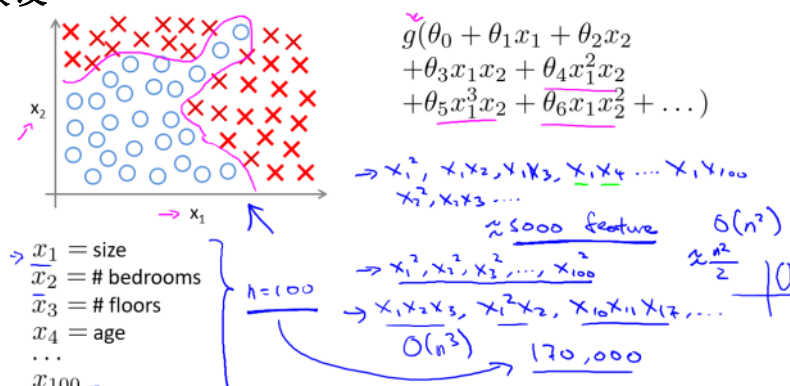


# 8 神经网络的表征

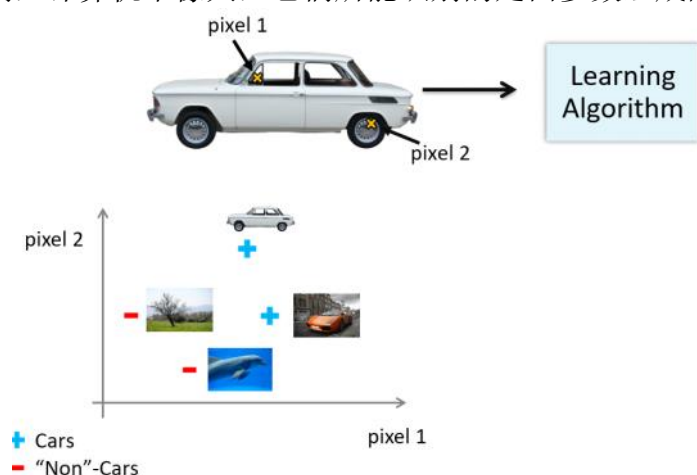
2022年10月31日 20:39

## 8.1 非线性假设

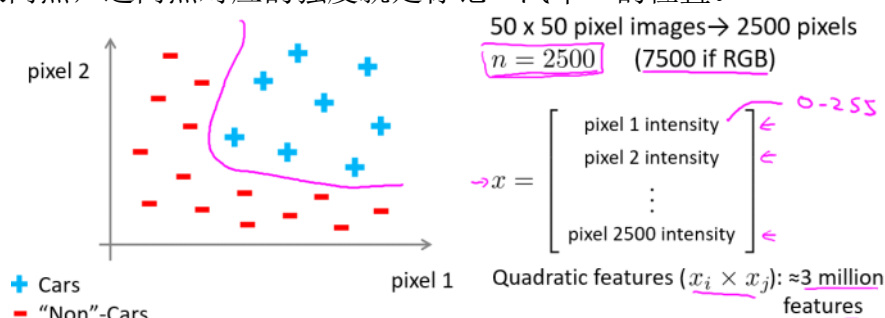


对于之前所讲过的逻辑回归和线性回归问题，当特征变量过多后，所产生的多项式的项数也会很多，同时可能带来过拟合的问题。

以计算机视觉为例，计算机不像人，它们所能识别的是由参数组成的矩阵。



在图片中取两点，这两点对应的强度就是标记“汽车”的位置。



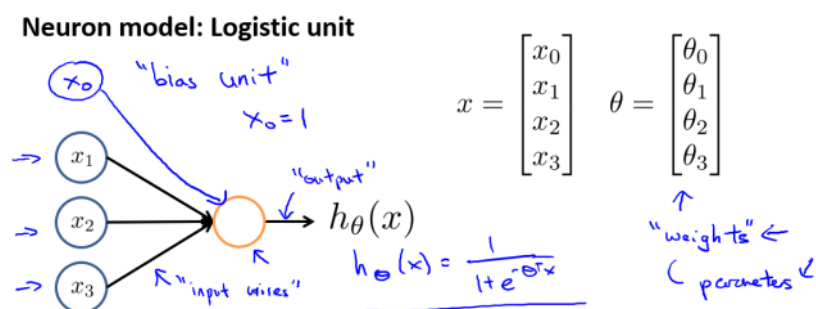
通过大量样本的绘制可以看到汽车和非汽车分布在坐标系中的不同区域，因此需要一个非线性假设来尽量分离两类样本。

对于一个50x50像素的图像，每一个像素点有一个强度，由0-255进行表示，那么对于灰度图像来说，表征这个图片的向量包含2500个元素；而对于一个RGB图像来说，则包含7500个元素。那么如果以每一个像素点作为一个特征量，那么多项式中的项数将会是惊人的。由此可见，对于传统的逻辑回归并不适用于大量特征的情况。神经网络就是一种很好的处理大特征量的非线性回归的方法。

## 8.2 神经元与大脑

神经网络建立的目的就是想像大脑一样处理问题。神经网络的诞生很早，但是近些年支持于计算量较大的硬件的保证使神经网络东山再起。

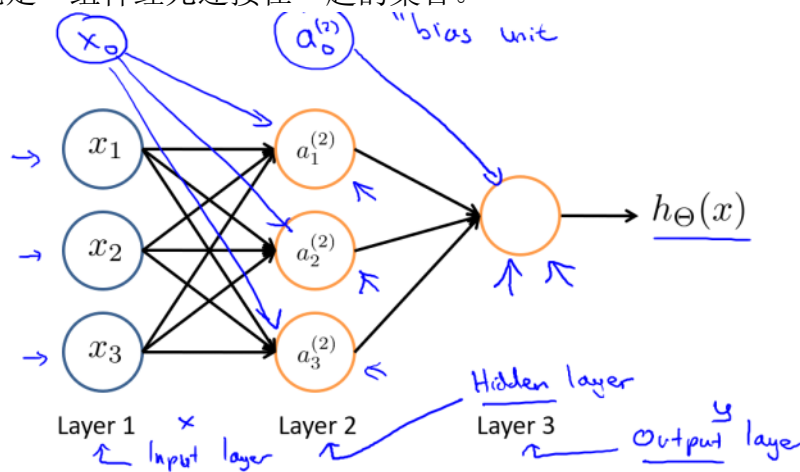
### 8.3 模型表示 I



**Sigmoid (logistic) activation function.**

$$g(z) = \frac{1}{1 + e^{-z}}$$

可以用一个简单的模型模拟神经元。我们前期所称的模型参数  $\theta$  也被称作权重。而神经网络就是一组神经元连接在一起的集合。



输入层 (Input layer)  $\rightarrow$  隐藏层 (Hidden layer)  $\rightarrow$  输出层 (Output layer)

$a_i^{(j)}$  表示第  $j$  层神经网络的第  $i$  个激活项 (activation)

$\theta^{(j)}$  表示权重矩阵，它控制从某一层前到某一层后的映射。

具体值对应如下：

$$\rightarrow a_1^{(2)} = g(\theta_{10}^{(1)} x_0 + \theta_{11}^{(1)} x_1 + \theta_{12}^{(1)} x_2 + \theta_{13}^{(1)} x_3)$$

$$\rightarrow a_2^{(2)} = g(\theta_{20}^{(1)} x_0 + \theta_{21}^{(1)} x_1 + \theta_{22}^{(1)} x_2 + \theta_{23}^{(1)} x_3)$$

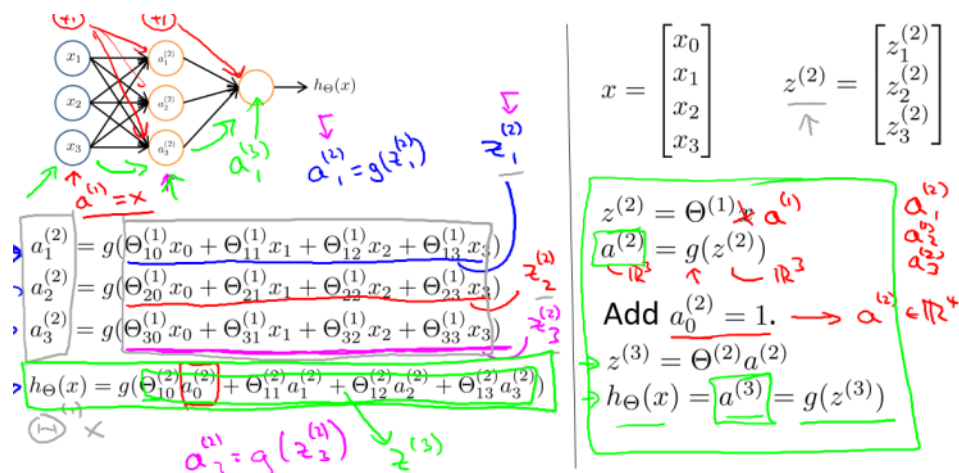
$$\rightarrow a_3^{(2)} = g(\theta_{30}^{(1)} x_0 + \theta_{31}^{(1)} x_1 + \theta_{32}^{(1)} x_2 + \theta_{33}^{(1)} x_3)$$

$$\rightarrow h_{\theta}(x) = a_1^{(3)} = g(\theta_{10}^{(2)} a_0^{(2)} + \theta_{11}^{(2)} a_1^{(2)} + \theta_{12}^{(2)} a_2^{(2)} + \theta_{13}^{(2)} a_3^{(2)})$$

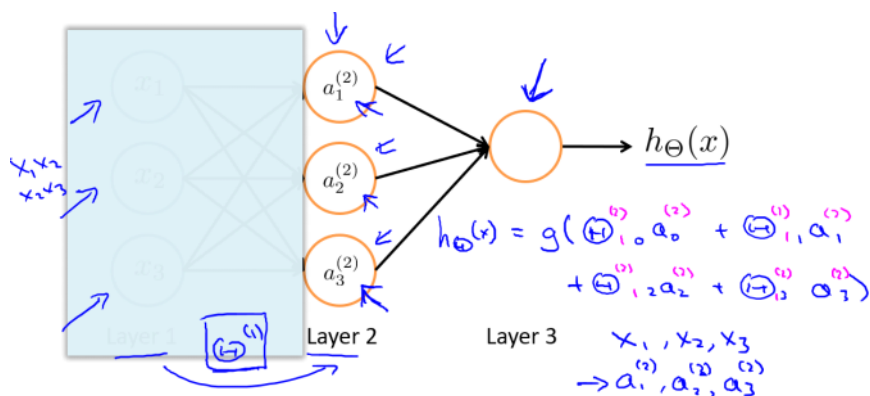
▷ If network has  $s_j$  units in layer  $j$ ,  $s_{j+1}$  units in layer  $j+1$ , then  $\Theta^{(j)}$  will be of dimension  $s_{j+1} \times (s_j + 1)$ .

### 8.4 模型表示 II

神经网络的计算向量化：

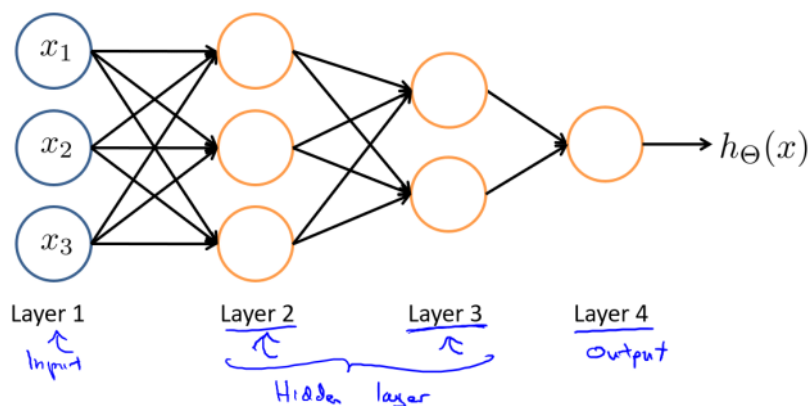


前向传播：从输入单元的激活项开始，前向传输给隐藏层的激活项，然后继续向前传输并计算输出层的激活项。



如图所示，将网络的输入层遮住，未遮住的部分和前述的逻辑回归的结构非常相似。

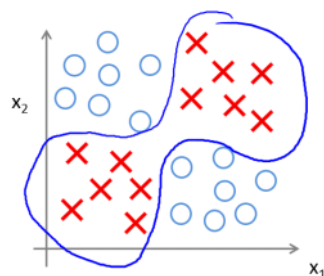
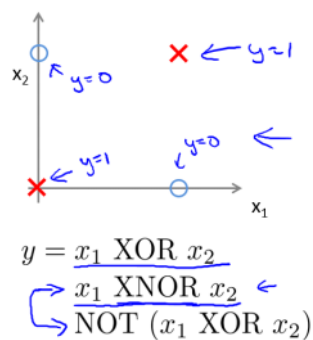
神经网络的架构，也即不同神经元的连接方式是多样的。



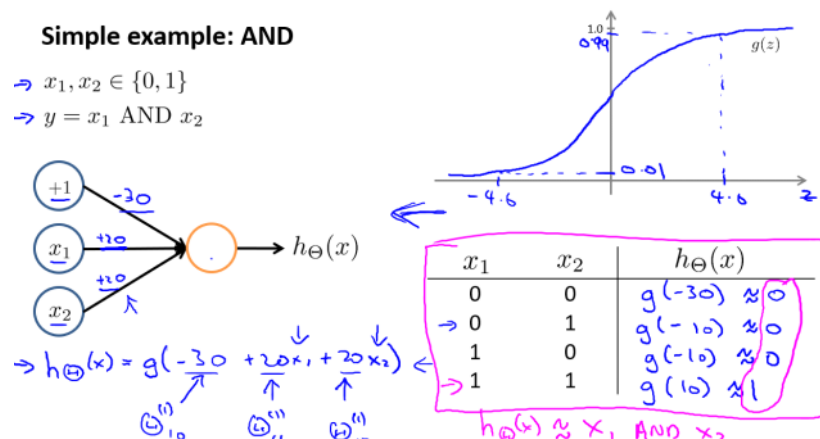
## 8.5 示例与理解 I

对于一个二特征量(二进制)的分类例子：

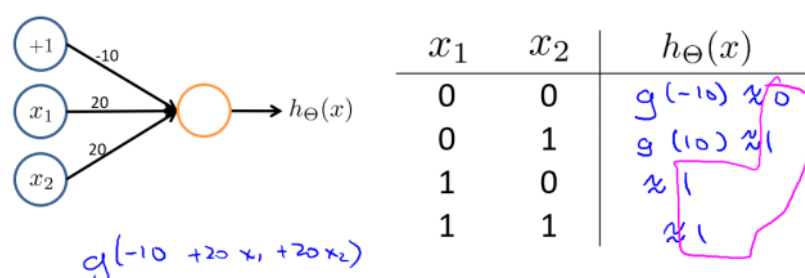
→  $x_1, x_2$  are binary (0 or 1).



假设函数为  $y = (x_1 // x_2)'$  ,为了解决这个问题, 以AND示例展开。

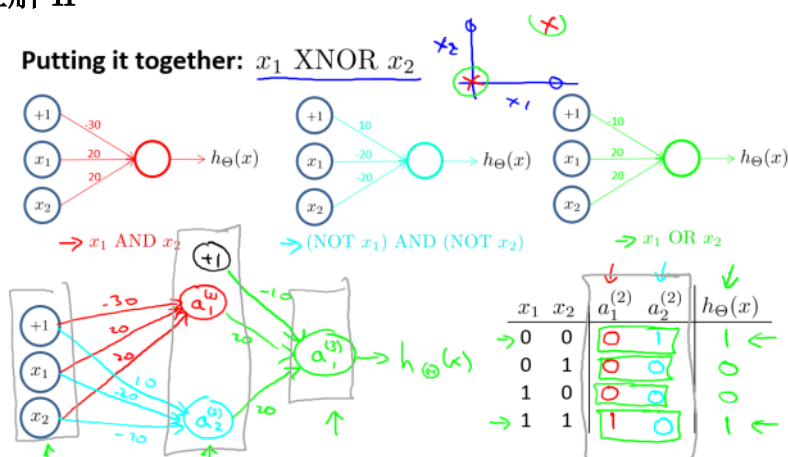


首先对输入权重进行赋值, 从而得到假设函数, 此时的g函数为sigmoid函数。计算出不同数据输入下的输出值。可以验证, 结果确实是逻辑与的计算结果。类似的, 构建或函数的神经网络:



其输出结果也是符合预期的。

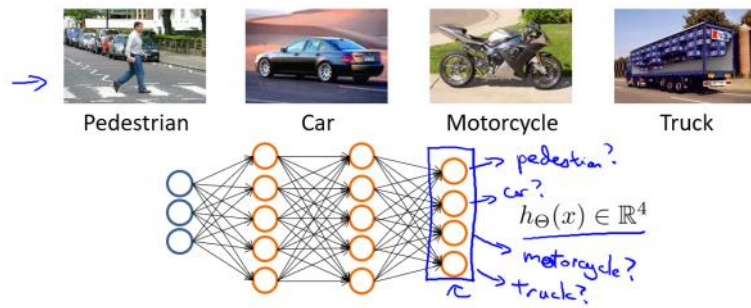
## 8.6示例与理解 II



或函数可以由与函数和非与函数的神经网络组合搭建而成。

## 8.7多元分类

依然以一个计算机视觉的问题为例:



Want  $h_{\Theta}(x) \approx \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$  when pedestrian,  $h_{\Theta}(x) \approx \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$  when car,  $h_{\Theta}(x) \approx \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$  when motorcycle, etc.

现在需要对四种情况进行分类，从而建立一个有四个输出单元的神经网络。

Training set:  $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})$

$y^{(i)}$  one of  $\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$ ,  $\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$ ,  $\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$ ,  $\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$  (x<sup>(i)</sup>, y<sup>(i)</sup>) ~~Previously~~  $y \in \{1, 2, 3, 4\}$   
pedestrian car motorcycle truck  $h_{\Theta}(x^{(i)}) \approx y^{(i)}$   
 $\in \mathbb{R}^4$

则数据集的设置的标签为一个含四个元素的向量。