

3.2 神经网络的表示 (Neural Network Representation)

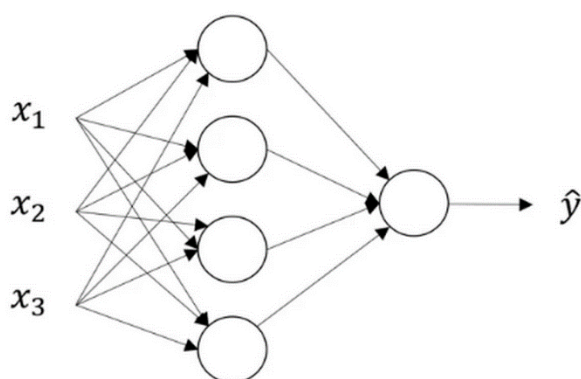


图 3.2.1

我们有输入特征 x_1 、 x_2 、 x_3 ，它们被竖直地堆叠起来，这叫做神经网络的**输入层**。它包含了神经网络的输入；然后这里有另外一层我们称之为**隐藏层**（图 3.2.1 的四个结点）。待会儿我会回过头来讲解术语“隐藏”的意义；在本例中最后一层只由一个结点构成，而这个只有一个结点的层被称为**输出层**，它负责产生预测值。解释隐藏层的含义：在一个神经网络中，当你使用监督学习训练它的时候，训练集包含了输入 x 也包含了目标输出 y ，所以术语隐藏层的含义是在训练集中，这些中间结点的准确值我们是不知道的，也就是说你看不见它们在训练集中应具有的值。你能看见输入的值，你也能看见输出的值，但是隐藏层中的东西，在训练集中你是无法看到的。所以这也解释了词语隐藏层，只是表示你无法在训练集中看到他们。

现在我们再引入几个符号，就像我们之前用向量 x 表示输入特征。这里有个可代替的记号 $a^{[0]}$ 可以用来表示输入特征。 **a 表示激活的意思**，它意味着网络中不同层的值会传递到它们后面的层中，输入层将 x 传递给隐藏层，所以我们将输入层的激活值称为 $a^{[0]}$ ；下一层即隐藏层也同样会产生一些激活值，那么我将其记作 $a^{[1]}$ ，所以具体地，这里的第一个单元或结点我们将其表示为 $a_1^{[1]}$ ，第二个结点的值我们记为 $a_2^{[1]}$ 以此类推。所以这里的是一个四维的向量如果写成 Python 代码，那么它是一个规模为 4×1 的矩阵或一个大小为 4 的列向量，如下公式，它是四维的，因为在本例中，我们有四个结点或者单元，或者称为四个隐藏层单元；

公式 3.7

$$a^{[1]} = \begin{bmatrix} a_1^{[1]} \\ a_2^{[1]} \\ a_3^{[1]} \\ a_4^{[1]} \end{bmatrix}$$

最后输出层将产生某个数值 a ，它只是一个单独的实数，所以的 \hat{y} 值将取为 $a^{[2]}$ 。这与逻辑回归很相似，在逻辑回归中，我们有 \hat{y} 直接等于 a ，在逻辑回归中我们只有一个输出层，所以我们没有用带方括号的上标。但是在神经网络中，我们将使用这种带上标的形式来明确地指出这些值来自于哪一层，有趣的是在约定俗成的符号传统中，在这里你所看到的这个例子，只能叫做一个两层的神经网络（图 3.2.2）。原因是当我们计算网络的层数时，输入层是不算入总层数内，所以隐藏层是第一层，输出层是第二层。第二个惯例是我们将输入层称为第零层，所以在技术上，这仍然是一个三层的神经网络，因为这里有输入层、隐藏层，还有输出层。但是在传统的符号使用中，如果你阅读研究论文或者在这门课中，你会看到人们将这个神经网络称为一个两层的神经网络，**因为我们不将输入层看作一个标准的层。**

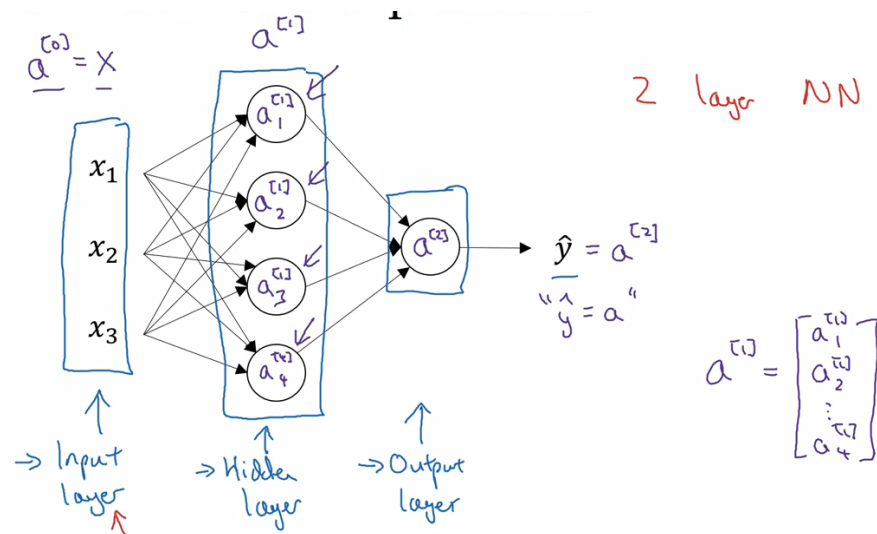


图 3.2.2

最后，我们要看到的隐藏层以及最后的输出层是带有参数的，这里的隐藏层将拥有两个参数 W 和 b ，我将给它们加上上标 $^{[1]}$ ($W^{[1]}, b^{[1]}$)，表示这些参数是和第一层这个隐藏层有关的。之后在这个例子中我们会看到 W 是一个 4×3 的矩阵，而 b 是一个 4×1 的向量，第一个数字 4 源自于我们有四个结点或隐藏层单元，然后数字 3 源自于这里有三个输入特征，我们之后会更加详细地讨论这些矩阵的维数，到那时你可能就更加清楚了。相似的输出层也有一些与之关联的参数 $W^{[2]}$ 以及 $b^{[2]}$ 。从维数上来看，它们的规模分别是 1×4 以及 1×1 。 1×4 是因为隐藏层有四个隐藏层单元而输出层只有一个单元，之后我们会对这些矩阵和向量的维

度做出更加深入的解释，所以现在你已经知道一个两层的神经网络什么样的了，即它是一个只有一个隐藏层的神经网络。