

## 8.5 特征和直观理解 1

参考视频: 8 - 5 - Examples and Intuitions I (7 min).mkv

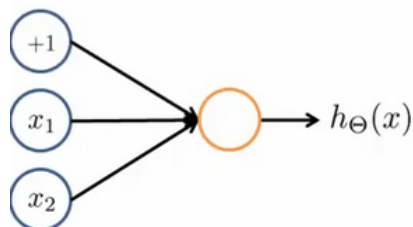
从本质上讲，神经网络能够通过学习得出其自身的一系列特征。在普通的逻辑回归中，我们被限制为使用数据中的原始特征 $x_1, x_2, \dots, x_n$ ，我们虽然可以使用一些二项式项来组合这些特征，但是我们仍然受到这些原始特征的限制。在神经网络中，原始特征只是输入层，在我们上面三层的神经网络例子中，第三层也就是输出层做出的预测利用的是第二层的特征，而非输入层中的原始特征，我们可以认为第二层中的特征是神经网络通过学习后自己得出的一系列用于预测输出变量的新特征。

神经网络中，单层神经元（无中间层）的计算可用来表示逻辑运算，比如逻辑与(AND)、逻辑或(OR)。

举例说明：逻辑与(AND)：下图中左半部分是神经网络的设计与 **output** 层表达式，右边上部分是 **sigmoid** 函数，下半部分是真值表。

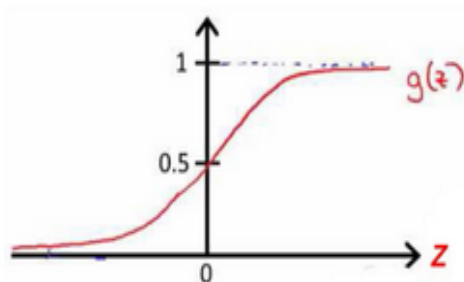
我们可以用这样的神经网络表示 **AND** 函数：

$$x_1, x_2 \in \{0, 1\}$$
$$y = x_1 \text{ AND } x_2$$



其中 $\theta_0 = -30, \theta_1 = 20, \theta_2 = 20$  我们的输出函数 $h_\theta(x)$ 即为： $h_\theta(x) = g(-30 + 20x_1 + 20x_2)$

我们知道 $g(x)$ 的图像是：



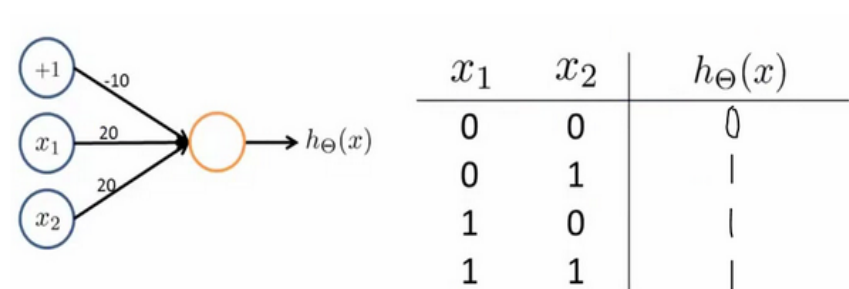
$x_1$	$x_2$	$h_{\Theta}(x)$
0	0	$g(-30) \approx 0$
$\rightarrow$ 0	1	$g(-10) \approx 0$
1	0	$g(-10) \approx 0$
1	1	$g(10) \approx 1$

所以我们有:  $h_{\theta}(x) \approx x_1 \text{AND } x_2$

所以我们的:

这就是 **AND** 函数。

接下来再介绍一个 **OR** 函数:



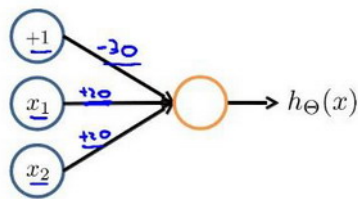
**OR** 与 **AND** 整体一样, 区别只在于的取值不同。

## 8.6 样本和直观理解 II

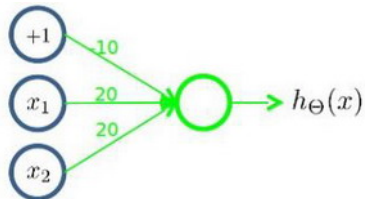
参考视频: 8 - 6 - Examples and Intuitions II (10 min).mkv

二元逻辑运算符（**BINARY LOGICAL OPERATORS**）当输入特征为布尔值（0 或 1）时，我们可以用一个单一的激活层可以作为二元逻辑运算符，为了表示不同的运算符，我们只需要选择不同的权重即可。

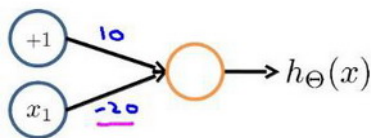
下图的神经元（三个权重分别为-30，20，20）可以被视为作用同于逻辑与（**AND**）：



下图的神经元（三个权重分别为-10，20，20）可以被视为作用等同于逻辑或（**OR**）：



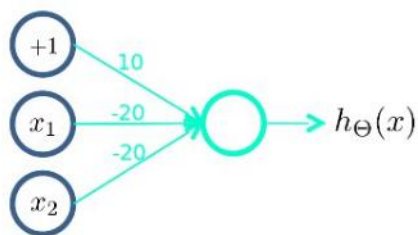
下图的神经元（两个权重分别为 10，-20）可以被视为作用等同于逻辑非（**NOT**）：



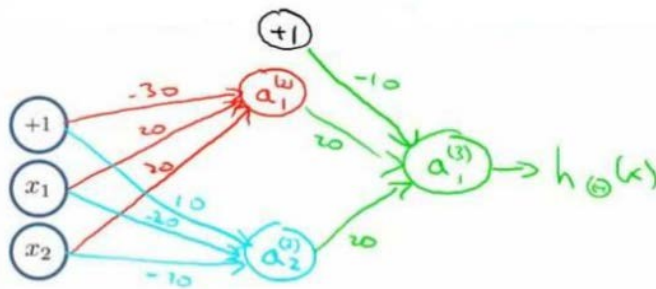
我们可以利用神经元来组合成更为复杂的神经网络以实现更复杂的运算。例如我们要实现 **XNOR** 功能（输入的两个值必须一样，均为 1 或均为 0），即：

$$\text{XNOR} = (x_1 \text{ AND } x_2) \text{ OR } ((\text{NOT } x_1) \text{ AND } (\text{NOT } x_2))$$

首先构造一个能表达  $(\text{NOT } x_1) \text{ AND } (\text{NOT } x_2)$  部分的神经元：



然后将表示 **AND** 的神经元和表示  $(\text{NOT } x_1)\text{AND}(\text{NOT } x_2)$  的神经元以及表示 **OR** 的神经元进行组合：



我们就得到了一个能实现 **XNOR** 运算符功能的神经网络。

按这种方法我们可以逐渐构造出越来越复杂的函数，也能得到更加厉害的特征值。

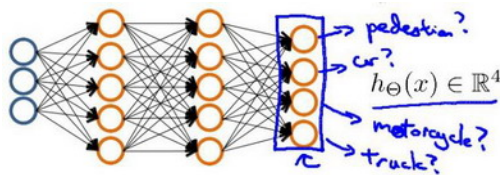
这就是神经网络的厉害之处。

## 8.7 多类分类

参考视频: 8 - 7 - Multiclass Classification (4 min).mkv

当我们有不止两种分类时（也就是 $y = 1, 2, 3 \dots$ ），比如以下这种情况，该怎么办？如果我们要训练一个神经网络算法来识别路人、汽车、摩托车和卡车，在输出层我们应该有 4 个值。例如，第一个值为 1 或 0 用于预测是否是行人，第二个值用于判断是否为汽车。

输入向量 $x$ 有三个维度，两个中间层，输出层 4 个神经元分别用来表示 4 类，也就是每一个数据在输出层都会出现 $[a \ b \ c \ d]^T$ ，且 $a, b, c, d$ 中仅有一个为 1，表示当前类。下面是该神经网络的可能结构示例：



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

神经网络算法的输出结果为四种可能情形之一：

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$