维输入的加权，外加偏差
- 使用平方损失来衡量预测值和真实值的差异
- 线性回归有显示解
- 线性回归可以看做是单层神经网络