

浅析激活函数之Relu函数

 LiBiscuit  关注 IP属地: 福建

 1 2019.01.11 10:54:08 字数 1,564 阅读 45,404

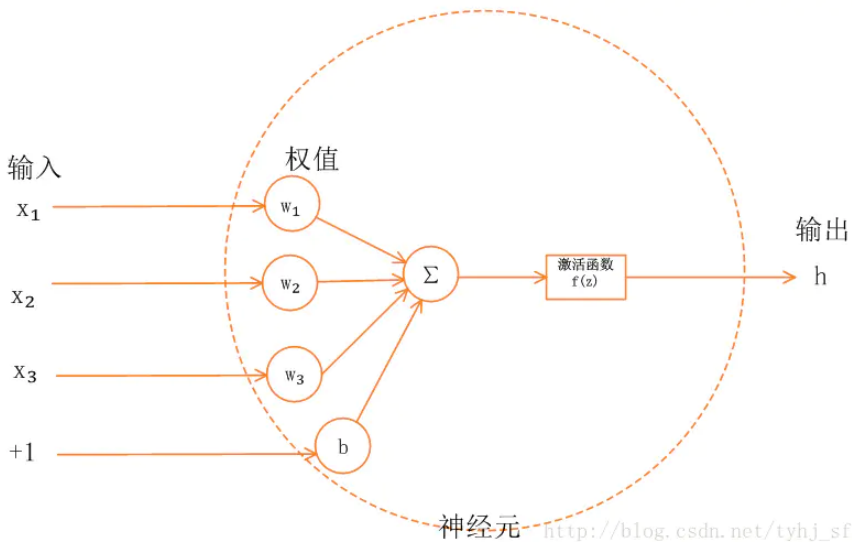
时隔一周 小李又来冒泡啦~下周就要结束在实验室搬砖的日子了诶 但也不能偷懒鸭!
提前的周末愉快鸭~

Relu函数

讲Relu函数前需要先了解关于激活函数的概念和作用。

什么是激活函数?

首先了解一下神经网络的基本模型



如上图所示，神经网络中的每个神经元节点接受上一层神经元的输出值作为本神经元的输入值，并将输入值传递给下一层，输入层神经元节点会将输入属性值直接传递给下一层（隐层或输出层）。在多层神经网络中，上层节点的输出和下层节点的输入之间具有一个函数关系，这个函数称为激活函数。

简单来说，**激活函数，并不是去激活什么，而是指如何把“激活的神经元的特征”通过函数把特征保留并映射出来，即负责将神经元的输入映射到输出端。**

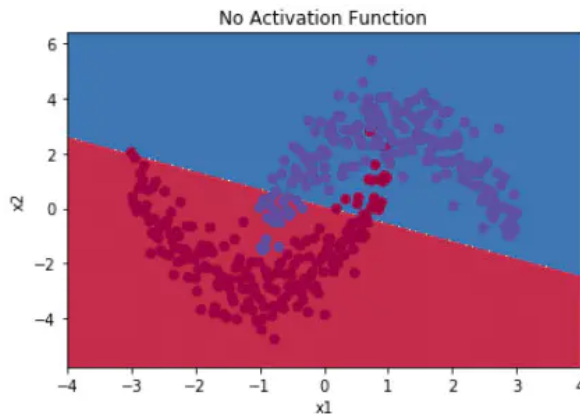
为什么需要激活函数?

首先明确一点，**激活函数是用来加入非线性因素的，因为线性模型的表达力不够。**

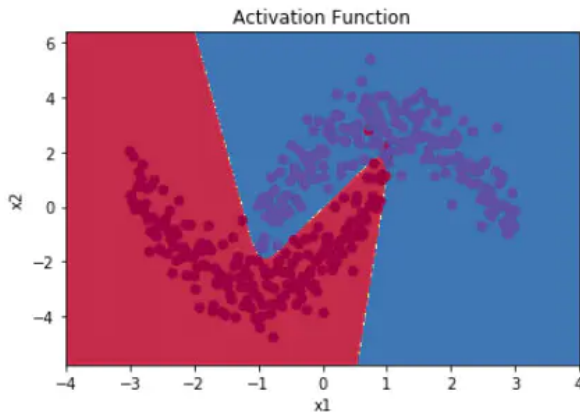
假设如果没有激活函数的出现，你每一层节点的输入都是上层输出的线性函数，很容易验证，无论你神经网络有多少层，输出都是输入的线性组合，与没有隐藏层效果相当，也就是说没有激活函数的每层都相当于矩阵相乘。就算你叠加了若干层之后，无非还是个矩阵相乘罢了。那么网络的逼近能力就相当有限。正因为上面的原因，我们决定引入

非线性函数作为激活函数，这样神经网络表达能力就更加强大（不再是输入的线性组合，而是几乎可逼近任意函数）

举个例子：二分类问题，如果不使用激活函数，例如使用简单的逻辑回归，只能作简单的线性划分，如下图所示



如果使用激活函数，则可以实现非线性划分，如下图所示：



可见，激活函数能帮助我们引入非线性因素，使得神经网络能够更好地解决更加复杂的问题。

为什么激活函数一般都是非线性的，而不能是线性的呢？

从反面来说，如果所有的激活函数都是线性的，则激活函数 $g(z)=z$ ，即 $a=z$ 。那么，以两层神经网络为例，最终的输出为：

$$\begin{aligned}
 A^{[2]} &= Z^{[2]} \\
 &= W^{[2]}A^{[1]} + b^{[2]} \\
 &= W^{[2]}(W^{[1]}X + b^{[1]}) + b^{[2]} \\
 &= (W^{[2]}W^{[1]})X + (W^{[2]}b^{[1]} + b^{[2]}) \\
 &= W'X + b'
 \end{aligned}$$

经过推导我们发现网络输出仍是 X 的线性组合。这表明，使用神经网络与直接使用线性模型的效果并没有什么两样。即便是包含多层隐藏层的神经网络，如果使用线性函数作为激活函数，最终的输出仍然是线性模型。这样的话神经网络就没有任何作用了。因此，隐藏层的激活函数必须要是非线性的。

补充一点：因为神经网络的数学基础是处处可微的，所以选取的激活函数要能保证数据输入与输出也是可微的，运算特征是不断进行循环计算，所以在每代循环过程中，每个神经元的值也是在不断变化的。

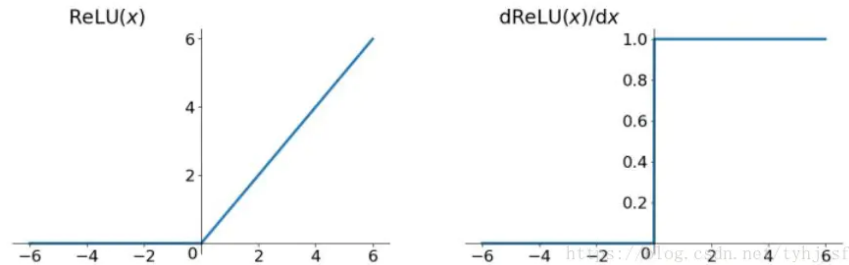
有了大概的概念后就可进入正题~

Relu函数是什么？

首先，relu函数是常见的激活函数中的一种，表达形式如下。

$$f(x) = \max(0, x)$$

Relu函数的图像和求导后的图像如下：



从表达式和图像可以明显地看出：Relu其实就是个取最大值的函数。

ReLU函数其实是分段线性函数，把所有的负值都变为0，而正值不变，这种操作被成为单侧抑制。（也就是说：**在输入是负值的情况下，它会输出0，那么神经元就不会被激活。这意味着同一时间只有部分神经元会被激活，从而使得网络很稀疏，进而对计算来说是非常有效率的。**）正因为有了这单侧抑制，才使得神经网络中的神经元也具有了稀疏激活性。尤其体现在深度神经网络模型(如CNN)中，当模型增加N层之后，理论上ReLU神经元的激活率将降低2的N次方倍。

那么问题来了：这种稀疏性有何作用？换句话说，我们为什么需要让神经元稀疏？不妨举栗子来说明。当看名侦探柯南的时候，我们可以根据故事情节进行思考和推理，这时用到的是我们的大脑左半球；而当看蒙面唱将时，我们可以跟着歌手一起哼唱，这时用到的则是我们的右半球。左半球侧重理性思维，而右半球侧重感性思维。也就是说，当我们在进行运算或者欣赏时，都会有一部分神经元处于激活或是抑制状态，可以说是各司其职。再比如，生病了去医院看病，检查报告里面上百项指标，但跟病情相关的通常只有那么几个。与之类似，当训练一个深度分类模型的时候，和目标相关的特征往往也就那么几个，因此通过ReLU实现稀疏后的模型能够更好地挖掘相关特征，拟合训练数据。

使用Relu函数有什么优势？

- 1.没有饱和区，不存在梯度消失问题。
- 2.没有复杂的指数运算，计算简单、效率提高。
- 3.实际收敛速度较快，比 Sigmoid/tanh 快很多。
- 4.比 Sigmoid 更符合生物学神经激活机制。

当然relu也存在不足：就是训练的时候很“脆弱”，很容易就“die”了。举个例子：一个非常大的梯度流过一个 ReLU 神经元，更新过参数之后，这个神经元再也不会对任何数据有激活现象了。如果这个情况发生了，那么这个神经元的梯度就永远都会是0.实际操作中，如果你的learning rate 很大，那么很有可能你网络中的40%的神经元都“dead”了。当然，如果你设置了一个合适的较小的learning rate，这个问题发生的情况其实也不会太频繁。

Ending~

参考资料 (<https://blog.csdn.net/u013146742/article/details/51986575>)

(<https://www.cnblogs.com/tianqizhi/p/9570975.html>)

(https://blog.csdn.net/tyhj_sf/article/details/79932893)

(https://www.sohu.com/a/214965417_100008678)

划水愉快！嘻嘻:)

