多层神经网络

通过前面的章节，我们了解到了机器学习领域中最常见的两个模型，线性回归模型和 Logistic 回归模型，他们分别是处理机器学习中最常见的两类问题-回归问题和分类问题。

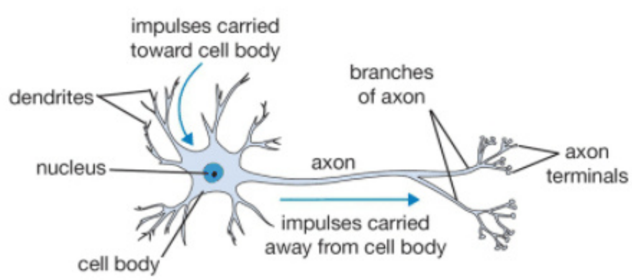
下面我们会讲第一个深度学习的模型，多层神经网络。

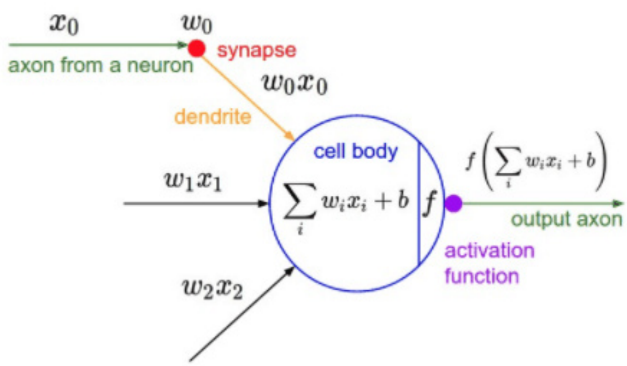
多层神经网络

在前面的线性回归中，我们的公式是 $y = wx + b$，而在 Logistic 回归中，我们的公式是 $y = Sigmoid(wx + b)$，其实它们都可以看成单层神经网络，其中 Sigmoid 被称为激活函数，之后我们会详细介绍激活函数以及为什么必须使用激活函数，下面我们从理解神经网络入手。

理解神经网络

神经网络的灵感来自于人脑的神经元系统，下面我们放一张人脑的神经元和神经网络的对比图(来自 cs231n)





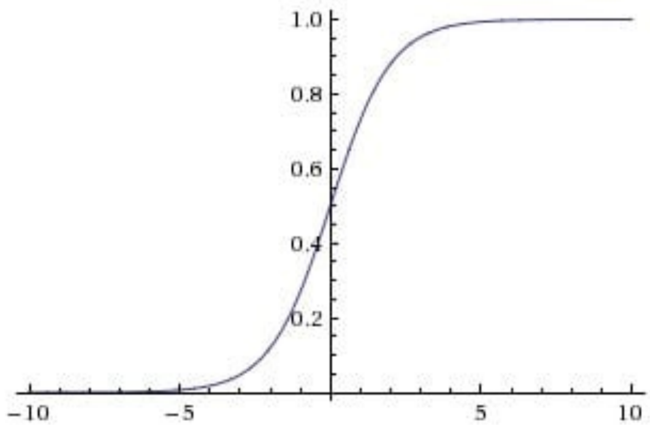
左边是一张神经元的图片，神经元通过突触接受输入，然后通过**神经激活**的方式传输给后面的神经元。这对比于右边的神经网络，首先接受数据输入，然后通过计算得到结果，接着经过**激活函数**，再传给第二层的神经元。

所以前面讲的 logistic 回归模型和线性回归模型都可以看做是一个单层神经网络，而 logistic 回归中使用了激活函数 sigmoid。

神经网络使用的激活函数都是非线性的，每个激活函数都输入一个值，然后做一种特定的数学运算得到一个结果，下面举几个例子

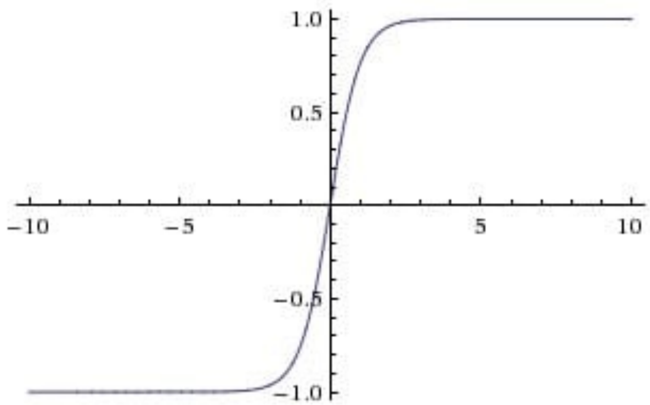
sigmoid 激活函数

$\sigma(x) = \frac{1}{1+e^{-x}}$



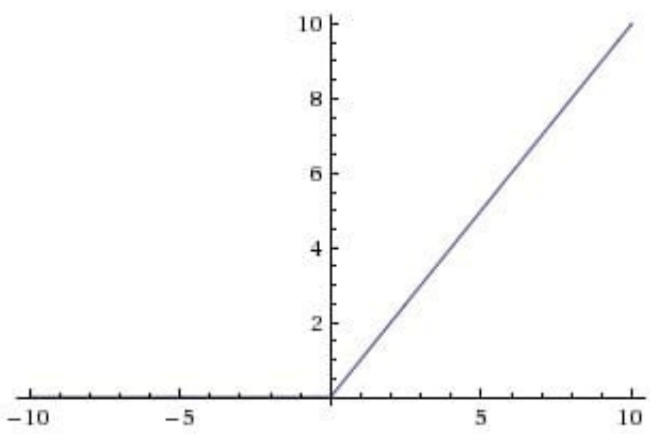
tanh 激活函数

$tanh(x) = 2\sigma(2x) - 1$



ReLU 激活函数

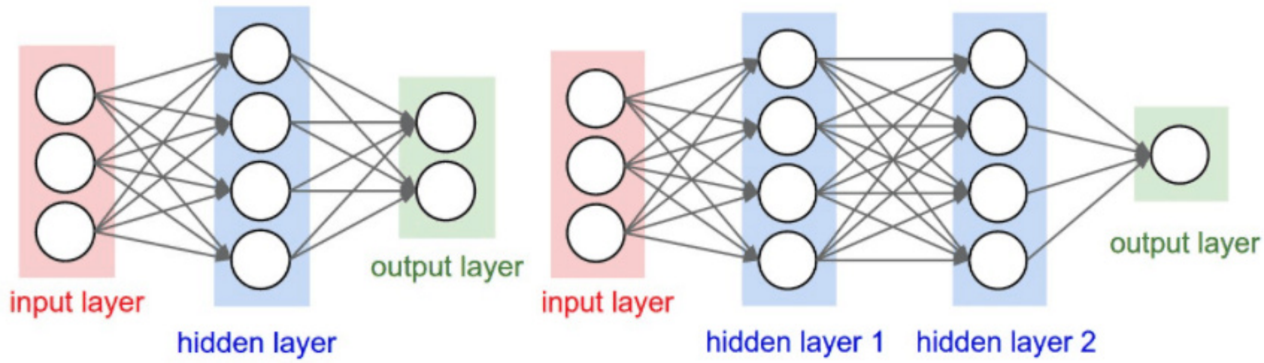
$ReLU(x) = max(0, x)$



我们下面重点讲一讲 ReLU 激活函数，因为现在神经网络中 90% 的情况都是使用这个激活函数。一般一个一层的神经网络的公式就是 $y = max(0, wx + b)$，一个两层的神经网络就是 $y = w_2\ max(0, w_1 x + b_1) + b_2$，非常简单，但是却很有效，使用这个激活函数能够加快梯度下降法的收敛速度，同时对比与其他的激活函数，这个激活函数计算更加简单，所以现在变得非常流行，之后你会发现我们激活在所有的神经网络中都会使用它。

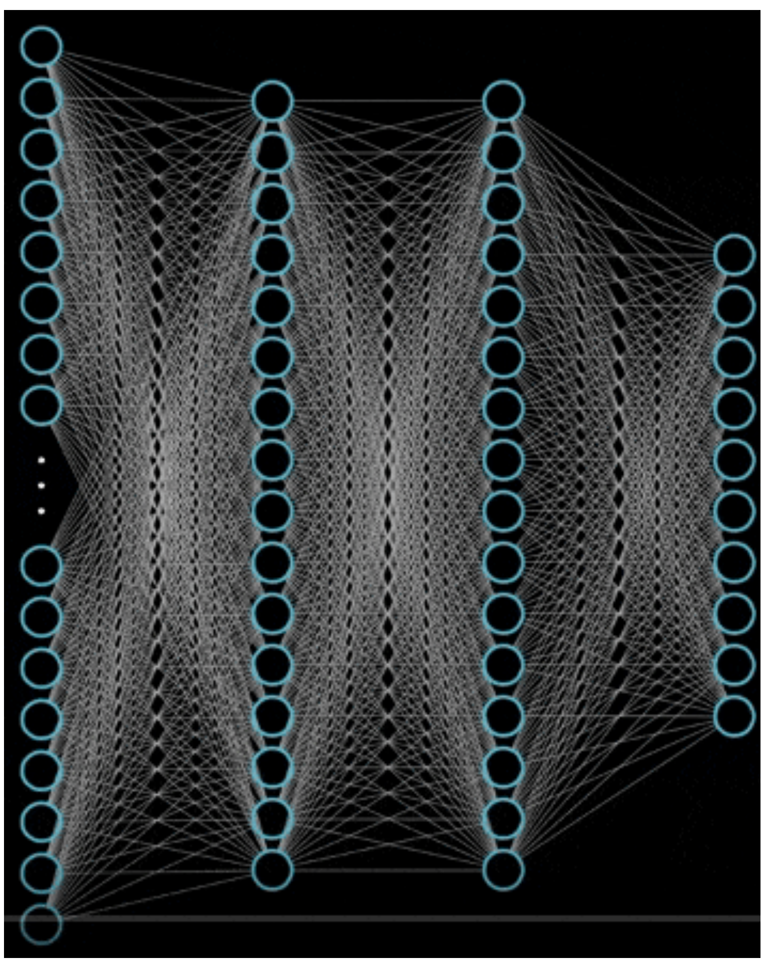
神经网络的结构

神经网络就是很多个神经元堆在一起形成一层神经网络，那么多个层堆叠在一起就是深层神经网络，我们可以通过下面的图展示一个两层的神经网络和三层的神经网络



可以看到，神经网络的结构其实非常简单，主要有输入层，隐藏层，输出层构成，输入层需要根据特征数目来决定，输出层根据解决的问题来决定，那么隐藏层的网路层数以及每层的神经元数就是可以调节的参数，而不同的层数和每层的参数对模型的影响非常大，我们看看这个网站的 [demo]

神经网络向前传播也非常简单，就是一层一层不断做运算就可以了，可以看看下面这个例子



为什么要使用激活函数

激活函数在神经网络中非常重要，使用激活函数也是非常必要的，前面我们从人脑神经元的角度理解了激活函数，因为神经元需要通过激活才能往后传播，所以神经网络中需要激活函数，下面我们从数学的角度理解一下激活函数的必要性。

比如一个两层的神经网络，使用 A 表示激活函数，那么

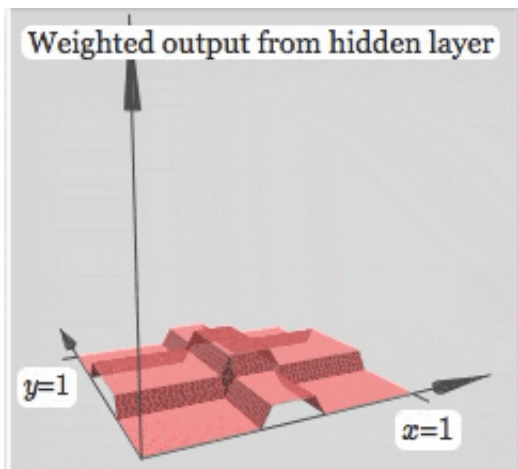
$$y = w_2 A(w_1 x) \tag{1}$$

如果我们不使用激活函数，那么神经网络的结果就是

$$y = w_2(w_1 x) = (w_2 w_1)x = \bar{w}x \tag{2}$$

可以看到，我们将两层神经网络的参数合在一起，用 $\bar{w}$ 来表示，两层的神经网络其实就变成了一层神经网络，只不过参数变成了新的 $\bar{w}$，所以如果不使用激活函数，那么不管多少层的神经网络，$y = w_n \cdots w_2 w_1 x = \bar{w}x$，就都变成了单层神经网络，所以在每一层我们都必须使用激活函数。

最后我们看看激活函数对神经网络的影响



可以看到使用了激活函数之后，神经网络可以通过改变权重实现任意形状，越是复杂的神经网络能拟合的形状越复杂，这就是著名的神经网络万有逼近定理。